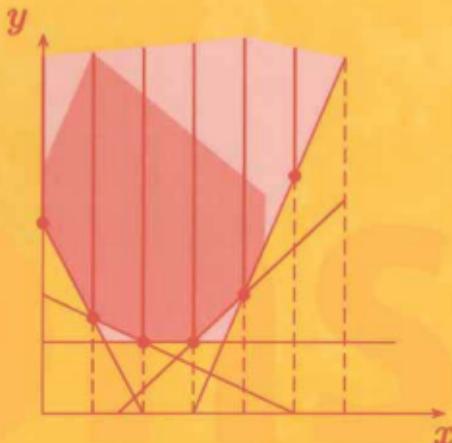


Dimitris Alevras

Manfred W. Padberg

# Linear Optimization and Extensions

Problems and Solutions



Springer

Universitext

**Springer-Verlag Berlin Heidelberg GmbH**

Dimitris Alevras  
Manfred W. Padberg

# Linear Optimization and Extensions

Problems and Solutions

With 67 Figures



Springer

Dimitris Alevras

IBM Corp.

1475 Phoenixville Pike  
West Chester, PA 19380, USA

e-mail: alevras@us.ibm.com

Manfred W. Padberg

Department of Operations Research

Stern School of Business

New York University

Washington Square

New York, NY 10012, USA

e-mail: manfred@padberg.com

Mathematics Subject Classification (2000): 51M20, 51Nxx, 65Kxx, 90Bxx, 90C05

Library of Congress Cataloging-in Publication Data

Alevras, Dimitris.

Linear optimization and extensions: problems and solutions / Dimitris Alevras,  
Manfred W. Padberg.

p. cm. – (Universitext) Includes bibliographical references.

ISBN 978-3-540-41744-6 ISBN 978-3-642-56628-8 (eBook)

DOI 10.1007/978-3-642-56628-8

1. Linear programming—Problems, exercises, etc. 2. Mathematical optimization—Problems,  
exercises, etc. I. Padberg, M.W. II. Title. III. Series.

T57.74 .A44 2001 519.7'2-dc21 2001020974

**ISBN 978-3-540-41744-6**

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broad-casting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2001

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready copy from the author using L<sup>A</sup>T<sub>E</sub>X

Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper SPIN: 10797138 46/3142/YL - 5 4 3 2 1 0

## Preface

Books on a technical topic – like linear programming – *without* exercises ignore the principal beneficiary of the endeavor of writing a book, namely the student – who learns best by doing exercises, of course. Books *with* exercises – if they are challenging or at least to some extent so – need a solutions manual so that students can have recourse to it when they need it. Here we give solutions to all exercises and case studies of M. Padberg's *Linear Optimization and Extensions* (second edition, Springer-Verlag, Berlin, 1999). In addition we have included several new exercises and taken the opportunity to correct and change some of the exercises of the book. Here and in the main text of the present volume the terms “book”, “text” etc. designate the second edition of Padberg's LP book and the page and formula references refer to that edition as well. All new and changed exercises are marked by a star \* in this volume. The changes that we have made in the original exercises are inconsequential for the main part of the original text where several of the exercises (especially in Chapter 9) are used on several occasions in the proof arguments. None of the exercises that are used in the estimations, etc. have been changed. Quite a few exercises instruct the students to write a program in a computer language of their own choice. We have chosen to do that in most cases in MATLAB without *any* regard to efficiency, etc. Our prime goal here is to use a macro-language that resembles as closely as possible the mathematical statement of the respective algorithms. Once students master this first level, they can then go ahead and discover the pleasures and challenges of writing efficient computer code on their own.

To make the present volume as self-contained as possible, we have provided here summaries of each chapter of Padberg's LP book. While there is some overlap with the text, we think that this is tolerable. The summaries are –in almost all cases– without proofs, thus they provide a “mini-version” of the material treated in the text. Indeed, we think that having such summaries without the sometimes burdensome proofs is an advantage to the reader who wants to acquaint herself/himself with the material treated at length in the text. To make the cross-referencing with the text easy for the reader, we have numbered all chapters (and most sections and subsections) as well as the formulas in these summaries exactly like in the text. Moreover, we have reproduced here most of the illustrations of the text as we find these visual aids very helpful in communicating the material. Finally, we have reproduced here the appendices of the text as the descriptions of the cases contained therein would have taken too much space anyway.

We have worked on the production of this volume over several years and did so quite frequently at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) in Berlin, Germany, where Alevras was a research fellow during some of this time. We are most grateful to ZIB's vice-president, Prof. Dr. Martin Grötschel, for his hospitality and tangible support of our endeavor. Padberg's work was also supported in part through an ONR grant and he would like to thank Dr. Donald Wagner of the Office of Naval Research, Arlington, VA, for his continued support.

New York City, January, 2001

Dimitris Alevras  
Manfred Padberg

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Minicases and Exercises . . . . .	1
<b>2</b>	<b>The Linear Programming Problem</b>	<b>39</b>
2.1	Exercises . . . . .	40
<b>3</b>	<b>Basic Concepts</b>	<b>47</b>
3.1	Exercises . . . . .	49
<b>4</b>	<b>Five Preliminaries</b>	<b>55</b>
4.1	Exercises . . . . .	57
<b>5</b>	<b>Simplex Algorithms</b>	<b>63</b>
5.1	Exercises . . . . .	68
<b>6</b>	<b>Primal-Dual Pairs</b>	<b>93</b>
6.1	Exercises . . . . .	100
<b>7</b>	<b>Analytical Geometry</b>	<b>125</b>
7.1	Points, Lines, Subspaces . . . . .	125
7.2	Polyhedra, Ideal Descriptions, Cones . . . . .	127
7.2.1	Faces, Valid Equations, Affine Hulls . . . . .	128
7.2.2	Facets, Minimal Complete Descriptions, Quasi-Uniqueness . . . . .	129
7.2.3	Asymptotic Cones and Extreme Rays . . . . .	130
7.2.4	Adjacency I, Extreme Rays of Polyhedra, Homogenization . . . . .	130
7.3	Point Sets, Affine Transformations, Minimal Generators . . . . .	131
7.3.1	Displaced Cones, Adjacency II, Images of Polyhedra . . . . .	132
7.3.2	Carathéodory, Minkowski, Weyl . . . . .	133
7.3.3	Minimal Generators, Canonical Generators, Quasi-Uniqueness . . . . .	133
7.4	Double Description Algorithms . . . . .	135
7.4.1	Correctness and Finiteness of the Algorithm . . . . .	136
7.4.2	Geometry, Euclidean Reduction, Analysis . . . . .	137
7.4.3	The Basis Algorithm and All-Integer Inversion . . . . .	138
7.4.4	An All-Integer Algorithm for Double Description . . . . .	139
7.5	Digital Sizes of Rational Polyhedra and Linear Optimization . . . . .	140
7.5.1	Facet Complexity, Vertex Complexity, Complexity of Inversion . . . . .	141
7.5.2	Polyhedra and Related Polytopes for Linear Optimization . . . . .	142
7.5.3	Feasibility, Binary Search, Linear Optimization . . . . .	142
7.5.4	Perturbation, Uniqueness, Separation . . . . .	144
7.6	Geometry and Complexity of Simplex Algorithms . . . . .	146
7.6.1	Pivot Column Choice, Simplex Paths, Big M Revisited . . . . .	147
7.6.2	Gaussian Elimination, Fill-In, Scaling . . . . .	148

7.6.3 Iterative Step I, Pivot Choice, Cholesky Factorization . . . . .	149
7.6.4 Cross Multiplication, Iterative Step II, Integer Factorization . . . . .	150
7.6.5 Division Free Gaussian Elimination and Cramer's Rule . . . . .	151
7.7 Circles, Spheres, Ellipsoids . . . . .	153
7.8 Exercises . . . . .	156
<b>8 Projective Algorithms</b>	<b>201</b>
8.1 A Basic Algorithm . . . . .	203
8.1.1 The Solution of the Approximate Problem . . . . .	203
8.1.2 Convergence of the Approximate Iterates . . . . .	205
8.1.3 Correctness, Finiteness, Initialization . . . . .	206
8.2 Analysis, Algebra, Geometry . . . . .	207
8.2.1 Solution to the Problem in the Original Space . . . . .	207
8.2.2 The Solution in the Transformed Space . . . . .	209
8.2.3 Geometric Interpretations and Properties . . . . .	211
8.2.4 Extending the Exact Solution and Proofs . . . . .	214
8.2.5 Examples of Projective Images . . . . .	215
8.3 The Cross Ratio . . . . .	215
8.4 Reflection on a Circle and Sandwiching . . . . .	218
8.4.1 The Iterative Step . . . . .	220
8.5 A Projective Algorithm . . . . .	221
8.6 Centers, Barriers, Newton Steps . . . . .	223
8.6.1 A Method of Centers . . . . .	224
8.6.2 The Logarithmic Barrier Function . . . . .	226
8.6.3 A Newtonian Algorithm . . . . .	228
8.7 Exercises . . . . .	230
<b>9 Ellipsoid Algorithms</b>	<b>263</b>
9.1 Matrix Norms, Approximate Inverses, Matrix Inequalities . . . . .	265
9.2 Ellipsoid "Halving" in Approximate Arithmetic . . . . .	266
9.3 Polynomial-Time Algorithms for Linear Programming . . . . .	269
9.4 Deep Cuts, Sliding Objective, Large Steps, Line Search . . . . .	272
9.4.1 Linear Programming the Ellipsoidal Way: Two Examples . . . . .	274
9.4.2 Correctness and Finiteness of the DCS Ellipsoid Algorithm . . . . .	277
9.5 Optimal Separators, Most Violated Separators, Separation . . . . .	278
9.6 $\varepsilon$ -Solidification of Flats, Polytopal Norms, Rounding . . . . .	280
9.6.1 Rational Rounding and Continued Fractions . . . . .	282
9.7 Optimization and Separation . . . . .	285
9.7.1 $\varepsilon$ -Optimal Sets and $\varepsilon$ -Optimal Solutions . . . . .	287
9.7.2 Finding Direction Vectors in the Asymptotic Cone . . . . .	287
9.7.3 A CCS Ellipsoid Algorithm . . . . .	288
9.7.4 Linear Optimization and Polyhedral Separation . . . . .	289
9.8 Exercises . . . . .	293

<b>10 Combinatorial Optimization: An Introduction</b>	<b>323</b>
10.1 The Berlin Airlift Model Revisited . . . . .	323
10.2 Complete Formulations and Their Implications . . . . .	327
10.3 Extremal Characterizations of Ideal Formulations . . . . .	331
10.4 Polyhedra with the Integrality Property . . . . .	334
10.5 Exercises . . . . .	336
<b>Appendices</b>	
<b>A Short-Term Financial Management</b>	<b>359</b>
A.1 Solution to the Cash Management Case . . . . .	362
<b>B Operations Management in a Refinery</b>	<b>371</b>
B.1 Steam Production in a Refinery . . . . .	371
B.2 The Optimization Problem . . . . .	374
B.3 Technological Constraints, Profits and Costs . . . . .	378
B.4 Formulation of the Problem . . . . .	380
B.5 Solution to the Refinery Case . . . . .	381
<b>C Automatized Production: PCBs and Ulysses' Problem</b>	<b>399</b>
C.1 Solutions to Ulysses' Problem . . . . .	411
<b>Bibliography</b>	<b>431</b>
<b>Index</b>	<b>445</b>

# 1. Introduction

Most exercises of this introductory chapter are posed in the form of “minicases” that we have used over many years in the classroom to familiarize newcomers to the field of linear optimization with the basic approach to linear modeling and problem solving. In our experience, students like to grapple and experiment with their own approaches to the simple problem solving situations captured by the following exercises. The analysis of the minicases that we present is, however, based on the material of later chapters and can be used to accompany the theoretical development of the subject throughout.

## 1.1 Minicases and Exercises

---

### \*Exercise 1.0 (Minicase I)

*Lilliputian Liquids Inc. (LLI) is engaged in the production and sale of two kinds of hard liquor. LLI purchases intermediate-stage products in bulk, purifies them by repeated distillation, mixes them, bottles the product under its own brand names and sells it to distributive channels. One product is a bourbon, the other one a whiskey. Sales of each product have always been independent of the other and market limits on sales have never been observed.*

*Labor is not a constraint on LLI. Production capacity, though, is inadequate to produce all that LLI might sell. The bourbon requires three machine hours per liter, but because of additional blending requirements the whiskey requires four hours of machine time per liter. A total capacity of 20,000 machine hours is available in the coming production period. Higher quality makes the direct operating costs of the bourbon \$3 per liter in contrast with the whiskey’s costs of \$2 dollars per liter. Funds available to finance direct costs are planned at \$4,000 for the coming production period. In addition, it is anticipated that 45% of bourbon and 30% of whiskey sales made during the production period are collected during the same period and that the cash proceeds will be available to finance operations. All direct costs have to be paid during the production period. The bourbon sells to the distributive channels for \$5 per liter and the whiskey for \$4.50 per liter.*

*Planning for company activities during the coming production period had led to disagreement among the members of LLI’s management. The production and marketing managers on one hand and the treasurer-controller on the other could not agree on the most desirable product mix and production volume to schedule, whereas the production manager and the treasurer-controller were unable to agree on a proposal to expend \$250 for repair of decrepit machinery currently lying idle. It had been estimated that 2,000 machine hours could be added to capacity for the coming production period by this expenditure, although it was anticipated that the machines would again be inoperable by the end of the current planning period. The treasurer-controller acknowledged the need for additional machine capacity, but argued that the scarcity of LLI’s working capital made it inadvisable to divert any cash from financing current production.*

- (i) *Formulate LLI’s problem as a linear program in two variables  $x_1$  and  $x_2$ .*

- (ii) Plot the set of “feasible” solutions, i.e., solutions that satisfy all inequalities of the formulation in the plane with coordinates  $x_1$  and  $x_2$ .
  - (iii) Plot the “isoprofit” line  $2x_1 + 2.5x_2 = 10,000$  in the plane. Are there any feasible product combinations that attain a profit of \$10,000? How do you find a product combination that maximizes profit? What is the optimal product mix?
  - (iv) Analyze the problem posed by the disagreement between the product manager and the treasurer-controller of LLI. What is your recommendation? How does the optimal product mix change if the proposal is accepted?
  - (v) How does your answer for part (iv) change if the proposal is to spend \$500 at a gain of 4,000 machine hours? What if LLI wants to produce at least 500 liters of bourbon to keep its bourbon brand “visible” in the market place?
  - (vi) Summarize your findings in an “executive summary” separate from your analysis.
- 

### **Executive Summary:**

- A.** The profit-optimal product mix for LLI's resource allocation problem is to produce

$$\boxed{2,857.14 \text{ liters of bourbon and } 2,857.14 \text{ liters of whiskey}}$$

in the upcoming production period. Given the current profit margins this should result in an optimal profit of  $\boxed{\$12,857.14}$  for LLI.

- B.** Spending \$250 on the repair of the currently unusable machinery with a concurrent gain of an additional 2,000 hours of machine time can be expected to yield an incremental profit of \$976.19, thus giving LLI a total profit of

$$\boxed{\$12,857.14 + \$976.19 = \$13,833.33}$$

for the upcoming production period. It is therefore recommend that LLI take appropriate actions. Due to the resulting changes in both working capital and machine hours, the correspondingly changed profit-optimal product mix for LLI is to produce 666.67 liters of bourbon and 5,000 liters of whiskey (if the repairs are carried out on time).

- C.** Spending \$500 on the repair of currently unusable machinery is not advisable since any amount over \$326 spent on repairs –under the assumption that machine time will increase by 8 machine hours for every dollar spent on repairs– will make working capital the bottleneck for profit-optimal production. To maintain visibility in the market place for the bourbon brand in the limits specified an expenditure of about \$269 will be required if this expenditure gains 2,152 additional machine hours. In this case the profit-optimal product mix consists of 500.19 liters of bourbon and 5,162.82 liters of whiskey with a total profit of \$13,907.52, of which \$1,050.38 are to be attributed to repairs.

**Solution to and Analysis of Minicase I:**

(i) To formulate LLI's problem denote by  $x_1$  the liters of bourbon and by  $x_2$  the liters of whiskey produced during the planning period. Since the market *absorbs* all that LLI can produce we have that the quantities sold equal the quantities produced. Hence there is no need to distinguish between "sales" and "production". To produce  $x_1$  liters of bourbon one needs  $3x_1$  machine hours and  $4x_2$  machine hours to produce  $x_2$  liters of whiskey, giving a total of  $3x_1 + 4x_2$  machine hours. Consequently, we have a *production capacity constraint*

$$3x_1 + 4x_2 \leq 20,000.$$

The direct cost of producing  $x_1$  liters of bourbon and  $x_2$  liters of whiskey is  $3x_1 + 2x_2$  which must be covered by the funds that are available for production. These consist of \$4,000 in cash plus the anticipated collections on bourbon sales of  $\$0.45(5x_1) = \$2.25x_1$  plus those on whiskey sales of  $\$0.30(4.50x_2) = \$1.35x_2$ , i.e.,

$$\begin{aligned} \text{cash available for production} &= \text{cash on hand} + \text{collections on accounts receivables} \\ &= \$4,000 + 2.25x_1 + 1.35x_2. \end{aligned}$$

Since all direct costs must be paid during the current period we get the inequality

$$3x_1 + 2x_2 \leq 4,000 + 2.25x_1 + 1.35x_2.$$

Simplifying we get a *working capital constraint*

$$0.75x_1 + 0.65x_2 \leq 4,000,$$

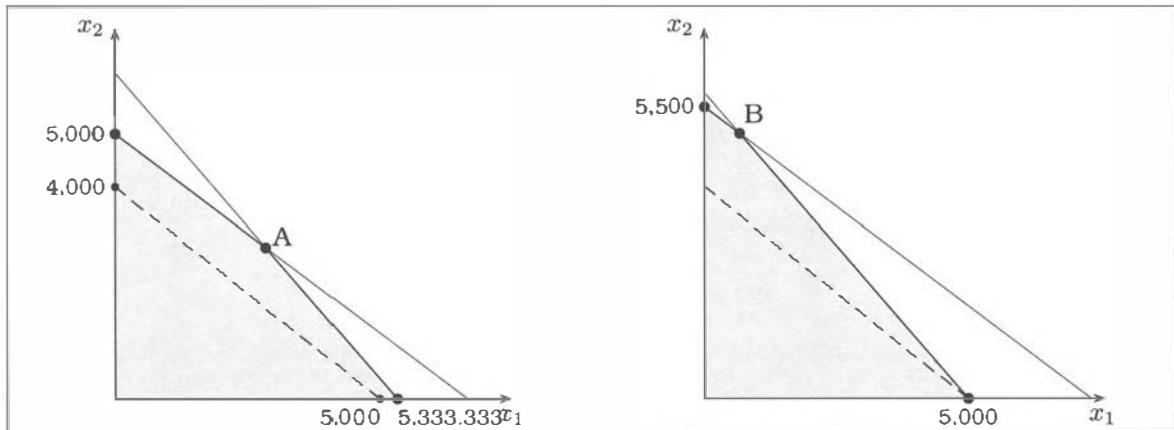
which makes sure that all direct costs are paid during the current period. Profit is revenue *minus* costs and thus we have a profit of  $\$5 - \$3 = \$2$  per liter of bourbon and of  $\$4.50 - \$2 = \$2.50$  per liter of whiskey. Including the nonnegativity constraints on the quantities of bourbon and whiskey produced during the current period LLI's decision problem is the following linear program:

$$\begin{array}{lll} \text{Maximize} & 2x_1 & + 2.5x_2 \\ \text{subject to} & 3x_1 & + 4x_2 \leq 20,000 \\ & 0.75x_1 & + 0.65x_2 \leq 4,000 \\ & x_1, x_2 & \geq 0. \end{array}$$

(ii) The "feasible region" is shown in the left part of Figure 1.1 as the shaded area.

(iii) The isoprofit line  $2x_1 + 2.5x_2 = 10,000$  is shown as a dashed line in the left part of Figure 1.1. All feasible points on the line, i.e., all points on the line segment between the points  $x_1 = 0, x_2 = 4000$  and  $x_1 = 5000, x_2 = 0$  are feasible product combinations that yield a profit of \$10,000. To get the optimal solution, we move the isoprofit line parallel to itself outwards (increasing profit) until we reach a position such that, if we move the line further, it does not "touch" the feasible region anymore. This is the case when the isoprofit line goes through the point A. Thus point A with coordinates  $x_1 = \frac{20,000}{7}, x_2 = \frac{20,000}{7}$  is the profit maximizing product combination with a profit of  $\frac{90000}{7} \approx \$12,857.14$ .

Input and solution including ranging analysis by LINDO are displayed next.



**Fig. 1.1.** Graphical solution to LLI's problem

```

! LLI's Problem
MAX 2x1 + 2.5x2
subject to
  3x1 + 4x2 < 20000 ! Production capacity
  0.75x1 + 0.65x2 < 4000 ! Working capital
                      ! All variables nonnegative

LP OPTIMUM FOUND AT STEP      2
      OBJECTIVE FUNCTION VALUE
      1)    12857.14
VARIABLE      VALUE          REDUCED COST
      X1    2857.142822    0.000000
      X2    2857.142822    0.000000

      ROW    SLACK OR SURPLUS      DUAL PRICES
      2)          0.000000        0.547619
      3)          0.000000        0.476190

NO. ITERATIONS=      2
RANGES IN WHICH THE BASIS IS UNCHANGED:
      OBJ COEFFICIENT RANGES
VARIABLE      CURRENT      ALLOWABLE      ALLOWABLE
                  COEF          INCREASE        DECREASE
      X1        2.000000     0.884615     0.125000
      X2        2.500000     0.166667     0.766667

      RIGHTHAND SIDE RANGES
      ROW      CURRENT      ALLOWABLE      ALLOWABLE
                  RHS          INCREASE        DECREASE
      2        20000.000000   4615.384277   3999.999756
      3        4000.000000    999.999939    750.000000

```

**(iv)** To analyze the disagreement between the production manager and the treasurer-controller of LLI we have several possibilities of which we discuss two.

The first is simple and direct, but not the most efficient way when applied to problems of larger scale. Spending \$250 to repair the idle machinery changes the right-hand side of the constraints as follows. The available capital becomes  $4,000 - 250 = \$3,750$ , while the available machine hours increase to  $20,000 + 2,000 = 22,000$ . These changes change the feasible region since the two lines corresponding to the constraints move parallel to themselves one outwards and the other inwards. The new feasible region is shown on the right part of Figure 1.1 above. The optimal product mix now is given by point B with coordinates  $x_1 = \frac{20,000}{7}$ ,  $x_2 = 5000$ . The optimal profit is  $\frac{41500}{3} \approx \$13,833.33$ . Since the expenditure of \$250 of available cash generates an increase of \$976.19 in profit, it is recommended to spend \$250 on the repair of the idle machinery.

The second way to analyze the disagreement uses the dual variables of LLI's linear program. Like before, reducing working capital by \$250 results in an increase of 2,000 machine hours in production capacity. Denote by  $b$  the right-hand side of LLI's problem and by  $g$  the vector of change in the right-side *per dollar spent* on the repair work, i.e.,

$$b = \begin{pmatrix} 20,000 \\ 4,000 \end{pmatrix}, \quad g = \begin{pmatrix} 8 \\ -1 \end{pmatrix}. \quad (1.1)$$

Thus we wish to analyze LLI's problem for the right-hand side  $b + \Delta g$  where  $\Delta = 250$ . We know that the optimal objective function value  $Z(\Delta) = Z(b + \Delta g)$  is given by

$$Z(\Delta) = c_B B^{-1}(b + \Delta g) = Z_B + \Delta(c_B B^{-1})g = Z_B + \Delta \sum_{i=1}^m y_i^* g_i, \quad (1.2)$$

where  $Z_B$  is the optimal profit for  $\Delta = 0$ ,  $y^* = c_B B^{-1}$  are the optimal values of the dual variables (displayed above) and  $g_i$  is the  $i$ -th component of the vector  $g$ . Thus we find from LINDO's output

$$Z(\Delta) \approx \$12,857.14 + \Delta(8 \times 0.547619 + (-1) \times 0.476190) = \$12,857.14 + 3.904762\Delta. \quad (1.3)$$

This quick calculation tells us that every dollar spent on repair produces "locally" a gain of \$3.904762 in (optimal) profit and thus for  $\Delta = 250$  we conclude from LINDO's output –like we did before– that the incremental profit we get from the repair is

$$3.904762 \times 250 \approx \$976.19.$$

The real question is whether or not this "local" analysis is correct for  $\Delta = 250$ . The ranging analysis (for changes of a *single* right-hand side value!) given by LINDO's output suggests that this may be true, but to answer the question we have to check whether or not

$$\overset{?}{x}_B(\Delta) = B^{-1}(b + \Delta g) = B^{-1}b + \Delta B^{-1}g \geq 0 \quad \text{for } \Delta = 250,$$

because *two* right-hand side values are changed simultaneously here. We know from the above that the optimal basis (for  $\Delta = 0$ ) contains the variables  $x_1$  and  $x_2$ , i.e., the optimal basis and its inverse (which we calculate directly) are given by

$$B = \begin{pmatrix} 3 & 4 \\ 0.75 & 0.65 \end{pmatrix}, \quad B^{-1} = \frac{1}{21} \begin{pmatrix} -13 & 80 \\ 15 & -60 \end{pmatrix}.$$

Now we are in the position to calculate  $\mathbf{x}_B(\Delta) = \mathbf{B}^{-1}(\mathbf{b} + \Delta\mathbf{g})$  exactly as

$$\begin{aligned}\mathbf{x}_B(\Delta) &= \mathbf{B}^{-1}(\mathbf{b} + \Delta\mathbf{g}) = \frac{1}{21} \begin{pmatrix} -13 & 80 \\ 15 & -60 \end{pmatrix} \left[ \begin{pmatrix} 20,000 \\ 4,000 \end{pmatrix} + \Delta \begin{pmatrix} 8 \\ -1 \end{pmatrix} \right] \\ &= \frac{1}{7} \begin{pmatrix} 20,000 \\ 20,000 \end{pmatrix} + \frac{1}{21} \begin{pmatrix} -184 \\ 180 \end{pmatrix} \Delta = \frac{1}{21} \begin{pmatrix} 60,000 - 184\Delta \\ 60,000 + 180\Delta \end{pmatrix} \stackrel{?}{\geq} \begin{pmatrix} 0 \\ 0 \end{pmatrix}.\end{aligned}$$

Reading the inequalities componentwise we find that

$$326 \frac{2}{23} = \frac{60,000}{184} \geq \Delta \geq -\frac{60,000}{180} = -333 \frac{1}{3}. \quad (1.4)$$

$\mathbf{x}_B(\Delta)$  has thus positive components for  $\Delta = 250$ . For any  $\Delta$  the changed solution is

$$\mathbf{x}_B(\Delta) = \begin{pmatrix} x_1(\Delta) \\ x_2(\Delta) \end{pmatrix} = \frac{1}{21} \begin{pmatrix} 60,000 - 184\Delta \\ 60,000 + 180\Delta \end{pmatrix} \stackrel{\Delta = 250}{=} \begin{pmatrix} \frac{2,000}{3} \\ 5,000 \end{pmatrix}, \quad (1.5)$$

which by (2) and using  $\mathbf{c}_B \mathbf{B}^{-1} = \begin{pmatrix} 2 & 2.50 \end{pmatrix} \mathbf{B}^{-1} = \begin{pmatrix} \frac{23}{42} & \frac{10}{21} \end{pmatrix}$  gives an optimal profit of

$$Z(\Delta) = Z_B + \Delta(\mathbf{c}_B \mathbf{B}^{-1})\mathbf{g} = Z_B + \Delta \begin{pmatrix} \frac{23}{42} & \frac{10}{21} \end{pmatrix} \begin{pmatrix} 8 \\ -1 \end{pmatrix} = Z_B + \frac{82}{21}\Delta \quad (1.6)$$

$$\stackrel{\Delta = 250}{=} \$\frac{90,000}{7} + \$976 \frac{4}{21} \approx \$13,833.33 \quad (1.7)$$

with an incremental profit of  $\$976 \frac{4}{21} \approx \$976.19$  for  $\Delta = 250$ , like we found above. We calculate  $\frac{82}{21} \approx 3.904762$  and thus our finding agrees with our quick calculation from (1.3) using the dual variable information provided by LINDO, which however was incomplete without the check on whether or not the local analysis remains valid for the range of  $\Delta$  considered here. Because it does, we now conclude correctly that spending \$250 on the repair of the idle machinery is profitable for LLI.

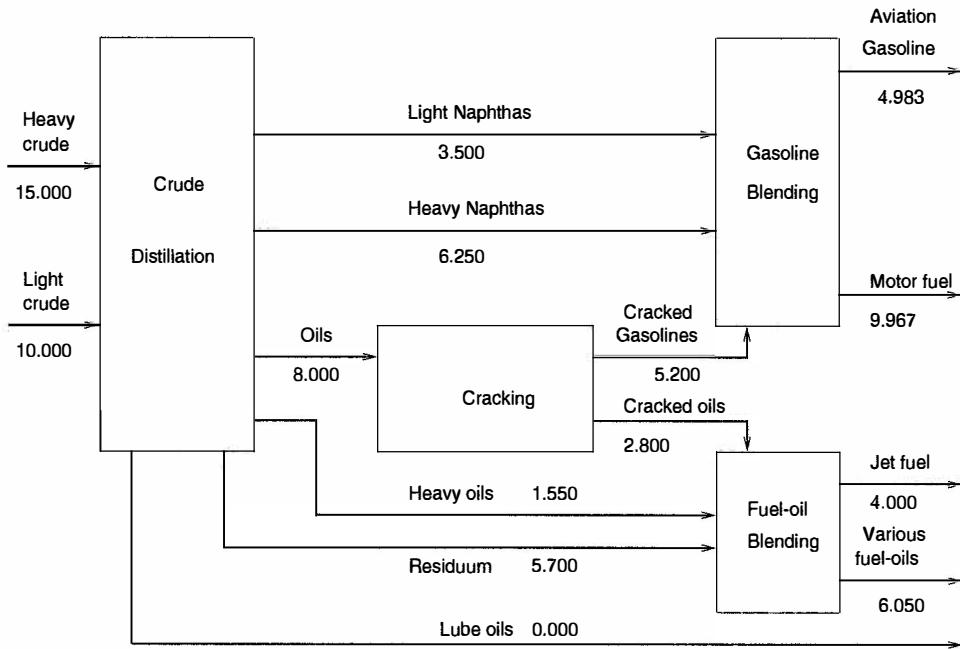
(v) If we gain 4,000 additional machine hours by spending \$500, then the *gain in machine time per dollar spent* equals 8 hours like in formula (1) above. So the analysis of part (iv) remains correct. Using the dual variable information provided by LINDO we find from (1.3) that the incremental profit from this decision equals  $\$3.904762 \times 500 \approx \$1,952.38$ . We would therefore conclude **-incorrectly** as we shall see– that the decision is highly profitable.

By our analysis in part (iv) we know that  $\Delta = 500$  falls outside of the “permissible” range (1.4). This means that one or more of the basic variables, in this case the variable  $x_1$  modeling the bourbon production, becomes negative. This happens when  $\Delta > 326 \frac{2}{23}$  and the optimal basis has to be changed. Consequently, either the slack of the production capacity constraint or the slack of the working capital constraint has to enter the basic set, because in our case these are the only possibilities. Since working capital is reduced by spending more dollars on repairs, it can only be the slack of the production capacity constraint. Thus spending \$500 on repair, LLI creates an over-capacity of production that cannot be utilized. From formula (1.6) we find

$$Z(\Delta = 326 \frac{2}{23}) = \$\frac{90,000}{7} + \$\frac{615,000}{483} \approx \$12,857.14 + \$1,273.29 = \$14,130.43$$

and from formula (1.5) we get the optimal product mix

$$\mathbf{x}_B(\Delta) = \begin{pmatrix} x_1(\Delta) \\ x_2(\Delta) \end{pmatrix} \stackrel{\Delta = 326 \frac{2}{23}}{=} \begin{pmatrix} 0 \\ \frac{130,000}{23} \end{pmatrix} \approx \begin{pmatrix} 0 \\ 5,652.17 \end{pmatrix}.$$



**Fig. 1.2.** Optimal solution for the simplified refinery problem

If we spend more than  $\$326 \frac{2}{23}$  on the repair of idle machinery, bourbon will not be produced at all. Because of the continued reduction of working capital, profit will decrease from  $\$14,130.43$  at the rate of  $\frac{50}{13} \approx 3.846\ldots$  per additional dollar spent (as you verify yourself), until the linear program becomes infeasible. Thus the “local” analysis based on the dual variable information is incorrect, because  $\Delta = 500$  is too large a change in this case.

Formula (1.5) shows that the increased production capacity due to the repair of idle machinery leads to a profit-optimal product mix where the production of bourbon is slowly reduced to zero. If LLI wants to produce and sell at least 500 liters of bourbon, then from formula (5) we get the inequality

$$\frac{1}{21}(60,000 - 184\Delta) \geq 500 \quad \text{and thus} \quad \Delta \leq \frac{12,375}{46} \approx 269.02.$$

Thus spending about \$269 on the repair of idle machinery with a gain of about 2,152 hours of additional machine time –if feasible– is the best LLI can do to increase its profit and maintain its bourbon “visible” in the market place.

---

### \*Exercise 1.1

Solve the simplified refinery example by an interactive LP solver such as CPLEX or LINDO and interpret the optimal LP solution on Figure 1.2 of the book.

---

The LP data in CPLEX lp input format is as follows:

```

Maximize
  obj: 6.5 x10 + 4.6 x11 + 3.5 x12 + 2.5 x13 + 0.8 x14 - 1.5 x1 - 1.7 x2
    - 0.4 x3 - 0.4 x4 - 0.9 x5 - 0.3 x6 - 0.3 x7 - 0.4 x8 - 0.3 x9
Subject To
  c1: x2 <= 10
  c2: x1 + x2 <= 25
  c3: x5 <= 8
  c4: 0.12 x1 + 0.17 x2 - x3 = 0
  c5: 0.23 x1 + 0.28 x2 - x4 = 0
  c6: 0.41 x1 + 0.34 x2 - x5 - x6 = 0
  c7: - x14 + 0.24 x1 + 0.21 x2 - x7 = 0
  c8: 0.65 x5 - x8 = 0
  c9: 0.35 x5 - x9 = 0
  c10: x10 + x11 - x3 - x4 - x8 = 0
  c11: x12 + x13 - x6 - x7 - x9 = 0
  c12: x10 - 1.5 x3 <= 0
  c13: x10 - 1.2 x3 - 0.3 x4 <= 0
  c14: x10 - 0.5 x11 <= 0
  c15: x12 <= 4
Bounds
  All variables are >= 0.
End

```

Running interactively CPLEX we get the following solution

Variable Name	Solution Value
x10	4.983333
x11	9.966667
x12	4.000000
x13	6.050000
x1	15.000000
x2	10.000000
x3	3.500000
x4	6.250000
x5	8.000000
x6	1.550000
x7	5.700000
x8	5.200000
x9	2.800000

All other variables in the range 1-14 are zero.

In Figure 1.2 we show the solution on the simplified flow chart. The number on each arc shows the flow in the respective arc.

\*Exercise 1.2 (Minicase II) (Source: unknown)

*The Washington Apple Canning Company (WACCO) has purchased 6,000,000 pounds of apples at a total cost of \$300,000 of which 20% was rated "A" quality and the remaining 80% was rated "B". WACCO's executives will soon meet to determine how to allocate the apples to three products, apple juice, apple sauce, and apple jelly, in a profit-optimal manner.*

*The marketing department has estimated that WACCO could sell all the apple juice they could make, but that the demand for sauce and jelly was limited. Their demand forecasts are shown in Table 1.1. Table 1.2 shows the figures on price, cost, and profit per case of the apple products that were obtained from WACCO's accounting department. These figures include the cost of apples which was 5 cents per pound delivered to the cannery.*

*The production manager indicated that it was impossible to produce only apple juice since too small a proportion of the crop was "A" quality. WACCO uses a numerical scale from 0 to 10 to rate the quality of both raw produce and prepared products, with the higher number representing higher quality. Grade A apples averaged 8 points per pound whereas grade B apples averaged 4 points per pound. The minimum acceptable average quality requirement for apple juice had been set at 7, for apple sauce at 5, and for apple jelly at 4. You have been asked to come up not only with a recommendation for WACCO's executives, but also with an evaluation of the purchase of additional grade A apples as detailed below.*

**Table 1.1.** Selling price and demand forecast per case

Product	Selling price per case	Demand in cases	Pounds per case
Apple Juice	\$ 3.18	1,600,000	18
Apple Sauce	\$ 3.45	100,000	20
Apple Jelly	\$ 3.05	160,000	25

**Table 1.2.** Product profits per case

Product	Apple juice	Apple sauce	Apple jelly
Selling Price	\$ 3.18	\$ 3.45	\$ 3.05
Variable Costs			
Direct Labor	\$ 1.00	\$ 1.13	\$ 0.48
Variable Selling	\$ 0.25	\$ 0.60	\$ 0.22
Packaging Material	\$ 0.45	\$ 0.40	\$ 0.50
Apple Cost	\$ 0.90	\$ 1.00	\$ 1.25
Total variable cost	\$ 2.60	\$ 3.13	\$ 2.45
Profit	\$ 0.58	\$ 0.32	\$ 0.60

(i) Let

$x_1$  be the pounds of "A" apples allocated to the production of apple juice,  
 $y_1$  be the pounds of "B" apples allocated to the production of apple juice,  
 $x_2$  be the pounds of "A" apples allocated to the production of apple sauce,  
 $y_2$  be the pounds of "B" apples allocated to the production of apple sauce,  
 $x_3$  be the pounds of "A" apples allocated to the production of apple jelly,  
 $y_3$  be the pounds of "B" apples allocated to the production of apple jelly.

Formulate the problem of finding a profit-optimal allocation of grade A and B apples to the three apple products of WACCO as a linear programming model in terms of the above six variables.

- (ii) Compute the profit-optimal amount of grade A and B apples to be allocated to apple juice, sauce and jelly production using an interactive LP solver such as CPLEX or LINDO. What is the optimal profit that WACCO can obtain?
  - (iii) WACCO has been offered an additional 160,000 pounds of grade A apples at 9.5 cents per pound and WACCO's purchasing manager has already acquired these additional grade A apples. How can the formulation of part (i) be altered to take this into consideration? What is the optimal product allocation and profit? Is the purchasing manager's decision profitable for WACCO?
  - (iv) WACCO has been offered an additional 2,000,000 pounds of grade A apples at an unspecified price that will be 8 cents or more per pound. How do you have to change your formulation of part (i) to incorporate the decision to buy or not additional grade A apples? If WACCO can buy these grade A apples at 8 cents per pound how many pounds should it buy? What are the optimal profit and product allocations? Answer the same questions when WACCO must pay 8.6 cents per pound. At what price per pound does it become unprofitable for WACCO to buy any additional grade A apples?
  - (v) Summarize your answers in an "executive summary" separate from your analysis.
- 

### Executive Summary:

A. Given the current demand forecasts and data, WACCO can realize an optimal profit of

\$ 150,708

by allocating the available 6,000,000 pounds of grade A and grade B apples to its three products as follows:

- 1,050,000 pounds of grade A and 350,000 pounds of grade B apples to juice production.
- 150,000 pounds of grade A and 450,000 pounds of grade B apples to sauce production.
- 4,000,000 pounds of grade B apples to jelly production.

**B.** Our computer analysis shows that given its current data configuration WACCO should pay

at most 9.033 cents per pound

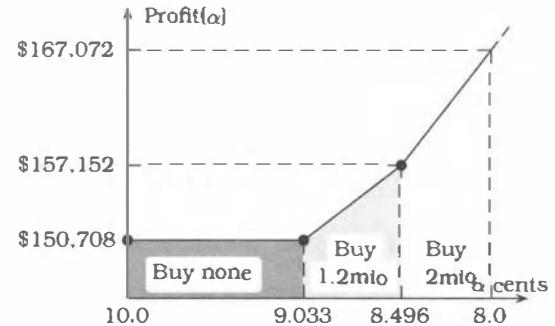
for the purchase of additional grade A apples. The scenario under point (iii) where WACCO's purchasing manager has already acquired 160,000 grade A apples at 9.5 cents per pound is regrettable, since it reduces WACCO's attainable profit under point A by about \$747. This is the case only if WACCO reallocates its resource availabilities in a profit-optimal manner:

- 1,230,000 pounds of grade A and 410,000 pounds of grade B apples to juice production,
- 130,000 pounds of grade A and 390,000 pounds of grade B apples to sauce production,
- 4,000,000 pounds of grade B apples to jelly production,

because otherwise WACCO's loss must be expected to be greater than \$747. An exact analysis (avoiding the numerical round-off errors of the computer calculation, which in this case are acceptably small) shows that WACCO can actually sustain profitably a slightly higher cost of 9.075 cents per pound of additional grade A apples. A formula to calculate the resulting optimal profit and the profit-optimal product allocation for additional grade A apple amounts of up to 1,200,000 is given in part(iii) of our analysis below.

**C.** Our analysis of the purchase of up to 2,000,000 pounds of grade A apples is summarized in the figure. If the cost per pound for grade A apples is less than or equal to 8.496 cents, then WACCO should buy the entire lot of 2,000,000 pounds. The optimal product allocation is to allocate 3,200,000 pounds of grade A and 1,066,667 pounds of grade B apples to juice production, 3,733,333 pounds of grade B apples to jelly production. The optimal profit is calculated by the formula  $\$157,152 + 20,000 \times (8.496 - \alpha)$ , where  $\alpha$  is the cost per pound in cents. If  $8.496 < \alpha < 9.033$ , WACCO should buy 1,200,000 pounds and allocate as follows:

2,400,000 pounds of grade A apples and 800,000 pounds of grade B apples to juice production, 4,000,000 pounds of grade B apples to jelly production with an optimal profit of  $\$150,708 + 12,000 \times (9.033 - \alpha)$ . Otherwise  $\alpha > 9.033$  and WACCO should buy no additional grade A apples at all and apply the product allocation of part A.



### Solution to and Analysis of Minicase II:

**(i)** Since 20% of the 6,000,000 pounds are grade A apples, WACCO has 1,200,000 pounds of grade A and 4,800,000 pounds of grade B apples on its premises. We thus get the constraints

$$x_1 + x_2 + x_3 \leq 1,200,000 \text{ and } y_1 + y_2 + y_3 \leq 4,800,000$$

for the product allocation. Since the demand is limited, see Table 1.1, we cannot allocate more than  $18 \times 1,600,000 = 28,800,000$  pounds of apples to juice production, no more than  $20 \times 100,000 =$

2,000,000 pounds to sauce production and no more than  $25 \times 160,000 = 4,000,000$  pounds to jelly production which gives the constraints

$$x_1 + y_1 \leq 28,800,000, \quad x_2 + y_2 \leq 2,000,000, \quad x_3 + y_3 \leq 4,000,000.$$

In addition we need to formulate the minimum acceptable average quality requirement for the three products. From WACCO's scoring method we get

$$\frac{8x_1 + 4y_1}{x_1 + y_1} \geq 7, \quad \frac{8x_2 + 4y_2}{x_2 + y_2} \geq 5, \quad \frac{8x_3 + 4y_3}{x_3 + y_3} \geq 4,$$

which gives the following linear constraints

$$x_1 - 3y_1 \geq 0, \quad 3x_2 - y_2 \geq 0, \quad 4x_3 \geq 0,$$

the last one of which is implied by the fact that we want nonnegative quantities.

To formulate the objective function, we note that the six million pounds of apples **have been acquired already** by WACCO. Thus their variable cost (of 5 cents/pound) does not enter into the allocation decision, but their total acquisition cost of \$300,000 must be taken in consideration to determine the profit (or loss) of the product allocation. Thus from Table 1.2 we know that we get  $\$0.58 + \$0.90 = \$1.48$  per case of juice,  $\$0.32 + \$1.00 = \$1.32$  per case of sauce and  $\$0.60 + \$1.25 = \$1.85$  per case of jelly. From Table 1.1 we know how to convert cases into pounds and vice versa. Approximating  $1.48/18 \approx 0.08222$  to five digits we thus get the objective function

$$\begin{aligned} \text{Maximize} \quad & \frac{1.48}{18}(x_1 + y_1) + \frac{1.32}{20}(x_2 + y_2) + \frac{1.85}{25}(x_3 + y_3) - 300,000 \\ & \approx 0.08222x_1 + 0.08222y_2 + 0.066x_2 + 0.066y_2 + 0.074x_3 + 0.074y_3 - 300,000. \end{aligned}$$

(ii) In CPLEX's lp format the above linear program reads as follows where we do not have included the constant 300,000 in the objective function, because CPLEX does not accept it.

```
Problem name: WACCO.lp
Maximize
  obj: 0.08222 x1 + 0.08222 y1 + 0.066 x2 + 0.066 y2 + 0.074 x3 + 0.074 y3
Subject To
  c1: x1 + x2 + x3 <= 1200000
  c2: y1 + y2 + y3 <= 4800000
  c3: x1 + y1 <= 28800000
  c4: x2 + y2 <= 2000000
  c5: x3 + y3 <= 4000000
  c6: x1 - 3 y1 >= 0
  c7: 3 x2 - y2 >= 0
End
```

Solving the problem we find the following solution:

```
Primal - Optimal: Objective = 4.5070800000e+05
Solution time = 0.00 sec. Iterations = 5 (0)
Variable Name      Solution Value
x1                  1050000.00000
```

```

y1           350000.000000
x2           150000.000000
y2           450000.000000
y3           4000000.000000

```

All other variables in the range 1-6 are zero.

The objective function value is  $\$4.50708 \times 10^5 = \$450,708$ . Thus subtracting out \$300,000 for the cost of the apples, WACCO realizes a net profit of **\$150,708**. WACCO should allocate

- 1,050,000 pounds of grade A and 350,000 pounds of grade B apples to juice production,
- 150,000 pounds of grade A and 450,000 pounds of grade B apples to sauce production,
- 4,000,000 pounds of grade B apples to produce jelly.

Input for and solution including ranging analysis for the right-hand side from LINDO follow.

```

! WACCO's problem
max 0.08222x1 + 0.08222y1 + 0.066x2 + 0.066y2 + 0.074x3 + 0.074y3
subject to
x1 + x2 + x3 < 1200000 ! Grade A~availability
y1 + y2 + y3 < 4800000 ! Grade B availability
x1 + y1 < 28800000 ! Sales limitation: juice
x2 + y2 < 2000000 ! Sales limitation: sauce
x3 + y3 < 4000000 ! Sales limitation: jelly
x1 - 3y1 > 0          ! Minimum acceptable quality: juice
3x2 - y2 > 0          ! Minimum acceptable quality: sauce
                        ! All variables are nonnegative

```

LP OPTIMUM FOUND AT STEP 5

OBJECTIVE FUNCTION VALUE

1) 450708.0

VARIABLE	VALUE	REDUCED COST
X1	1050000.000000	0.000000
Y1	350000.000000	0.000000
X2	150000.000000	0.000000
Y2	450000.000000	0.000000
X3	0.000000	0.032440
Y3	4000000.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.090330
3)	0.000000	0.057890
4)	27400000.000000	0.000000
5)	1400000.000000	0.000000
6)	0.000000	0.016110
7)	0.000000	-0.008110
8)	0.000000	-0.008110

RANGES IN WHICH THE BASIS IS UNCHANGED:	RIGHTHAND SIDE RANGES
-----------------------------------------	-----------------------

ROW	CURRENT RHS	ALLOWABLE INCREASE	ALLOWABLE DECREASE
2	1200000.000000	1200000.000000	933333.312500
3	4800000.000000	933333.312500	400000.000000
4	28800000.000000	INFINITY	27400000.000000
5	2000000.000000	INFINITY	1400000.000000
6	4000000.000000	400000.000000	933333.312500
7	0.000000	933333.312500	1200000.000000
8	0.000000	2800000.000000	400000.000000

(iii) Since the purchasing manager **has already acquired** the additional 160,000 pounds of grade A apples, the acquisition itself is **not** part of the decision process. Thus all we have to do is to replace the availability of 1,200,000 pounds of grade A apples by  $1,200,000 + 160,000 = 1,360,000$  pounds of grade A apples. The cost of apples now is  $300,000 + 0.095 \times 160,000 = \$315,200$ . In a simple analysis we can thus run the problem again with the changed data. We find an optimal profit (rounded to dollars) of  $\$465,161 - \$315,200 = \$149,961$ . This shows that the acquisition makes WACCO loose **\$747** by comparison to the previous scenario of point (ii), even if a profit-optimal product allocation

- 1,230,000 pounds of grade A and 410,000 pounds of grade B apples to juice production,
- 130,000 pounds of grade A and 390,000 pounds of grade B apples to sauce production,
- 4,000,000 pounds of grade B apples to jelly production,

is followed by WACCO. Note that if WACCO decides to **not use at all** the 160,000 pounds of apples in its production decisions, that it then loses all of the \$15,200 in acquisition costs. By readjusting its product allocation in a best possible manner this loss is reduced to merely **\$747**. We can avoid the rerunning of the problem if we want to analyze the profitability of the purchasing manager's decision. From the solution report and the ranging analysis given by LINDO's output we see that the grade A apple availability can be increased by 1,200,000 pounds and/or decreased by 933,333.3125 pounds without the necessity of changing the optimal basis to WACCO's problem. Denote by  $\Delta$  the change in grade A apple availability. Thus  $\Delta = 160,000$  falls into the range where the profit-optimal basis does not change. Like in the analysis of Minicase I we have that the optimal profit changes according to the value of the optimal dual variable of the grade A availability constraint, i.e., from the computer solution report under part (ii) we know that

$$\begin{aligned} Z(\Delta) &= c_B B^{-1}(b + \Delta u_1) - 300,000 - \alpha\Delta \\ &= 150,708 + (0.09033 - \alpha)\Delta \text{ for all } \Delta \text{ with } -933,333.3125 \leq \Delta \leq 1,200,000, \end{aligned}$$

where  $\alpha$  is the cost per pound of grade A apples in **dollars** and  $u_1 \in \mathbb{R}^7$  is the first unit vector according to the first of the seven constraints of WACCO's linear program. Remember that we have left out the cost per pound of the apples in the calculation of the objective function coefficients of the linear program. We get

$$(0.09033 - 0.095)\Delta = -0.00467\Delta \stackrel{\Delta = 160,000}{=} -\$747,$$

which shows why WACCO is loosing money on the purchase manager's acquisition. On the other hand, if grade A apples can be acquired at less than 9.033 cents per pound, then WACCO's

optimal profit goes up by the difference multiplied by the amount  $\Delta$  acquired. The optimal product allocation must, of course, be adjusted according to the formula  $B^{-1}(\mathbf{b} + \Delta \mathbf{u}_1)$  given below.

From the computer calculations we know that the variables  $x_1, y_1, x_2, y_2, y_3$  and the slack variables of the constraints c3 and c4 (in the listing of the CPLEX input) are in the optimal basis. Consequently, the optimal basis and its inverse (which we calculate directly) are given by

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -3 & 3 & -1 & \end{pmatrix}, \quad B^{-1} = \frac{1}{8} \begin{pmatrix} 9 & -3 & 3 & -1 & -3 \\ 3 & -1 & 1 & -3 & -1 \\ -1 & 3 & -3 & 1 & 3 \\ -3 & 9 & -9 & 3 & 1 \\ & & 8 & & \\ -12 & 4 & 8 & -4 & 4 & 4 \\ 4 & -12 & 8 & 12 & -4 & -4 \end{pmatrix}.$$

Now we can verify the above analysis. Using the *approximate* calculation that  $1.48/18 \approx 0.08222$  we get precisely the above expression for  $Z(\Delta)$ . By an *exact* calculation with  $1.48/18$  we find

$$\begin{aligned} Z(\Delta) &= \mathbf{c}_B B^{-1}(\mathbf{b} + \Delta \mathbf{u}_1) - 300,000 - \alpha \Delta \\ &= 150,711 \frac{1}{9} + (0.09075 - \alpha) \Delta \text{ for all } \Delta \text{ with } -933,333 \frac{1}{3} \leq \Delta \leq 1,200,000, \end{aligned}$$

which shows that due to the round-off error in the calculation of  $1.48/18$  we commit a small, but tolerable(?) numerical error. WACCO can indeed sustain the slightly higher cost of 9.075 cents per pound of grade A apples profitably. Likewise we can compute now the profit-optimal product allocation for any amount  $\Delta$  of grade A apples in the same range as above from the formula  $B^{-1}(\mathbf{b} + \Delta \mathbf{u}_1)$ , which applies also for  $\Delta = 160,000$ :

- $1,050,000 + \frac{9}{8}\Delta$  pounds of grade A and  $350,000 + \frac{3}{8}\Delta$  pounds of grade B apples to juice production,
- $150,000 - \frac{1}{8}\Delta$  pounds of grade A and  $450,000 - \frac{3}{8}\Delta$  pounds of grade B apples to sauce production,
- 4,000,000 pounds of grade B apples to jelly production.

**(iv)** To analyze the purchase of additional grade A apples we make the acquisition decision part of the linear programming formulation. We regard the amount of grade A apples offered at a certain cost per pound as the maximum possible amount to be purchased, i.e., in our case we get at most 2,000,000 pounds. We then determine the actual amount that WACCO buys at that price from our analysis. This can be modeled in different ways: In the first approach, we introduce new variables  $z_1, z_2, z_3$  to reflect the purchase and the allocation of additional grade A apples to the three products. In the second approach, we model the decision by a single new variable.

In the first approach, the first two constraints of the formulation of part(i) remain the same and we get a new (third) constraint

$$z_1 + z_2 + z_3 \leq 2,000,000$$

from the limited availability of the additional grade A apples. Constraints (c2), (c3) and (c4) of the formulation of part(i) (see the CPLEX input) become the fourth, fifth and sixth constraint of the enlarged model

$$x_1 + y_1 + z_1 \leq 28,800,000, \quad x_2 + y_2 + z_2 \leq 2,000,000, \quad x_3 + y_3 + z_3 \leq 4,000,000.$$

The constraints for the minimum acceptable average quality requirement become

$$x_1 - 3y_1 + z_1 \geq 0, \quad 3x_2 - y_2 + 3z_2 \geq 0, \quad 4x_3 + 4z_3 \geq 0,$$

where the last one is again implied by the nonnegativity constraints. We thus have eight constraints and nine variables in the enlarged model. Suppose now that we pay  $100\alpha$  cents or  $\alpha$  dollars per pound of additional grade A apples. Then the objective function becomes

$$\begin{aligned} \text{Maximize} \quad & 0.08222x_1 + 0.08222y_2 + 0.066x_2 + 0.066y_2 + 0.074x_3 + 0.074y_3 \\ & + (0.08222 - \alpha)z_1 + (0.066 - \alpha)z_2 + (0.074 - \alpha)z_3 - 300,000. \end{aligned}$$

In general we would have to model some constraints that make sure that the existing grade A apples get allocated first, i.e., before purchasing new grade A apples we must make sure that the existing stock is utilized in the product allocation decisions. However, we think of  $\alpha$  as being *positive*. Then such constraints are automatically satisfied because the  $x_i$  and the  $z_i$  variables have the same coefficients in the constraint set, except for the objective function where the coefficients of the  $x_i$  are bigger than those of the  $z_i$ . Consequently, the sense of optimization –maximization– takes care of these considerations automatically.

The determination of the maximum cost per pound of grade A apples that can profitably be sustained by WACCO can be carried out in different ways. We can e.g. use a binary search procedure (“interval halving”) on the possible values of  $\alpha$  as follows. We start with  $\alpha = 0.085$  which is profitable with an incremental profit of \$6,396.80 and a purchase of 1,200,000 pounds of grade A apples, i.e., we do not buy the whole lot at that price. Running the problem with  $\alpha = 0.95$  we find (like in part(iii)) that buying apples is unprofitable. Indeed, the optimal solution to this problem consists of simply setting  $z_1 = z_2 = z_3 = 0$  and taking the solution from part(ii), which is (trivially) feasible to the enlarged problem. So the incremental profit from the available grade A apples is zero and WACCO decides to forgo the offer completely. Halving the interval given by 0.085 and 0.095 (the so-called “interval of uncertainty”) we run the problem with the new trial value of  $\alpha = 0.09$ , which turns out to be profitable. Then we use  $\alpha = 0.0925$  and so forth. This procedure converges rapidly to the value  $\alpha = 0.09033$  established in the first part of our analysis, but it does not give the optimal product allocations when the purchasing cost  $\alpha$  for grade A apples vary.

The procedure based on binary search is a convenient way for a “lazy” analysis of part of the proposition that WACCO faces: we replace analysis by a reasonable number of computer runs and let the computer do the work. What the proposition really calls for is a parametric analysis of the objective function of the three new variables  $z_1$ ,  $z_2$ , and  $z_3$ .

To carry out the parametric analysis we run the enlarged problem for e.g.  $\alpha = 0.091$ .

```

! WACCO's changed problem alpha=0.091
max 0.08222x1 + 0.08222y1 + 0.066x2 + 0.066y2 + 0.074x3 + 0.074y3
      - 0.00878z1 - 0.025z2 - 0.017z3
subject to
x1 + x2 + x3 < 1200000 ! Grade A~availability
y1 + y2 + y3 < 4800000 ! Grade B availability
z1 + z2 + z3 < 2000000 ! Additional Grade A~apples
x1 + y1 + z1 < 28800000 ! Sales limitation: juice

```

```

x2 + y2 + z2 < 2000000 ! Sales limitation: sauce
x3 + y3 + z3 < 4000000 ! Sales limitation: jelly
x1 - 3y1+ z1 >0          ! Minimum acceptable quality: juice
3x2 - y2+3z2 >0          ! Minimum acceptable quality: sauce
                           ! All variables are nonnegative

LP OPTIMUM FOUND AT STEP      6
OBJECTIVE FUNCTION VALUE
   1)    450708.0
VARIABLE      VALUE      REDUCED COST
   X1    1050000.00000      0.000000
   Y1    350000.00000      0.000000
   X2    150000.00000      0.000000
   Y2    450000.00000      0.000000
   X3      0.000000      0.032440
   Y3    4000000.00000     0.000000
   Z1      0.000000      0.000670
   Z2      0.000000      0.000670
   Z3      0.000000      0.033110
ROW  SLACK OR SURPLUS      DUAL PRICES
   2)      0.000000      0.090330
   3)      0.000000      0.057890
   4)    2000000.00000     0.000000
   5)  27400000.00000     0.000000
   6)  1400000.00000      0.000000
   7)      0.000000      0.016110
   8)      0.000000     -0.008110
   9)      0.000000     -0.008110

```

Since  $9.1 > 9.033$  we see that WACCO's optimal decision is to buy no additional grade A apples at all. From the solution report we find that the optimal primal and dual solutions are both *nondegenerate*. Moreover, all of the new variables  $z_1$ ,  $z_2$ , and  $z_3$  are nonbasic. In the following we think of the variables  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$ ,  $x_3$ ,  $y_3$ ,  $z_1$ ,  $z_2$ ,  $z_3$  and the logicals of the eight constraints as being indexed *sequentially*  $1, 2, \dots, 17$ .

Denote by  $c \in \mathbb{R}^{17}$  the profit vector (inclusive of logicals) that results from setting  $\alpha = 0.091$ . Let  $d \in \mathbb{R}^{17}$  be the vector of change, i.e.,

$$d = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$$

and thus the objective function of the parametric problem is  $c + \delta d$ , where  $\delta$  is initially zero. The objective function coefficients we want to change are (in the sequential indexing)

$$c_7 = -0.0878 + \delta, \ c_8 = -0.025 + \delta, \ c_9 = -0.017 + \delta.$$

From the computer output and using LINDO's sign convention (nonnegative reduced cost for optimality, i.e., LINDO *minimizes  $-cx$* ) we find that the reduced cost of  $z_1$ ,  $z_2$ , and  $z_3$  equal

$$\bar{c}_7 = 0.00067 - \delta, \ \bar{c}_8 = 0.00067 - \delta, \ \bar{c}_9 = 0.03311 - \delta.$$

Thus the basis remains optimal for all  $\delta \leq 0.00067$ , but we loose optimality for  $\delta > 0.00067$ , i.e., when the cost of grade A apples drops below  $9.033 = 9.1 - 0.067$  cents per pound. At this point we need to pivot. The basis changes and e.g. variable  $z_1$  enters and  $y_2$  leaves (there are ties!). The basis that we will study has variables  $x_1, y_1, x_2, y_3, z_1$ , and the slacks of constraints 3, 4 and 5 of the enlarged problem basic. The basis and its inverse (computed directly) are as follows.

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 & 1 \end{pmatrix}, \quad B^{-1} = \frac{1}{3} \begin{pmatrix} 3 & & & & -1 \\ & 3 & & -3 & \\ & & 3 & & 1 \\ -3 & 9 & & -9 & 3 & 1 \\ 3 & -9 & 3 & & 9 & -3 & -1 \\ -3 & -3 & 3 & 3 & 3 & 1 & \\ & & & & 3 & & -1 \end{pmatrix}.$$

We calculate the solution given by the new basis to be (in million pounds)  $x_1 = 1.2, y_1 = 0.8, x_2 = 0, y_3 = 4.0, z_1 = 1.2$ , and the slacks are set correspondingly. To find the range for which the new basis is optimal we reset the objective function coefficients of  $z_1, z_2$ , and  $z_3$  such that optimality is again exhibited for  $\delta = 0$ , i.e., we write  $\alpha = 0.09033 - \delta$  and thus

$$c_7 = -0.00811 + \delta, \quad c_8 = -0.02433 + \delta, \quad c_9 = -0.01633 + \delta.$$

Since the variable  $z_1$  is now basic, we denote by  $c_B(\delta)$  the components of the objective function corresponding to the variables in the basis. We compute

$$\begin{aligned} c_B(\delta)B^{-1} &= (0.08222, 0.08222, 0.066, 0.074, -0.00811 + \delta, 0, 0, 0) B^{-1} \\ &= (0.09033 - \delta, 0.05789 + 3\delta, 0, 0, 0, 0.01611 - 3\delta, -0.00811 + \delta, -0.00811 + \frac{1}{3}\delta). \end{aligned}$$

The computation of the reduced cost for the structural nonbasic variables  $y_2, x_3, z_2$ , and  $z_3$  yields (in the sequential indexing and LINDO's sign convention for optimality)

$$\bar{c}_4 = \frac{8}{3}\delta, \quad \bar{c}_5 = 0.03244 - 4\delta, \quad \bar{c}_8 = 0, \quad \bar{c}_9 = 0.03244 - 4\delta.$$

It follows that the new basis is optimal for all  $\delta$  in the range

$$0 \leq \delta \leq 0.00537.$$

The new "breakpoint" is consequently  $\alpha = 0.09033 - 0.00537 = 0.08496$ . At this point the basis changes. Consider the new basis with basic variables  $x_1, y_1, y_3, z_1$ , and the slacks corresponding to constraints 4, 5, 6, and 8 of the enlarged model. The basis and its inverse (computed directly) are

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ & & & -1 \end{pmatrix}, \quad B^{-1} = \frac{1}{3} \begin{pmatrix} 3 & & & & -1 \\ 1 & 1 & 1 & & 1 \\ -1 & 3 & -1 & 3 & \\ & & -1 & 3 & 1 \\ -4 & & & 3 & \\ & 1 & -3 & 1 & 3 & -1 \\ & & & & & -3 \end{pmatrix}.$$

We calculate the solution given by the new basis to be (in million pounds):  $x_1 = 1.2$ ,  $y_1 = 1.066\dots$ ,  $y_3 = 3.733\dots$ ,  $z_1 = 2.0$  and the slacks are set correspondingly. To find the range for which the new basis is optimal we reset the objective function coefficients of  $z_1$ ,  $z_2$ , and  $z_3$  such that optimality is exhibited for  $\delta = 0$ , i.e., we write now  $\alpha = 0.08496 - \delta$  and thus

$$c_7 = -0.00274 + \delta, \quad c_8 = -0.01896 + \delta, \quad c_9 = -0.01096 + \delta.$$

With the same notation as above we compute

$$\begin{aligned} c_B(\delta)B^{-1} &= (0.08222, 0.08222, 0.074, -0.00274 + \delta, 0, 0, 0, 0) B^{-1} \\ &= (0.08496, 0.074, \delta, 0, 0, 0, -0.00274, 0). \end{aligned}$$

The computation of the reduced cost for the structural nonbasic variables  $x_2$ ,  $y_2$ ,  $x_3$ ,  $z_2$ , and  $z_3$  yields (in the sequential indexing and LINDO's sign convention for optimality)

$$\bar{c}_3 = 0.01896, \quad \bar{c}_4 = 0.008, \quad \bar{c}_5 = 0.01096, \quad \bar{c}_8 = 0.01896, \quad \bar{c}_9 = 0.01096.$$

It follows that the new basis is optimal for all  $\delta \geq 0$  and thus there is no other "breakpoint". Summarizing, we have established that for  $\alpha \leq 0.08496$  the profit-optimal decision is to buy the entire lot of 2 million pounds of grade A apples with an optimal profit of

$$\$157,152 + \$2,000,000 \times (0.08496 - \alpha).$$

For  $0.08496 < \alpha < 0.09033$  the optimal decision is to buy 1.2 million pounds of grade A apples with an optimal profit of

$$\$150,708 + \$1,200,000 \times (0.09033 - \alpha).$$

For  $\alpha \geq 0.09033$  the optimal decision is to buy none of the available grade A apples at all giving an optimal profit of \$150,708 as in part(ii) of our analysis. Here, like before,  $\alpha$  is the cost per pound of grade A apples in dollars.

In the second approach to model the buying and allocation decision of additional grade A apples, we note that –up to the Apple Cost– the financial data of Table 1.2 remain the same. Hence it suffices to change the original model of part (ii) as follows: Let  $z \geq 0$  be the amount of additional grade A apples bought at the cost of  $\alpha$  dollars per pound. As before, we assume the cost of apple  $\alpha$  to be somewhere in the range Of 0.08 and 0.095. For  $\alpha = 0.09$  the linear program thus becomes:

```
Problem name: WACCOnew.lp for alpha=0.09
Maximize
  obj: 0.08222 x1 + 0.08222 y1 + 0.066 x2 + 0.066 y2 + 0.074 x3 + 0.074 y3
        -0.09z -300000x0
Subject To
  c0: x0 = 1
  c1: x1 + x2 + x3 - z <= 1200000
  c2: y1 + y2 + y3 <= 4800000
  c3: z <= 2000000
  c4: x1 + y1 <= 28800000
  c5: x2 + y2 <= 2000000
  c6: x3 + y3 <= 4000000
```

```
c7: x1 - 3 y1 >= 0
c8: 3 x2 - y2 >= 0
End
```

Note that now the objective function contains the “sunk” cost of the acquisition of the 6,000,000 pounds of apples explicitly and because all of the financial data of Table 1.2 except Apple Cost remain the same, the objective function of WACCOnew.lp calculates the profit for the product mix  $x_1, y_1, x_2, y_2, x_3, y_3$  and the acquisition of  $0 \leq z \leq 2,000,000$  pounds of grade A apples correctly. Solving the problem e.g. with CPLEX we find the following solution:

```
Primal - Optimal: Objective = 1.5110400000e+05
Solution time = 0.00 sec. Iterations = 6 (0)
```

Variable Name	Solution Value	Constraint Name	Dual Price
x1	2400000.000000	c0	-300000.000
y1	800000.000000	c1	0.09000
y3	4000000.000000	c2	0.05888
z	1200000.000000	c6	0.01512
		c7	-0.00778
		c8	-0.00800

All other variables and dual prices are zero.

The analysis of the problem becomes, of course, much easier as we now need to change only a single coefficient of the objective function parametrically. The results of the analysis are the same as before and the details of it are left to the reader. It is clear that the first approach is more general as it permits to change financial data from Table 1.2 other than the cost of apples as well.

---

### \*Exercise 1.3 (Minicase III) (Source: unknown)

ABCO manufactures two products X and Y. ABCO’s manager is interested in production and liquidity planning for the next two weeks of operation. The first week has four working days and the second week has five working days. Each working day has one eight-hour shift.

The firm has 200 machines that can be used for the production of both X and Y. A unit of product X takes 10 hours of machine time, whereas a unit of Y takes 16 hours. Sales of product Y cannot exceed 800 units during the next two weeks, whereas there is no such restriction on the sales of product X. Within those limits, ABCO does not produce on inventory and sells immediately what it produces. Sales price, wages to labor, and cost of raw material per unit of product X and product Y do not vary over time and are given in Table 1.3.

Wages are proportional to production volume and paid in the week when production occurs; raw materials are bought on credit of one week and delivered immediately, i.e., ABCO uses “just-in-time” delivery on a weekly basis and maintains no inventory of raw materials for its current products. All sales are on credit, and payments are received one week from the date of sale. The salary of the manager is \$200 per week; it is not affected by the production volume, but it is paid weekly like the wages are. The balance sheet at the beginning of the planning period (BoP) is given in Table 1.4.

**Table 1.3.** Cost and Revenue Data (in dollars per unit)

	Product X	Product Y
Sales price	\$55	\$125
Labor	\$30	\$48
Raw materials	\$10	\$50
Profit	\$15	\$27

From it we see that ABCO receives \$32,000 from accounts receivable and pays out \$9,500 at the start of the first week.

**Table 1.4.** ABCO's Balance Sheet as of BoP

Assets		Liabilities and Equity	
Cash	\$2000	Accounts Payable	\$9,500
Marketable Securities	\$600	Bank Loans	\$7,500
Accounts Receivable	\$32,000	Current Liabilities	\$17,000
Inventories	\$2,000	Long-term Debt	\$22,500
Current Assets	\$36,600	Net Equity	\$47,100
Fixed Assets	\$50,000		
Total Assets	\$86,600	Total Liabilities & Equity	\$86,600

As a matter of policy, ABCO maintains inventories of \$2,000 in parts of some obsolete models. The current balance of marketable securities is earmarked for tax payments which are due later in the year. ABCO maintains a minimum cash balance of \$500 and has an open line of credit of \$7,500 from a local bank. This line of credit is presently fully used, but repayment and interest payments on its loans are scheduled for later in the year. In order to preserve a favorable picture of the firm's financial situation, the manager has agreed with the local bank that ABCO's "quick ratio" should not drop below 2.0.

- (i) Verify that at the beginning of the period ABCO has a quick ratio greater-than-or-equal-to 2.0. Show your calculations. The "quick ratio" is defined as

$$\text{Quick Ratio} = \frac{\text{Current Assets} - \text{Inventories}}{\text{Current Liabilities}} .$$

- (ii) Let

- $x_1$  = the number of units of X produced and sold in week 1,
- $y_1$  = the number of units of Y produced and sold in week 1,

- $x_2$  = the number of units of X produced and sold in week 2,
- $y_2$  = the number of units of Y produced and sold in week 2.

Formulate the problem of finding a profit-optimal production-mix for ABCO that is within the various physical and policy restrictions as well as the limited scope of ABCO's financial capabilities as a two-period linear program in the above four variables.

- (iii) Compute a profit-optimal product-mix for ABCO using an interactive LP solver such as CPLEX, LINDO or OSL. What is the optimal profit ABCO can obtain?
  - (iv) What is the cash balance of ABCO at the end of week 1 and week 2? What are the respective two values of the quick ratio? Summarize your findings in a "cash budget" listing the cash position at the beginning of the period (BoP), total receipts, total cash available, total expenditures, the cash position at the end of the period (EoP), the minimum cash balance and excess cash in weeks 1 and 2 including the total cash available to finance operations in week 3.
  - (v) To ensure a "smooth" production schedule ABCO's management wants that the two weekly production volumina to schedule for product X and product Y, respectively, do not fluctuate too widely. More precisely, management wants the absolute difference of the production volumina of product X to not exceed 20% of the total production volume for product X over the two periods and likewise for product Y. Reformulate the linear program to include these considerations. What are the optimal production volumina to be scheduled and what is the optimal profit for the changed linear program? How much does it cost ABCO to smooth its production this way? How does this answer change if management insists on a maximum fluctuation of 10% rather than 20%?
  - (vi) Reformulate the problem of part (ii) using "auxiliary" free variables in order to facilitate the report writing for cash budget and quick ratio calculations.
  - (vii) Summarize your answers in an "executive summary" separate from your analysis.
- 

### **Executive Summary:**

- A. A linear programming analysis shows that ABCO can attain a maximum two-week profit of

\$23,600

while keeping full employment and "smoothing" the production over the two periods so that the (adjusted) fluctuations in production volumina for product X and Y, respectively, do not exceed 10%. To do so ABCO should produce

- 64 units of product X and 360 units of product Y in week 1,
- 96 units of product X and 440 units of product Y in week 2.

- B. The liquidity analysis for ABCO is very promising. The cash budget corresponding to the above product allocations shows that

$$\text{Quick Ratio (EoP1)} = \frac{\$54,220}{\$26,140} > 2.07, \quad \text{Quick Ratio (EoP2)} = \frac{\$71,660}{\$30,460} > 2.35,$$

at the end of week 1 and week 2, respectively. The excess cash for week 2 generated by ABCO is \$10,280 and –after paying for the raw materials used in the production of week 2– ABCO has \$48,100 to finance salary and wages in week 3. This suggests that in week 2 ABCO should consider investing a portion of its excess cash in revenue-producing short-term financial instruments such as marketable securities. Alternatively, ABCO may want to discuss with its local bank an early retirement of a part of its outstanding bank loans. The amount which should be applied in either case depends, of course, upon the market forecasts for week 3 and beyond.

### Solution to and Analysis of Minicase III:

(i) From the balance sheet of ABCO in Table 1.4 we find that

$$\begin{aligned}\text{Quick Ratio} &= \frac{\text{Cash} + \text{marketable securities} + \text{accounts receivable}}{\text{Accounts payable} + \text{bank loans}} \\ &= \frac{\$2,000 + \$600 + \$32,000}{\$9,500 + \$7,500} > 2.035.\end{aligned}$$

(ii) By assumption ABCO sells all of product X it can produce. The only sales restriction is the total sales of product Y which is limited to 800 units in the two-week period, i.e.,

$$y_1 + y_2 \leq 800$$

and of course,  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $y_1 \geq 0$ ,  $y_2 \geq 0$ . Given four work days of eight hours in week 1 we have 32 work hours and likewise, in week 2 we have 40 work hours of operation. ABCO has 200 machines and thus a total of 6,400 hours in week 1 and of 8,000 hours in week 2 of machine time for operations. Product X takes 10 hours of machine time per unit, while product Y takes 16 hours. We get the production capacity constraints

$$10x_1 + 16y_1 \leq 6,400 \quad \text{and} \quad 10x_2 + 16y_2 \leq 8,000,$$

which limit the production in the two week period.

Since ABCO wants to maintain a minimum cash balance of \$500 per week we need to formulate two cash balance constraints of the form

$\text{ending balance} = \text{beginning balance} + \text{inflows} - \text{outflows} \geq \text{minimum balance}$

for the two weeks. In the first week ABCO has \$2,000 in cash on hand and receives \$32,000 from its accounts receivable. ABCO has to pay \$9,500 on accounts payable, \$200 for the salary of the manager plus the wages for labor of  $30x_1 + 48y_1$  during the first week. Since raw materials are bought on one week's credit the cash balance at the end of week 1 must satisfy

$$\$2,000 + \$32,000 - \$9,500 - \$200 - (30x_1 + 48y_1) \geq \$500$$

or simplifying and rearranging we get the cash balance constraint for week 1

$$30x_1 + 48y_1 \leq 23,800.$$

The second week begins with a cash position of  $23,800 + 500 - (30x_1 + 48y_1)$ . ABCO receives  $55x_1 + 125y_1$  dollars from accounts receivable. Cash outflows are wages and salary for week 2 plus the payment on accounts payable for the raw materials bought in week 1, i.e.,

$$(24,300 - 30x_1 - 48y_1) + (55x_1 + 125y_1) - (30x_2 + 48y_2 + 200 + 10x_1 + 50y_1) \geq 500,$$

or simplifying and arranging we get the cash balance constraint for week 2

$$-15x_1 - 27y_1 + 30x_2 + 48y_2 \leq 23,600.$$

Since ABCO needs to maintain a quick ratio of at least 2.0 at the end of both weeks we get two more constraints of the general form

$$\text{Quick Ratio} = \frac{\text{Cash} + \text{marketable securities} + \text{accounts receivable}}{\text{Accounts payable} + \text{bank loans}} \geq 2.0$$

From the above we know that ABCO finishes the first week with  $24,300 - 30x_1 - 48y_1$  dollars in cash, marketable securities valued at \$600 and  $55x_1 + 125y_1$  dollars on accounts receivable. Accounts payable at the end of the first week are  $10x_1 + 50y_1$  dollars and bank loans amount to \$7,500. Consequently,

$$\frac{24,300 - 30x_1 - 48y_1 + 600 + 55x_1 + 125y_1}{10x_1 + 50y_1 + 7,500} \geq 2.0$$

and simplifying we get the quick ratio constraint for week 1

$$-5x_1 + 23y_1 \leq 9,900.$$

Likewise, we end week 2 with  $24,100 + 15x_1 + 27y_1 - 30x_2 - 48y_2$  dollars in cash, marketable securities valued at \$600 and  $55x_2 + 125y_2$  dollars on accounts receivable. Accounts payable at the end of week 2 are  $10x_2 + 50y_2$  dollars and bank loans remain at \$7,500. Writing down the corresponding quick ratio constraint for week 2 and simplifying we get

$$-15x_1 - 27y_1 - 5x_2 + 23y_2 \leq 9,700.$$

As we are concerned with the very short-term horizon of two weeks we can take the total attainable profit over the two weeks as the objective function, i.e., we can ignore the "time value" of money and avoid "discounting" the second week's profit to make it truly comparable to the first week's profit. The resulting linear program is shown in part (iii) in the form of input for the LP solver LINDO.

(iii) From the formulation of part (ii) the input to LINDO and its solution are as follows.

```
! ABCO's problem
max 15x1 + 27y1 + 15x2 + 27y2
subject to
      y1      +    y2 <   800 ! Sales limitation
      10x1 + 16y1           <  6400 ! Production week 1
                  10x2 + 16y2 <  8000 ! Production week 2
      30x1 + 48y1           < 23800 ! Cash balance week 1
     -15x1 - 27y1 + 30x2 + 48y2 < 23600 ! Cash balance week 2
      -5x1 + 23y1           <  9900 ! Quick ratio week 1
```

```

-15x1 - 27y1 - 5x2 + 23y2 < 9700 ! Quick ratio week 2
! All vars nonnegative

LP OPTIMUM FOUND AT STEP      4
OBJECTIVE FUNCTION VALUE
1)    24000.00
VARIABLE      VALUE      REDUCED COST
X1      160.000000      0.000000
Y1      300.000000      0.000000
Y2      500.000000      0.000000
ROW    SLACK OR SURPLUS      DUAL PRICES
2)      0.000000      3.000000
3)      0.000000      1.500000
4)      0.000000      1.500000

```

After paying the manager \$400, ABCO can thus attain a two-week profit of **\$23,600** with the product-mix:

- 160 units of product X and 300 units of product Y in week 1.
- 0 units of product X and 500 units of product Y in week 2.

The resulting production schedule looks rather unbalanced, but guarantees full employment for ABCO's work force for the upcoming two weeks (why?).

Inspection of the (long) computer output (including the zero primal and dual variables) shows that the reduced profit or cost of variable  $x_2$  equals zero. Thus there may be alternative optima for ABCO's problem. To find one we use a **perturbation device** for the objective function.

We replace the term  $15x_2$  in the objective function by  $(15 + \varepsilon)x_2$ , where  $\varepsilon > 0$  because we are maximizing. Setting  $\varepsilon = 0.00001$  and rerunning the problem we find after one iteration the alternative optimal production schedule

$$x_1 = 0, y_1 = 400, x_2 = 160, y_2 = 400,$$

which also gives an objective function value of \$24,000. Like the first one it is rather unbalanced as far as product X is concerned, but guarantees full employment as well. The optimal solution reported by LINDO and most LP solvers depends on the **order of the variables and constraints** in which they are read (and stored internally) by the computer during the input phase. So your solution may be different, but not the objective function value.

**(iv)** ABCO's cash budget for the (first) product-mix of part (iii) for the two weeks is shown in Table 1.5. The items that have to be calculated from the actual production schedule are explained in the footnotes.

Likewise we compute the quick ratios at the end of week 1 (EoP1) and week 2 (EoP2):

$$\text{Quick Ratio (EoP1)} = \frac{\$5,100 + \$600 + \$46,300}{\$16,600 + \$7,500} = \frac{\$52,000}{\$24,100} > 2.157,$$

$$\text{Quick Ratio (EoP2)} = \frac{\$10,600 + \$600 + \$62,500}{\$25,000 + \$7,500} = \frac{\$73,700}{\$32,500} > 2.267.$$

**Table 1.5.** ABCO's Cash budget

	<i>Week 1</i>	<i>Week 2</i>	<i>Week 3</i>
<i>Cash balance BoP</i>	\$2,000	\$5,100	\$10,600
<i>Total receipts</i>	\$32,000	\$46,300 <sup>2)</sup>	\$62,500 <sup>4)</sup>
<i>Total cash available</i>	\$34,000	\$51,400	\$73,100
<i>Total disbursements</i>	\$28,900 <sup>1)</sup>	\$40,800 <sup>3)</sup>	
<i>Cash balance EoP</i>	\$5,100	\$10,600	
<i>Minimum cash balance</i>	\$500	\$500	\$500
<i>Excess cash</i>	\$4,600	\$10,100	

<sup>1)</sup> Accounts payable + Salary + Wages = \$9,500 + \$200 + (\$30 × 160 + \$48 × 300)

<sup>2)</sup> Accounts receivable = \$55 × 160 + \$125 × 300

<sup>3)</sup> Accounts payable + Salary + Wages = \$16,600 + \$200 + (\$30 × 0 + \$48 × 500)

<sup>4)</sup> Accounts receivable = \$55 × 0 + \$125 × 500

These numbers show that ABCO –after the accounts payable accrued in week 2 are paid for– has \$48,000 in cash to cover salary and wages in week 3, which seems excessive.

(v) To smooth ABCO's production over the two-week period for product X and product Y we want

$$|x_1 - x_2| \leq 0.\alpha(x_1 + x_2) \text{ and } |y_1 - y_2| \leq 0.\alpha(y_1 + y_2),$$

where  $\alpha$  is the percentage fluctuation that management is willing to permit. Equivalently we can write the linear constraints

$$-0.\alpha(x_1 + x_2) \leq x_1 - x_2 \leq 0.\alpha(x_1 + x_2) \text{ and } -0.\alpha(y_1 + y_2) \leq y_1 - y_2 \leq 0.\alpha(y_1 + y_2).$$

Simplifying and rearranging we get for  $\alpha = 20\%$  the following four constraints for ABCO's production and liquidity planning model

$$-1.2x_1 + 0.8x_2 \leq 0, \quad 0.8x_1 - 1.2x_2 \leq 0, \quad -1.2y_1 + 0.8y_2 \leq 0, \quad 0.8y_1 - 1.2y_2 \leq 0.$$

The input and solution report from LINDO follow.

```
! ABCO's problem with production smoothing
max 15x1 + 27y1 + 15x2 + 27y2
subject to
      y1      +    y2 < 800 ! Sales limitation
      10x1 + 16y1           < 6400 ! Production week 1
      10x2 + 16y2 < 8000 ! Production week 2
      30x1 + 48y1           < 23800 ! Cash balance week 1
     -15x1 - 27y1 + 30x2 + 48y2 < 23600 ! Cash balance week 2
      -5x1 + 23y1           < 9900 ! Quick ratio week 1
     -15x1 - 27y1 - 5x2 + 23y2 < 9700 ! Quick ratio week 2
     -1.2x1      +0.8x2       < 0 ! Smoothing product X
      0.8x1      -1.2x2       < 0 ! Smoothing product X
```

```

-1.2y1      +0.8y2 <   0 ! Smoothing product Y
  0.8y1      -1.2y2 <   0 ! Smoothing product y
                           ! All vars nonnegative

LP OPTIMUM FOUND AT STEP      5
OBJECTIVE FUNCTION VALUE
 1)    24000.00
VARIABLE      VALUE      REDUCED COST
  X1      64.000000      0.000000
  Y1      360.000000     0.000000
  X2      96.000000      0.000000
  Y2      440.000000     0.000000

ROW  SLACK OR SURPLUS      DUAL PRICES
 2)      0.000000      3.000000
 3)      0.000000      1.500000
 4)      0.000000      1.500000

```

From the solution report of the smoothed production and liquidity planning problem we find yet another alternative optimal solution, different from the one obtained in part (iii). Given its current data configuration ABCO's profit is not reduced by the additional production smoothing constraints and full employment of labor is assured as well.

After paying the manager's \$400, ABCO can thus attain a two-week profit of  $\boxed{\$23,600}$  with the product-mix:

- 64 units of product X and 360 units of product Y in week 1,
- 96 units of product X and 440 units of product Y in week 2.

If the fluctuation in the production schedule for the two-week period is reduced to  $\alpha = 10\%$  then this policy decision reduces ABCO's operating profit by  $\$277.67$ . More precisely, with this stronger fluctuation restriction ABCO attains a two-week profit of  $\boxed{\$23,333.33}$  with a product-mix of 64 units of product X and 360 units of product Y in week 1 and 78.22 units of product X and 440 units of product Y in week 2. This is achieved by not working the full five days in week 2 which may cause problems with labor representatives.

Given the unequal lengths of the two work weeks it does not seem reasonable to insist on smoothing the (uncomparable) production volumes of week 1 and week 2. Rather to make the two different work weeks comparable we should "scale up" the production in week 1 by a factor of 1.25 (to account for the four days of week 1 and a "normal" five day work week). We then require that

$$|1.25x_1 - x_2| \leq 0.\alpha(1.25x_1 + x_2) \text{ and } |1.25y_1 - y_2| \leq 0.\alpha(1.25y_1 + y_2).$$

Thus the two linear smoothing constraints for product X become

$$-1.25(1 + 0.\alpha)x_1 + (1 - 0.\alpha)x_2 \leq 0 \text{ and } -1.25(1 - 0.\alpha)x_1 + (1 + 0.\alpha)x_2 \leq 0,$$

and likewise for product Y. With this adjustment the production-mix found initially for  $\alpha = 20\%$  produces an adjusted fluctuation for product X and product Y of

$$\frac{|1.25x_1 - x_2|}{1.25x_1 + x_2} = \frac{16}{176} \approx 9.1\%, \quad \frac{|1.25y_1 - y_2|}{1.25y_1 + y_2} = \frac{10}{990} \approx 1\%,$$

**Table 1.6.** ABCO's Cash budget for smoothed production

	<i>Week 1</i>	<i>Week 2</i>	<i>Week 3</i>
<i>Cash balance BoP</i>	\$2,000	\$5,100	\$10,780
<i>Total receipts</i>	\$32,000	\$48,520 <sup>2)</sup>	\$60,280 <sup>4)</sup>
<i>Total cash available</i>	\$34,000	\$53,620	\$71,060
<i>Total disbursements</i>	\$28,900 <sup>1)</sup>	\$42,840 <sup>3)</sup>	
<i>Cash balance EoP</i>	\$5,100	\$10,780	
<i>Minimum cash balance</i>	\$500	\$500	\$500
<i>Excess cash</i>	\$4,600	\$10,280	

<sup>1)</sup> Accounts payable + Salary + Wages = \$9,500 + \$200 + (\$30 × 64 + \$48 × 360)

<sup>2)</sup> Accounts receivable = \$55 × 64 + \$125 × 360

<sup>3)</sup> Accounts payable + Salary + Wages = \$18,640 + \$200 + (\$30 × 96 + \$48 × 440)

<sup>4)</sup> Accounts receivable = \$55 × 96 + \$125 × 440

which – with full employment in the two weeks– is below the 10% range desired.

Policy constraints of this nature are typically **soft constraints**, i.e., constraints for which “small” violations can usually be tolerated by management and labor. To measure the amount of “permissible” violation of a soft constraint an auxiliary variable  $x_{aux}$  is introduced. Variable  $x_{aux}$  gets a **penalty term** in the objective function like in any Big-M method. E.g. if we have a maximization problem and a soft less-than-or-equal-to constraint, then we replace the corresponding right-hand side  $b$ , say, by  $b + x_{aux}$  and add the term  $-Mx_{aux}$  to the objective function. The numerical value for  $M$  is typically obtained by some heuristic argument or by managerial decision. In some cases a contractual agreement (specifying penalties for production delays or such) may exist or a theoretical analysis may be possible to determine numerical values for the penalties on the amount of violation more precisely.

The cash budget for the smoothed production allocation is shown in Table 1.6.

Like above we compute the quick ratios at the end of week 1 (EoP1) and week 2 (EoP2):

$$\text{Quick Ratio (EoP1)} = \frac{\$5,100 + \$600 + \$48,520}{\$18,640 + \$7,500} = \frac{\$54,220}{\$26,140} > 2.07,$$

$$\text{Quick Ratio (EoP2)} = \frac{\$10,780 + \$600 + \$60,280}{\$22,960 + \$7,500} = \frac{\$71,660}{\$30,460} > 2.35.$$

Like in the first analysis, these numbers show that ABCO –after the accounts payable accrued in week 2 are paid for– has \$48,100 in cash to cover salary and wages in week 3, which seems excessive.

(vi) To facilitate the report writing after the problem solving phase, but also in order to increase the “transparency” of the problem formulation during the formulation phase, linear programmers frequently use **auxiliary free variables** in their formulations. In the case of ABCO’s problem e.g. the following auxiliary free variables suggest themselves:

- Let  $A_{pi}$  for  $i = 0, 1, 2$  be the position of the Accounts payable at the end of periods 0, 1 and 2, respectively, where the end of period 0, say, is the beginning of period 1. Since ABCO pays

salary and wages during the periods and accumulates only the cost of the raw materials on its Accounts payable, we get the relations

$$Ap_0 = 9,500, \quad Ap_1 = 10x_1 + 50y_1, \quad Ap_2 = 10x_2 + 50y_2.$$

- Let  $Ar_i$  for  $i = 0, 1, 2$  be the position of the Accounts receivable at the end of periods 0, 1 and 2, respectively. Because ABCO sells its products on one week's credit we get the relations

$$Ar_0 = 32,000, \quad Ar_1 = 55x_1 + 125y_1, \quad Ar_2 = 55x_2 + 125y_2.$$

- Let  $SW_i$  for  $i = 0, 1, 2$  be the Salary and Wages paid by ABCO during week 1 and week 2, respectively. We get the relations

$$SW_1 = 200 + 30x_1 + 48y_1, \quad SW_2 = 200 + 30x_2 + 48y_2.$$

- Let  $Ca_i$  for  $i = 0, 1, 2$  be the Cash position of ABCO at the end of periods 0, 1 and 2, respectively. Using the previously introduced auxiliary free variables we get the relations

$$Ca_0 = 2,000, \quad Ca_1 = Ca_0 + Ar_0 - Ap_0 - SW_1, \quad Ca_2 = Ca_1 + Ar_1 - Ap_1 - SW_2.$$

- The quick ratio constraints in the auxiliary free variables now become

$$\frac{Ca_1 + Ar_1 + 600}{Ap_1 + 7,500} \geq 2.0, \quad \frac{Ca_2 + Ar_2 + 600}{Ap_2 + 7,500} \geq 2.0.$$

- The minimum cash balance constraints for ABCO read:  $Ca_1 \geq 500, \quad Ca_2 \geq 500$ .

Rearranging and simplifying the above relations to fit the usual linear programming format we get the following input for LINDO, which has now 15 variables and 18 constraints. We have 4 variables that must be nonnegative as before and 11 auxiliary free variables.

```
! ABCO's problem with auxiliary free variables
Max 15x1 + 27y1 + 15x2 + 27y2
Subject to
      y1      +   y2      <  800 ! Sales
    10x1 + 16y1      < 6400 ! Production 1
      10x2 + 16y2      < 8000 ! Production 2
                           Ap0      = 9500 ! Payables EoP0
  -10x1 - 50y1      + Ap1      =     0 ! Payables EoP1
  -10x2 - 50y2      + Ap2      =     0 ! Payables EoP2
                           Ar0      = 32000 ! Receivables EoP0
  -55x1 -125y1      + Ar1      =     0 ! Receivables EoP1
  -55x2 -125y2      + Ar2      =     0 ! Receivables EoP2
  -30x1 - 48y1      + SW1      =  200 ! Wages 1
  -30x2 - 48y2      + SW2      =  200 ! Wages 2
                           Ca0      = 2000 ! Cash EoP0
  SW1 + Ap0 - Ar0 - Ca0 + Ca1 =     0 ! Cash EoP1
  SW2 + Ap1 - Ar1 - Ca1 + Ca2 =     0 ! Cash EoP2
```

```

-2Ap1 + Ar1 + Ca1 >14400 ! Q Ratio EoP1
-2Ap2 + Ar2 + Ca2 >14400 ! Q Ratio EoP2
Ca1 > 500 ! Min cash EoP1
Ca2 > 500 ! Min cash EoP2
END
FREE Ap0 Ap1 Ap2 Ar0 Ar1 Ar2 SW1 Sw2 Ca0 Ca1 Ca2

```

The actual LINDO input requires a separate line for each free variable; in the above formulation it recognizes only  $Ap_0$  as a free variable and ignores the other variables, which by default become nonnegative variables. In our particular case this does not hurt, but it may in general. The solution to ABCO's problem is shown below. It permits to find most entries necessary to set up the cash budget and to do the quick ratio calculations. Every (skilled) computer programmer will recognize that with the "trick" of introducing auxiliary free variables it becomes possible to fully integrate the linear programming calculations with an **automatic report generation**. Most contemporary commercial LP solvers take care automatically of such free variables reducing sometimes the overall computing times dramatically by so-called **preprocessing techniques**. The amazing student version of LINDO that we used managed to take longer as the output shows.

LP OPTIMUM FOUND AT STEP		19	OBJECTIVE FUNCTION VALUE					
1)	24000.00		VARIABLE	VALUE	REDUCED COST	VARIABLE	VALUE	REDUCED COST
X1	160.000000	0.000000	SW1	19400.0	0.00			
Y1	300.000000	0.000000	SW2	24200.0	0.00			
X2	0.000000	0.000000	CA0	2000.0	0.00			
Y2	500.000000	0.000000	CA1	5100.0	0.00			
AP0	9500.000000	0.000000	CA2	10600.0	0.00			
AP1	16600.000000	0.000000				ROW SLACK/SURPLUS DUAL PRICES		
AP2	25000.000000	0.000000				2)	0.000	3.00
AR0	32000.000000	0.000000				3)	0.000	1.50
AR1	46300.000000	0.000000				4)	0.000	1.50
AR2	62500.000000	0.000000						

#### \*Exercise 1.4 (Minicase IV)

**History:** Like the rest of Germany after her unconditional surrender in May of 1945, the city of Berlin was divided into an American, British, French and a Russian sector. Berlin was geographically located in the "Soviet" (Russian) zone and access to the western (American, British and French) sectors of the city was assured by agreement with the Soviets via air, rail and three highways from Hamburg, Hannover and Hof in Northern Bavaria, respectively, until fairly recently (1990) when history took a different turn again. When the three western sectors of the city of Berlin adopted in early 1948 the "new" currency that had just been created in the "tri-zone", i.e., the American, British and French zones of Germany, the Soviets –in the person of Joseph Vissarionovich Djugashvili, a.k.a. Joseph Stalin (1879–1953)– reacted with a total blockade of road and rail access to the city

**Table 1.7.** Relevant cost data in MU's per unit

	Period 1	Period 2	Period 3	Period 4
New plane	200.0	195.0	190.0	185.0
Idle pilot	7.0	6.9	6.8	6.7
New pilot	10.0	9.9	9.8	9.7
Resting pilot	5.0	4.9	4.8	4.7

of Berlin for all westerns, including American, British and French forces. A massive and costly “airlift” to Berlin (called “Operation Fiddle” by the military) was organized by the American, British and French forces that reportedly transported over 2.3 million tons of food, clothing, fuel, asphalt, pet food, etc into West Berlin to save West Berliners (and their beloved ones) from starving and freezing. Operation Fiddle lasted 463 days or about 15 months –from June 1948 to September 1949– when the Soviets finally backed off. Clearly, like in modern warfare, such a massive effort requires a great deal of logistics planning.

**Problem:** You have been assigned to come up with a feasibility study to apply linear and integer programming techniques to aid the logistics planning staff in their analysis. To do so you have been given a very simplified version of the problem faced by the logistics managers. More specifically, you have been asked to plan the logistics of the operation at a very aggregate level for four consecutive quarters, e.g. four three-month periods, assuming the following scenario.

- For each quarter you have a forecast of the cargo that must be airlifted. One unit of cargo corresponds to 100,000 tons, say, and in quarters 1, 2, 3 and 4 you need to airlift exactly 2, 3, 3 and 4 units of cargo, respectively. Each unit of airlifted cargo requires 50 airplanes and three pilots are necessary to man and operate one plane.
- At the beginning of the first quarter you have 330 pilots –personnel that can operate planes or train new pilots– and 110 airplanes. You have to plan the recruitment of new personnel and the procurement of new aircraft ahead of time.
- In each quarter 20% of the flying personnel and aircraft are lost (for the rest of the planning period) due to planes that go down in the Soviet zone before reaching the tri-zone. To simplify matters, “lost” crew is equated to the corresponding lost aircraft and aircraft is never lost on the way to Berlin, but only on the way back from Berlin, when the pilots are tired. Aircraft procured in any period is ready for use in the following period.
- Pilots that do not operate an aircraft are either idle or train new pilots. You have been told to assume a ratio of 1 to 20, i.e., in every quarter each pilot trainer “produces” 20 new pilots (including himself) that are ready to operate aircraft in the following period.
- Crews that have operated an aircraft during one quarter are given leave the following quarter and are available again after their rest period for a new round of duty. Despite the enormous “attrition” rate of 20% for lost crew and aircraft morale among the personnel is high and all personnel that were given leave return to service after their rest period.
- The relevant cost data for your analysis are given in monetary units (MU’s) in Table 1.7. All other cost are immaterial and assumed to be zero.

- (i) Let  $Ca_j$  be the units of cargo airlifted in period  $j$  for  $j = 1, \dots, 4$ ,  
 $UP_j$  be the number of unused or “idle” planes in period  $j$  for  $j = 1, \dots, 4$ ,  
 $NP_j$  be the number of new planes procured in period  $j$  for  $j = 1, \dots, 4$ ,  
 $Pi_j$  be the number of idle pilots in period  $j$  for  $j = 1, \dots, 4$ ,  
 $Pn_j$  be the number of trainees (including their trainers) in period  $j$  for  $j = 1, \dots, 4$ ,  
 $Pr_j$  be the number of resting pilots in period  $j$  for  $j = 1, \dots, 4$ .  
Formulate the problem of finding a cost-minimal logistics plan as a linear programming (LP) model in terms of the above 24 variables.
- (ii) Solve the problem as a linear program using an interactive LP solver such as CPLEX, LINDO or OSL. Is the LP solution implementable?
- (iii) Solve the problem as a linear program in integer variables using an ordinary branch-and-bound solver, such as e.g. given by the default options of LINDO. Summarize your observations.
- (iv) Suppose you know that there are integer solutions with an objective function value of less than 47,000. Rerun your integer program by supplying the solver an upper bound of 47,000 on the objective function, e.g. by specifying in LINDO’s options menu an “IP Objective Hurdle” of 47,000. Summarize your observations.
- (v) Joe Doe, an integer programming whiz kid, has analyzed your problem and tells you that given your cargo forecasts every integer solution to your logistics planning model must satisfy the additional constraints

$$Pi_1 \leq 7, \quad 20Pi_1 + Pi_2 \leq 146, \quad 400Pi_1 + 20Pi_2 + Pi_3 \leq 2,924.$$

Add these constraints (called “cuts” or “cutting planes”) to your formulation and rerun the problem. Summarize your observations.

- (vi) Joe Doe also suggested that you replace the variables  $Pn_j$  for  $j = 1, \dots, 4$  by

$$Pt_j = \text{number of pilot trainers in period } j \text{ for } j = 1, \dots, 4.$$

Why did he do that? How does your formulation change? Run the reformulated problem and summarize your observations.

- (vii) Suppose you eliminate the variables modeling the cargo shipments from the LP model through substitution. How does the model change and what are its implications? State your observations succinctly. Do you think that your findings are typical for this kind of logistics management?
- (viii) Summarize your findings in an “executive” summary. Be sure to include an operational schedule for crew management and aircraft utilization and procurement for the simplified scenario of Operation Fiddle.

### **Executive Summary:**

The problem posed to us has been formulated successfully as a linear programming problem in integer variables. The integrality of the decision variables is essential for the usefulness of computer generated logistics schedules. The analysis of the simplified model for Operation

**Table 1.8.** Aircraft utilization and procurement

Period	1	2	3	4
Cargo	2	3	3	4
Total Planes	110	150	150	200
Flying Planes	100	150	150	200
Idle Planes	10	0	0	0
Lost Planes	20	30	30	40
Procurement	60	30	80	0

**Table 1.9.** Crew management

Period	1	2	3	4
Total crew	330	707	826	964
Flying crew	300	450	450	600
Lost crew	60	90	90	120
Resting crew	0	240	360	360
Idle crew	7	6	4	4
Trainers	23	11	12	0
Trainees	437	209	228	0

Fiddle shows that this approach to the problems faced by the operations logistics managers is an invaluable tool that has great promise to automatize and improve many of the tedious aspects of the scheduling process. In Tables 1.8 and 1.9 we summarize the computer generated, **cost-minimal** results of our analysis in a convenient form for the logistics managers by stating the new aircraft to be procured as well as the trainees to be recruited for each quarter.

### Solution to and Analysis of Minicase IV:

(i) Using the variable definitions introduced above in part (i) we formulate the problem as follows.

- 1.) The cargo shipments give rise to four equations:  $Ca_1 = 2$ ,  $Ca_2 = 3$ ,  $Ca_3 = 3$ ,  $Ca_4 = 4$ .
- 2.) To account for the aircraft we need to make sure that in each period

$$\text{required flying aircraft} \leq \text{available aircraft}.$$

The slack in the inequalities is the unused (idle) aircraft that stays on the ground for possible use in the next period. Since 1 unit of cargo requires 50 planes,  $50Ca_j$  airplanes take off to carry  $Ca_j$  units of cargo in period  $j$  and thus for period 1

$$50Ca_1 + UP_1 = 110.$$

Of the  $50Ca_j$  planes that take off in the  $j$ -th period we loose 20% or  $10Ca_j$  planes. Thus e.g.  $40Ca_1$  plus the  $UP_1$  unused ones of the original 110 aircraft remain available for use in period 2. We procure  $NP_j$  new aircraft in period  $j$  that are ready for use in period  $j+1$ . Hence

$$50Ca_j + UP_j = 40Ca_{j-1} + UP_{j-1} + NP_{j-1} \quad \text{for } j = 2, 3, 4.$$

- 3.) To man  $50Ca_j$  aircraft we need  $3 \times 50Ca_j = 150Ca_j$  pilots in every period. To have  $Pn_j$  newly trained pilots in period  $j+1$  requires  $\frac{1}{20}Pn_j = 0.05Pn_j$  trainers in period  $j$  because to get 20 new pilots (including the trainer) we need 1 trainer. Thus for period 1

$$150Ca_1 + 0.05Pn_1 + Pi_1 = 110.$$

Likewise in periods 2, 3, 4 we need  $150Ca_j + 0.05Pn_j$  pilots. The number of pilots available in period  $j$  are the  $Pi_{j-1}$  pilots that were idle in the previous period, the  $Pn_{j-1}$  newly

**Table 1.10.** Example of an airlift model

Cargo				Planes				Crews												
$Ca_1$	$Ca_2$	$Ca_3$	$Ca_4$	Idle	New	Idle	New	Resting												
$UP_1$	$UP_2$	$UP_3$	$UP_4$	$NP_1$	$NP_2$	$NP_3$	$NP_4$	$Pi_1$	$Pi_2$	$Pi_3$	$Pi_4$	$Pn_1$	$Pn_2$	$Pn_3$	$Pn_4$	$Pr_1$	$Pr_2$	$Pr_3$	$Pr_4$	
0	0	0	0	0	0	200	195	190	185	7	6.96	86.7	10	9.9	9.8	9.7	5	4.94	84.7	min
1																		= 2		
	1																	= 3		
		1																= 3		
			1															= 4		
50	-40	50	-40	50	-40	50	1	-1	1	-1	-1	-1						= 110		
																		= 0		
																		= 0		
																		= 0		
150	150	150	150										1	.05				= 330		
													-1	.05				= 0		
													-1	1				= 0		
													-1	1				= 0		
-120	-120	-120	-120													1		= 0		
																1		= 0		
																	1		= 0	
																		1	= 0	

trained pilots (including their trainers) and the  $Pr_{j-1}$  “rested” pilots that were given leave for a previous duty period and returned to base, i.e., with  $Pi_j$  pilots idle in period  $j$  we have

$$150Ca_j + 0.05Pn_j + Pi_j = Pi_{j-1} + Pn_{j-1} + Pr_{j-1} \quad \text{for } j = 2, 3, 4.$$

- 4.) Only flying personnel that returned safely to the base is given leave in the period following their tour of duty. Since we have a 20% attrition rate  $30Ca_j$  of the  $150Ca_j$  pilots that took off are dead or captives in Soviet hands and  $120Ca_j$  return to base. Hence

$$Pr_1 = 0, \quad Pr_j = 120Ca_{j-1} \quad \text{for } j = 2, 3, 4.$$

- 5.) Since the cost in Table 1.7 are on a per unit basis and our variables are in those units we get the objective function by multiplying each cost term by its respective variable and adding up.
- 6.) Requiring that all variables are nonnegative integers we have the formulation displayed in Table 1.10 for Operation Fiddle. We should, but do not model explicitly, the requirement that the  $Pn_j$  must be multiples of 20, i.e., that  $Pn_j = 20x_j$  for some  $x_j \geq 0$  and  $x_j = \text{integer}$  for  $j = 1, \dots, 4$ . In this first shot at a formulation we simply assume that this constraint will be satisfied automatically (which happens to be true for our data). As we will see, we will pay a price for this omission; see below.

(ii) Input of the linear program displayed in Table 1.10 in LINDO's format is shown below. In Table 1.1 both the LP solution and the integer programming (IP) solution to the problem Operation Fiddle are displayed (to an accuracy of three digits after the point). From both solutions we read that 60 new aircraft must be procured in period 1, 30 in period 2 and 80 in period 3. As for crew management we read from the LP solution that  $Pi_1 = 7.311$  pilots are idle and  $Pn_1 = 453.789$  new pilots should be trained in period 1 for a tour of duty in period 2. This requires that  $453.789/20 \approx 22.69$  pilots train 431.099 new recruits in period 1. Pilots like aircraft are, of course, "indivisible" and thus these numbers are at best guidelines for the decisions faced by the logistics managers. We could use "rounding" and then trace the result through the system of equations. We would then still have to be "lucky" to find a feasible solution in nonnegative integers to the scheduling problem. Thus the use of integer programming techniques becomes necessary. The second half of Table 1.1 shows the solution obtained by invoking the integer programming facilities of LINDO. As we see the optimal number of pilot trainers is  $460/20 = 23$  in period 1, 11 in period 2 and 12 in period 3, i.e., in this particular case the divisibility by 20 is satisfied automatically, but this need, of course, not be the case for other data constellations.

```

! Operation FIDDLE
min 0Ca1 + 0Ca2 + 0Ca3 + 0Ca4 + 0Up1 + 0Up2 + 0Up3 + 0Up4 +
200Np1 + 195Np2 + 190Np3 + 185Np4 + 7Pi1 + 6.9Pi2 + 6.8Pi3 + 6.7Pi4 +
10Pn1 + 9.9Pn2 + 9.8Pn3 + 9.7Pn4 + 5Pr1 + 4.9Pr2 + 4.8Pr3 + 4.7Pr4
subject to
    Ca1                      = 2 ! Cargo period 1
    Ca2                      = 3 ! Cargo period 2
    Ca3                      = 3 ! Cargo period 3
    Ca4                      = 4 ! Cargo period 4
    50Ca1 + Up1              = 110 ! Planes period 1
    -40Ca1 + 50Ca2 - Up1 + Up2 - Np1 = 0 ! Planes period 2
    -40Ca2 + 50Ca3 - Up2 + Up3 - Np2 = 0 ! Planes period 3
    -40Ca3 + 50Ca4 - Up3 + Up4 - Np3 = 0 ! Planes period 4
    150Ca1 + Pi1 + .05Pn1     = 330 ! Pilots period 1
    150Ca2 - Pi1 + Pi2 - Pn1 + .05Pn2 = 0 ! Pilots period 2
    150Ca3 - Pi2 + Pi3 - Pn2 + .05Pn3 - Pr2 = 0 ! Pilots period 3
    150Ca4 - Pi3 + Pi4 - Pn3 + .05Pn4 - Pr3 = 0 ! Pilots period 4
                                Pr1 = 0 ! Resting pilots period 1
    -120Ca1                  + Pr2 = 0 ! Resting pilots period 2
    -120Ca2                  + Pr3 = 0 ! Resting pilots period 3
    -120Ca3                  + Pr4 = 0 ! Resting pilots period 4
                                ! All vars nonnegative

```

(iii) Running LINDO with the additional requirement that all variables of the problem displayed above are general integers (of the type GIN in LINDO's language) results in a disaster, even though the problem is by contemporary standards a "lilliputian" integer program: LINDO's solution report states that 5,216 "branches" of the branch-and-bound search tree have been explored and a suboptimal integer solution with objective function value 51,759.8 was found during the search, which is rather bad in terms of the optimal objective function value of 46,920.4.

**(iv)** Running LINDO like under part (iii) but with a value of 47,000 for the LINDO parameter IP Objective Hurdle –which is found in the OPTIONS menu of LINDO– produces a run involving only 52 branches of the branch-and-bound search tree. This run finds the optimal integer solution displayed in Table 1.1. It shows that ordinary, vanilla-varietyp IP solvers like LINDO can sometimes be sped up considerably by supplying the IP solver with a “good” upper bound on the optimal integer programming objective function value. Of course, to find such bounds for general IP problems is just as difficult as finding an optimal integer solution to the problem.

**(v)** Running LINDO like under part (iii) but with the additional three constraints found by Joe Doe reduces this integer program into an ordinary linear program: Solving the associated linear programming relaxation (without any integrality requirement) we find the optimal integer solution displayed in Table 1.1. Thus “branching” becomes unnecessary. Such “cutting planes” are at the heart of the approach called “branch-and-cut” which is at the cutting-edge of research on the solution of integer and mixed-integer programming problems. As happens often with recent research results, these new methods are finding their entry slowly (too slowly!) into commercial codes for integer programming such as CPLEX, LINDO and OSL.

**(vi)** Changing the variables  $Pn_j$  to  $Pt_j$  as defined in part (vi) of the problem formulation changes the accounting constraints for crew management and the objective function of the problem formulated under part (i), point 3. For period 1 we get

$$150Ca_1 + Pt_1 + Pi_1 = 330.$$

Since every pilot trainer produces 20 new pilots (including himself) we obtain

$$150Ca_j + Pt_j + Pi_j = Pi_{j-1} + 20Pt_{j-1} + Pr_{j-1} \quad \text{for } j = 2, 3, 4.$$

Since Table 1.7 gives the cost per unit we have to multiply the cost per new pilot by 20 since, loosely speaking, 1 trainer unit = 20 new pilot units. This change of variables now reflects correctly the integrality of the number of trainers to be assigned. The integrality of the number of newly trained pilots follows because of the factor of 20. This change of variables corresponds exactly to what we have discussed in part (i), point 6: The variables  $x_j$  used there are precisely the number  $Pt_j$  of trainers assigned. Running the changed problem using LINDO shows the dramatic effect that our initial omission of the divisibility requirement has on the solution of the problem. If Joe Doe’s constraints are not used, then only two branches of the branch-and-bound search tree need to be explored to find and prove optimality of the integer solution displayed in the right part of Table 1.1. If Joe Doe’s cutting planes are added to the reformulated linear program, like before, branching is not necessary at all. This shows that –besides cutting-planes– the judicious choice of the decision variables of an integer program has a dramatic effect upon its solvability with standard “off-the-shelf” IP solvers.

**Table 1.11:** Airlift Solution

Variable	Value	Value
$Ca_1$	2.000	2
$Ca_2$	3.000	3
$Ca_3$	3.000	3
$Ca_4$	4.000	4
$UP_1$	10.000	10
$NP_1$	60.000	60
$NP_2$	30.000	30
$NP_3$	80.000	80
$Pi_1$	7.311	7
$Pi_2$	0.000	6
$Pi_3$	0.000	4
$Pi_4$	0.000	4
$Pn_1$	453.789	460
$Pn_2$	222.000	220
$Pn_3$	240.000	240
$Pr_2$	240.000	240
$Pr_3$	360.000	360
$Pr_4$	360.000	360
Objective	46,784.867	46,920.4

**Table 1.12.** Example of the total decomposition of a model

Planes				
Idle	New			
$UP_1 UP_2 UP_3 UP_4$	$NP_1 NP_2 NP_3 NP_4$			
0 0 0 0	200 195 190 185			min
1 -1 1 -1 1 -1 1	-1 -1 -1	= 10 = -70 = -30 = -80		

Crews				
Idle	New	Resting		
$P_{i_1} P_{i_2} P_{i_3} P_4$	$P_{n_1} P_{n_2} P_{n_3} P_{n_4}$	$P_{r_1} P_{r_2} P_{r_3} P_{r_4}$		
7 6.96.86.7	10 9.9 9.8 9.7	5 4.94.84.7		min
1 -1 1 -1 1 -1 1	.05 -1 .05 -1 .05 -1 .05		-1 -1	= 30 = -450 = -450 = -600
		1 1 1 1		= 0 = 240 = 360 = 360

(vii) If we eliminate all cargo shipment variables  $C_{aj}$  from the formulation of part (i), then the overall model “decomposes” completely into two smaller problems, one of which concerns only aircraft management and procurement and the other one of which concerns only crew management. As can be seen from Table 1.12 there are no “linking” constraints between the two parts of the problem. We can thus solve the resulting two smaller problems separately to find an optimal solution to the overall problem. Whenever such decomposition is possible, it should be exploited by the planner: Smaller problems are easier to solve than bigger ones, which helps the overall solution effort.

It would be wrong to believe that logistics problems can always be simplified in this manner. It is here entirely a consequence of the *vast* simplification of the real task that the simplified scenario of Operation Fiddle permits us to do.

In live applications of (mixed-integer) linear programming it is, however, sometimes advisable to decompose or “separate out” subproblems from a highly complex, enormous decision problem. A *total systems approach* to many practical problems –while desirable– may lead to programming problems of unmanageable size and complexity.

Breaking up a complex task into components and the optimization of the subproblems will, in general, lead to *suboptimal* solutions to the overall problem. But, evidently, one has to trade off the desirable against what can realistically be done and the interactions between the various components of a highly complex decision problem can frequently be accounted for in a satisfactory manner through other means such as e.g. simulation using the outcomes of possibly many interrelated subproblems.

## 2. The Linear Programming Problem

Linear programming is the problem of optimizing a linear function subject to finitely many linear constraints in finitely many variables. The **standard** form of the linear programming problem is

$$\min\{c\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

for data  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$  and  $\mathbf{b} \in \mathbb{R}^m$  satisfying that the rank of  $A$  equals its row rank, i.e.,  $r(A) = m$ , whereas the **canonical** form of a linear program is

$$\max\{c\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

with (possibly different) data  $c \in \mathbb{R}^{n'}$ ,  $A \in \mathbb{R}^{m' \times n'}$  and  $\mathbf{b} \in \mathbb{R}^{m'}$ . Replacing equations by two inequalities and “free” variables, i.e., variables  $x_j$  not restricted in sign, by the difference of two nonnegative variables  $x_j = x_j^+ - x_j^-$  where  $x_j^+ \geq 0$  and  $x_j^- \geq 0$ , by adding slack and/or surplus variables, etc., any linear program can be brought into either the standard or the canonical form. E.g. by bringing the linear programming problem first into canonical form and then adding slack variables, one obtains a linear program in standard form, i.e., one can indeed assume WROG (=without restriction of generality) that the rank of the constraint matrix  $A$  of the linear program in standard form equals its row rank. Apart from the first exercise, which is a pure drilling problem and the answers to which should be obvious to anyone having read the material of Chapter 2 of the book, the other exercises of this chapter are meant to elucidate the importance of inequalities and the usefulness of linear programming in data analysis.

For any  $m \times n$  matrix  $A$  we denote by  $N = \{1, \dots, n\}$  and by  $M = \{1, \dots, m\}$  the index set of the columns and the rows of  $A$ , respectively. We denote the elements of  $A$  by  $a_{j,i}^i$  where  $i \in M$  is the row index and  $j \in N$  the column index. Row  $i$  of  $A$  is the (row) vector  $\mathbf{a}^i = (a_1^i \dots a_n^i)$  and

column  $j$  of  $A$  is the (column) vector  $\mathbf{a}_j = \begin{pmatrix} a_j^1 \\ \vdots \\ a_j^m \end{pmatrix}$ . Thus the matrix  $A$  can be written as

$$A = \begin{pmatrix} \mathbf{a}^1 \\ \vdots \\ \mathbf{a}^m \end{pmatrix} = (\mathbf{a}_1 \dots \mathbf{a}_n).$$

For any subset  $C \subseteq N$  and  $R \subseteq M$  we denote by  $A_C^R$  the submatrix  $(a_{j,i}^i)_{j \in C, i \in R}$  of  $A$ . When  $C = N$  or  $R = M$  we drop the sub- or superscript, i.e., for instance  $A = A^M = A_N = A_N^M$ .  $r(A)$  denotes the rank of  $A$  and  $\det A$  is the determinant of a (square) matrix  $A$ .  $I_n$  is the  $n \times n$  identity matrix.

A formula that is useful to study structured linear programs, i.e., linear programming problems where the constraint matrix exhibits a certain “pattern”, is the following *inversion formula for partitioned matrices*.

Let  $A$  be square and partitioned as follows

$$A = \begin{pmatrix} B & D \\ C & E \end{pmatrix},$$

where  $B$  is nonsingular. It can be shown that  $A$  is nonsingular if and only if  $F = E - CB^{-1}D$  is nonsingular. Moreover, the inverse of  $A$  is given by

$$A^{-1} = \begin{pmatrix} B^{-1} + B^{-1}DF^{-1}CB^{-1} & -B^{-1}DF^{-1} \\ -F^{-1}CB^{-1} & F^{-1} \end{pmatrix}.$$

The inversion of  $A$  is thus reduced to the inversion of two “smaller” matrices. To prove the formula verify that  $A$  can be written in “product form” as

$$A = \begin{pmatrix} B & O \\ C & I_E \end{pmatrix} \begin{pmatrix} I_B & B^{-1}D \\ O & E - CB^{-1}D \end{pmatrix},$$

where  $I_B$  and  $I_E$  are identity matrices of the size of  $B$  and  $E$ , respectively, and invert the two factors separately which is trivial. Note that

$$\det A = \det B \det(E - CB^{-1}D).$$

The matrix  $E - CB^{-1}D$  is called the *Schur complement* of  $B$  in  $A$ . If we assume that  $E$  is nonsingular in the above partitioning of  $A$ , then the product form of  $A$  becomes

$$A = \begin{pmatrix} I_B & D \\ O & E \end{pmatrix} \begin{pmatrix} B - DE^{-1}C & O \\ E^{-1}C & I_E \end{pmatrix},$$

from which one can determine the inverse of  $A$  if the Schur complement  $B - DE^{-1}C$  of  $E$  in  $A$  is nonsingular. Moreover,

$$\det A = \det E \det(B - DE^{-1}C).$$

For any two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , say, we write  $\mathbf{x} \geq \mathbf{y}$  if  $x_j \geq y_j$  for all  $j = 1, \dots, n$ ,  $\mathbf{x} \not\geq \mathbf{y}$  to signal the existence of an index  $j \in \{1, \dots, n\}$  such that  $x_j < y_j$ ,  $\mathbf{x} > \mathbf{y}$  if  $x_j > y_j$  for all  $j = 1, \dots, n$ , and  $\mathbf{x} = \mathbf{y}$ , of course, if we have equality for all components. *Scaling* a vector  $\mathbf{x} \in \mathbb{R}^n$  means multiplying all components of  $\mathbf{x}$  by a positive scalar. Other notation can be found in the book.

## 2.1 Exercises

---

### \*Exercise 2.0

(i) Consider the following linear program (LP):

$$(LP) \quad \begin{array}{lllllll} \min & 7x_1 & +3x_2 & -4x_3 & & +x_5 & \\ \text{s.t.} & x_1 & +x_2 & & -x_4 & & = 10 \\ & 7x_1 & & +3x_3 & & +x_5 & = 20 \\ & x_1 & & & & -4x_5 & \leq 0 \\ & & x_2 & & +x_4 & & \leq 15 \\ & x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \text{ free}, & x_5 \text{ free} & \end{array}$$

1. Bring (LP) into canonical form and specify its data in matrix/vector form.

2. Bring (LP) into standard form and specify its data in matrix/vector form.

(ii) Consider the following  $3 \times 5$  matrix  $A$  and the  $5 \times 2$  matrix  $B$ :

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

1. Find the ranks  $r(A)$  and  $r(B)$ . Calculate the matrix products  $AB$  and  $B^T A^T$ .

2. Let  $R = \{2, 3\}$  and  $C = \{3, 4\}$ . Write down the submatrix  $A_C^R$  and calculate  $\det A_C^R$ .

---

### \*Exercise 2.1

(i) Suppose we want to solve the linear optimization problem without inequalities

$$\min\{cx : Ax = b\},$$

where  $A$  is any  $m \times n$  matrix. Show:

1. If there exist  $x^1 \neq x^2$  with  $Ax^1 = Ax^2 = b$  and  $c x^2 > c x^1$ , say, then the minimum is not bounded from below.
2. If there exist  $x \in \mathbb{R}^n$  with  $Ax = b$  and the minimum is bounded from below, then  $c x = \text{const}$  for all solutions  $x$  to  $Ax = b$  for some number const.

(ii) Given a linear programming problem in standard form with  $n$  variables and  $m$  equations, i.e.,

$$\min\{cx : Ax = b, x \geq 0\},$$

show that the linear programming problem in canonical form  $\max\{-cx : Ax \leq b, ax \leq a_0, x \geq 0\}$  solves the original problem where  $ax \leq a_0$  is a suitably chosen additional inequality. Find one that works!

---

(i) Let  $y = x^1 - x^2$ . Then  $Ay = 0$  and  $cy < 0$ . Thus  $x(\lambda) = x^1 + \lambda y$  satisfies  $Ax(\lambda) = b$  for all  $\lambda \geq 0$  and  $c x(\lambda) \rightarrow -\infty$  for  $\lambda \rightarrow +\infty$ . Thus if there is a feasible  $x \in \mathbb{R}^n$  and the minimum is bounded, then necessarily  $c x$  is some constant for all feasible solutions to  $Ax = b$ , i.e., a linear optimization problem **without inequalities** is mathematically trivial.

(ii) Let  $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ . Introducing a “dummy” slack variable vector we write equivalently

$$\begin{aligned} \mathcal{X} &= \{x \in \mathbb{R}^n : \exists s \in \mathbb{R}^m \text{ s.t. } Ax + s = b, x \geq 0, s = 0\} \\ &= \{x \in \mathbb{R}^n : \exists s \in \mathbb{R}^m \text{ s.t. } s = b - Ax, x \geq 0, es \leq 0, s \geq 0\} \\ &= \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0, -eAx \leq -eb\}. \end{aligned}$$

Since  $\min\{\mathbf{c}\mathbf{x} : \mathbf{x} \in \mathcal{X}\} = -\max\{-\mathbf{c}\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$  we have shown that solving the linear programming problem  $\max\{-\mathbf{c}\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{a}\mathbf{x} \leq \mathbf{a}_0\}$  where  $\mathbf{a} = -\mathbf{e}A$  and  $a_0 = -\mathbf{e}\mathbf{b}$  one solves also the original linear programming problem, where  $\mathbf{e} = (1, \dots, 1) \in \mathbb{R}^m$  is a row vector of  $m$  ones.

---

### \*Exercise 2.2

- (i) *Describe the set  $\{\mathbf{x} \in \mathbb{R}^n : |x_j| \leq 1 \text{ for } 1 \leq j \leq n\}$  by way of linear inequalities.*
  - (ii) *Describe the set  $\{\mathbf{x} \in \mathbb{R}^n : \sum_{j=1}^n |x_j| \leq 1\}$  by way of linear inequalities.*
- 

**(i)** The constraint  $|x_j| \leq 1$  is equivalent to  $-1 \leq x_j \leq 1$  for any  $j$  and hence

$$\{\mathbf{x} \in \mathbb{R}^n : -1 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$$

is a linear description of the set  $\{\mathbf{x} \in \mathbb{R}^n : |x_j| \leq 1 \text{ for } 1 \leq j \leq n\}$ . Alternatively, we can write  $x_j = x_j^+ - x_j^-$  with  $x_j^+ \geq 0$  and  $x_j^- \geq 0$ . The set  $\{\mathbf{x} \in \mathbb{R}^n : |x_j| \leq 1 \text{ for } 1 \leq j \leq n\}$  is the orthoprojection of the set

$$\{(\mathbf{x}^+, \mathbf{x}^-) \in \mathbb{R}^{2n} : x_j^+ + x_j^- \leq 1, x_j^+ \geq 0, x_j^- \geq 0 \text{ for } j = 1, \dots, n\},$$

which is also a linearization of the original set in a higher-dimensional space.

**(ii)** Let  $\Delta$  be the  $2^n \times n$  matrix the rows of which correspond to all vectors with  $n$  components equal to  $+1$  or  $-1$ . We claim that

$$\{\mathbf{x} \in \mathbb{R}^n : \sum_{j=1}^n |x_j| \leq 1\} = \{\mathbf{x} \in \mathbb{R}^n : \Delta\mathbf{x} \leq \mathbf{e}\}$$

where  $\mathbf{e}$  is the vector with  $2^n$  components equal to one. Let  $\mathbf{x} \in \{\mathbf{x} \in \mathbb{R}^n : \sum_{j=1}^n |x_j| \leq 1\}$  and  $\delta_j \in \{1, -1\}$  for  $1 \leq j \leq n$  be arbitrary. Then

$$\sum_{j=1}^n \delta_j x_j \leq \sum_{j=1}^n |\delta_j| |x_j| = \sum_{j=1}^n |x_j| \leq 1$$

and thus  $\Delta\mathbf{x} \leq \mathbf{e}$  is satisfied. On the other hand, let  $\mathbf{x} \in \mathbb{R}^n$  be such that  $\Delta\mathbf{x} \leq \mathbf{e}$ . If  $x_j \geq 0$  let  $\delta_j = 1$ , while if  $x_j < 0$  let  $\delta_j = -1$ . Thus

$$1 \geq \sum_{j=1}^n \delta_j x_j = \sum_{j=1}^n |x_j|$$

and the proof is complete. Alternatively, we write like in the previous exercise  $x_j = x_j^+ - x_j^-$  where  $x_j^+ \geq 0$  and  $x_j^- \geq 0$ . The orthoprojection of

$$\{(\mathbf{x}^+, \mathbf{x}^-) \in \mathbb{R}^{2n} : \sum_{j=1}^n (x_j^+ + x_j^-) \leq 1, x_j^+ \geq 0, x_j^- \geq 0 \text{ for } j = 1, \dots, n\}$$

is the set  $\{x \in \mathbb{R}^n : \sum_{j=1}^n |x_j| \leq 1\}$ . The second linear description in the higher-dimensional space requires only  $2n + 1$  inequalities, while in the original space exponentially many inequalities are needed to obtain a linear description.

---

### \*Exercise 2.3

Prove that the constraint matrix of the **transportation problem**, i.e., the matrix given by the constraints

$$\sum_{j=1}^m x_j^i = a_i \text{ for } i = 1, \dots, n, \quad \sum_{i=1}^n x_j^i = b_j \text{ for } j = 1, \dots, m,$$

has a density of  $200/(n+m)\%$  and a rank of  $m+n-1$ .

---

The matrix of the constraints of the transportation problem is

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix},$$

where we have ordered the variables  $x_j^i$  sequentially as in

$$x_1^1, x_2^1, \dots, x_m^1, x_1^2, x_2^2, \dots, x_m^2, \dots, x_1^n, x_2^n, \dots, x_m^n.$$

Thus every column of  $A$  has precisely two entries equal to one, the rest is zero. Consequently, the density of  $A$  equals  $2(nm)/((n+m)(nm)) = 2/(n+m)$  or  $200/(n+m)\%$ .

Adding the top  $n$  rows of  $A$  we obtain a vector of  $nm$  ones and likewise if we add the bottom  $m$  rows of  $A$ . Thus there exists a nonzero  $\lambda \in \mathbb{R}^{n+m}$  such that  $\lambda A = 0$  and thus  $r(A) \leq n+m-1$ .

To prove that equality is attained it suffices to exhibit a nonsingular submatrix of  $A$  of size  $(n+m-1) \times (n+m-1)$ . We drop the  $n$ -th row of  $A$ . Then the last  $m$  rows contain the identity matrix  $I_m$  in the  $m$  last columns. By choosing e.g. the columns corresponding to  $x_1^1, x_1^2, \dots, x_1^{n-1}$  we thus find a lower triangular submatrix of size  $(n+m-1) \times (n+m-1)$  of  $A$  having all entries on its main diagonal equal to one and the proof is complete.

---

### \*Exercise 2.4

In multiple linear regression one is given  $m$  observations  $x_j^i$  for  $n$  independent variables  $x_j$  as well as  $m$  observations  $y^i$  on some dependent variable  $y$  where  $1 \leq i \leq m$  and  $1 \leq j \leq n$ . On the basis of

these observations one tries to establish a linear relation of the form

$$y = \beta_0 + \sum_{j=1}^n x_j \beta_j + \varepsilon$$

where  $\beta_j$  for  $0 \leq j \leq n$  are some coefficients and  $\varepsilon$  is an error term. In classical statistics one estimates the  $\beta_j$  from the given numerical data by minimizing the **sum of squared errors**, i.e., one determines  $\beta_0, \beta_1, \dots, \beta_n$  such that  $\sum_{i=1}^m (y^i - \beta_0 - \sum_{j=1}^n \beta_j x_j^i)^2$  is minimized ( $\ell_2$ -regression).

- (i) Formulate the problem of finding  $\beta_0, \beta_1, \dots, \beta_n$  which minimize the **sum of absolute errors** as a linear program. (This is called MSAE (or MAD) or  $\ell_1$ -regression).
  - (ii) Formulate the problem of finding  $\beta_0, \beta_1, \dots, \beta_n$  which minimize the **maximum absolute error** as a linear program. (This is called Chebycheff or  $\ell_\infty$ -regression).
- 

(i) To minimize the sum of absolute errors we must find  $\beta_0, \beta_1, \dots, \beta_n$  such that

$$\sum_{i=1}^m |y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j|$$

is minimized. So let  $\varepsilon_i^+, \varepsilon_i^- \geq 0$  be “new” variables. Then we get the linear program

$$\begin{aligned} \min \quad & \sum_{i=1}^m \varepsilon_i^+ + \sum_{i=1}^m \varepsilon_i^- \\ \text{s.t.} \quad & \beta_0 + \sum_{j=1}^n x_j^i \beta_j + \varepsilon_i^+ - \varepsilon_i^- = y^i \quad \text{for } 1 \leq i \leq m, \\ & \varepsilon_i^+ \geq 0, \quad \varepsilon_i^- \geq 0 \quad \text{for } 1 \leq i \leq m. \end{aligned}$$

A different way to formulate  $\ell_1$ -regression as a linear program goes as follows. Let  $\varepsilon_i$  for  $i = 1, \dots, m$  be new “free” variables. Then the linear program

$$\begin{aligned} \min \quad & \sum_{i=1}^m \varepsilon_i \\ \text{s.t.} \quad & \beta_0 + \sum_{j=1}^n x_j^i \beta_j + \varepsilon_i \geq y^i \quad \text{for } 1 \leq i \leq m, \\ & -\beta_0 - \sum_{j=1}^n x_j^i \beta_j + \varepsilon_i \geq -y^i \quad \text{for } 1 \leq i \leq m, \end{aligned}$$

solves the  $\ell_1$ -regression problem as well, because

$$\varepsilon_i \geq \max\{y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j, -y^i + \beta_0 + \sum_{j=1}^n x_j^i \beta_j\} = |y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j|.$$

In  $\ell_1$ -regression the parameters  $\beta_j$  are free variables; however, additional linear restrictions on the parameters, such as nonnegativity, can easily be incorporated in the linear programming framework. This is not the case when the sum of squared errors is minimized.

(ii) To minimize the largest absolute error we must find  $\beta_0, \beta_1, \dots, \beta_n$  such that

$$\max\{|y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j| : i = 1, \dots, m\}$$

is minimized. So let  $z$  be a new “free” variable. Then the linear program

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & \beta_0 + \sum_{j=1}^n x_j^i \beta_j + z \geq y^i \quad \text{for } 1 \leq i \leq m, \\ & -\beta_0 - \sum_{j=1}^n x_j^i \beta_j + z \geq -y^i \quad \text{for } 1 \leq i \leq m, \end{aligned}$$

solves the Chebycheff regression problem, because

$$z \geq \max\{y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j, -y^i + \beta_0 + \sum_{j=1}^n x_j^i \beta_j\} = |y^i - \beta_0 - \sum_{j=1}^n x_j^i \beta_j| \quad \text{for } i = 1, \dots, m.$$

The same remarks as under part (i) apply to Chebycheff regression.

### 3. Basic Concepts

Consider the linear programming problem in standard form

$$(LP) \quad \min\{cx : Ax = b, x \geq 0\}$$

where  $A$  is a  $m \times n$  matrix of full rank, i.e.,  $r(A) = m$ . We denote by

$$\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

the solution set of (LP). A vector  $x \in \mathcal{X}$  is called *feasible solution*, while  $x \notin \mathcal{X}$  is called *infeasible*.

Let  $N = \{1, \dots, n\}$ . Given a feasible solution  $x \in \mathcal{X}$  we denote by

$$I_x = \{j \in N : x_j > 0\} \quad \text{and} \quad N - I_x = \{j \in N : x_j = 0\},$$

the set of the positive and the zero components of  $x$ , respectively. Then  $x$  satisfies the system of linear equations  $Ax = b$ ,  $x_j = 0$  for all  $j \in N - I_x$  or

$$\begin{pmatrix} A_{I_x} & A_{N-I_x} \\ \mathbf{O} & I_{n-k_x} \end{pmatrix} \begin{pmatrix} x_{I_x} \\ x_{N-I_x} \end{pmatrix} = \begin{pmatrix} b \\ \mathbf{0} \end{pmatrix} \text{ where } k_x = |I_x| \text{ and } n = |N|. \quad (3.1)$$

$A_{I_x}$  is a  $m \times k_x$  submatrix and  $A_{N-I_x}$  a  $m \times (n - k_x)$  submatrix of  $A$  and we compute

$$r \begin{pmatrix} A_{I_x} & A_{N-I_x} \\ \mathbf{O} & I_{n-k_x} \end{pmatrix} = r(A_{I_x}) + n - k_x.$$

The system (3.1) is uniquely solvable if and only if  $r(A_{I_x}) + n - k_x = n$ , i.e., if and only if  $r(A_{I_x}) = |I_x|$ , since by assumption a solution to (3.1) exists.

For the system (3.1) to be solvable uniquely, we thus need in particular that the number of rows of (3.1) satisfies  $m + n - k_x \geq n$ , i.e.,  $|I_x| \leq m$ . We shall use the following definitions.

$x$  is called *basic feasible solution* if  $r(A_{I_x}) = |I_x|$ .

$x$  is called *degenerate basic feasible solution* if  $r(A_{I_x}) = |I_x| < m$ .

An  $m \times m$  submatrix  $B$  of  $A$  is called *basis* if  $r(B) = m$ .

A basis  $B$  is called *feasible* if  $B^{-1}b \geq 0$ .

A feasible solution  $\bar{x} \in \mathcal{X}$  is *optimal* if  $c\bar{x} \leq cx$  for all  $x \in \mathcal{X}$ .

An optimal solution  $\bar{x}$  is *finite* or *bounded* if there exists a  $K > 0$  such that  $\bar{x}_j \leq K$  for all  $j \in N$ . For any finite optimal solution  $\bar{x}$ ,  $c\bar{x} > -\infty$ .

Given a basis  $B$  we partition  $A$  and write  $Ax = Bx_B + Rx_R$ , where  $R$  is the “rest” of  $A$ , i.e., the columns of  $A$  not in the basis  $B$ . Rather than denoting  $I_B$  the columns of  $B$  and thus  $x_{I_B}$  the subvector of  $x$  corresponding to the columns of  $B$ , etc. we write  $x_B$  and  $x_R$  for short.

Multiplying the equation system  $Bx_B + Rx_R = b$  on both sides by  $B^{-1}$  we get the *equivalent* system of equations  $x_B + B^{-1}Rx_R = B^{-1}b$ .

If a basis  $B$  is feasible then it *defines* a basic feasible solution to  $Ax = b, x \geq 0$  by  $x_B = B^{-1}b$ ,  $x_R = 0$ .

The following remarks, lemma and theorem are proven in detail in the book. Several of the exercises below illustrate the proof techniques employed there.

**Remark 3.1** If  $b = 0$ , then either  $x = 0$  is an optimal solution to (LP) or the minimum of (LP) is not bounded from below.

**Lemma 1** If  $x$  is a basic feasible solution, then  $x$  has at most  $m$  positive components and the submatrix  $A_{I_x}$  defined in (3.1) can be extended to a feasible basis  $B$  that defines  $x$  by adjoining to  $A_{I_x}$  suitable columns of  $A_{N-I_x}$ .

**Remark 3.2** For every basic feasible solution  $x \in \mathcal{X}$  there exists at least one feasible basis  $B$  of  $A$  that defines  $x$  and every feasible basis defines a basic feasible solution  $x \in \mathcal{X}$ .

**Remark 3.3**  $\bar{x}$  is a basic feasible solution if and only if there exists a vector  $c \in \mathbb{R}^n$  with integer components such that  $\min \{cx : x \in \mathcal{X}\} = c\bar{x}$  and  $cx > c\bar{x}$  for all  $x \in \mathcal{X}, x \neq \bar{x}$ , i.e., the minimizer is unique.

**Theorem 1 (Fundamental Theorem of Linear Programming)** Given a linear programming problem in standard form the following statements are correct:

- (a) If there exists a vector  $x \in \mathcal{X}$ , then there exists a basic feasible solution  $\bar{x} \in \mathcal{X}$ .
- (b) If there exists a finite optimal solution  $x \in \mathcal{X}$ , then there exists an optimal basic feasible solution  $x^* \in \mathcal{X}$ .

**Remark 3.4** From the viewpoint of (very) classical mathematics, the fundamental theorem “solves” the linear programming problem since there are at most

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

possible bases and hence at most that many feasible bases.

Since basic feasible solutions play an important role in linear programming we adopt the following notation.

1. We denote by  $B$  any basis of  $A$ , i.e., any  $m \times m$  nonsingular submatrix of  $A$ , and by  $R$  the submatrix of  $A$  given by the columns of  $A$  not in  $B$ , i.e., the “rest” of  $A$ . Any column in a basis  $B$  is characterized by its *position* in the basis and by its *original* column index in the numbering  $1, \dots, n$  of the columns of  $A$ .
2. Rather than writing  $I_B$ , we denote by  $I = \{k_1, \dots, k_m\}$  the index set of the basic variables.  $k_i$  is the original index of the variable which belongs the  $i^{th}$  column of the basis  $B$  if we number the columns in  $B$  consecutively starting with 1. (Note that  $\ell \in I$  does not imply  $x_\ell > 0$ , because the solution  $x \in \mathcal{X}$  defined by a feasible basis  $B$  may be degenerate.)
3. Likewise, rather than writing  $x_{I_B} = (x_j)_{j \in I_B}^T$ , we write  $x_B$  to denote the vector of basic variables *in the same order* as in  $B$ .
4. For  $j \in I$ , we denote by  $p_j \in \{1, 2, \dots, m\}$  the *position number* of variable  $j$  in the basis. Here  $j$  is the *original* column index in the list of columns of  $A$ . Thus if  $k_i \in I$  is the variable in position  $i$  of the basis then  $p_{k_i} = i$  for  $i = 1, \dots, m$ .

5. The scalar  $z_B = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}$  is the objective function value given by the basis  $B$ , where  $\mathbf{c}_B$  is the row vector of the objective function coefficients of the basic variables.
6.  $N - I$  is the index set of the nonbasic variables, where  $N = \{1, 2, \dots, n\}$  is the index set of all variables. Instead of writing  $\mathbf{x}_{N-I}$ , we write  $\mathbf{x}_R = (x_j)_{j \in N-I}^T$  to mean the vector of nonbasic variables and  $\mathbf{c}_R$  for the subvector of  $\mathbf{c}$  of the components corresponding to the nonbasic variables. Nonbasic variables have, of course, no “position” numbers.
7. We shall call
- $\bar{\mathbf{b}} = \mathbf{B}^{-1} \mathbf{b} = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m)^T$  the “transformed right hand side”,
  - $\bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} = (\bar{c}_1, \bar{c}_2, \dots, \bar{c}_n)$  the “reduced cost vector”, and
  - $\mathbf{y}_j = \mathbf{B}^{-1} \mathbf{a}_j = (y_j^1, y_j^2, \dots, y_j^m)^T$  the  $j^{th}$  “transformed column” of  $A$ .

### 3.1 Exercises

---

#### \*Exercise 3.0

You are given the following linear program (LP) in standard form:

$$(LP) \quad \begin{array}{lllll} \min & -x_1 & -x_2 & & \\ \text{s.t.} & 2x_1 & +3x_2 & +x_3 & = 12 \\ & x_1 & & & = 5 \\ & x_1 & +4x_2 & & +x_5 = 16 \\ & x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_5 \geq 0. \end{array}$$

(i) How many bases does (LP) have at most? Find the exact number by enumerating all possibilities. How many of them are feasible bases?

(ii) Consider the point  $\mathbf{x} \in \mathbb{R}^5$  given by  $x_2 = 4$ ,  $x_4 = 5$ ,  $x_1 = x_3 = x_5 = 0$ .

1. Is  $\mathbf{x}$  feasible for (LP)? Is it a basic feasible solution (bfs)? Is it degenerate?

2. Starting from the basis  $B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$  given by the columns 3, 4 and 5 use the exchange procedure of the proof of Lemma 1 of the text to find a feasible basis that defines this point  $\mathbf{x} \in \mathbb{R}^5$ .

3. How many different (feasible) bases define this point  $\mathbf{x} \in \mathbb{R}^5$ ?

4. Use the proof of Remark 3.3 to find  $\mathbf{c} \in \mathbb{R}^5$  which is uniquely minimized by this point.

(iii) Consider the point  $\mathbf{x} \in \mathbb{R}^5$  given by  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 4$ ,  $x_4 = 4$ ,  $x_5 = 7$ .

1. Is  $\mathbf{x}$  feasible for (LP)? Is it a basic feasible solution (bfs) to (LP)?

2. Use the construction of the first part of Theorem 1 of the text to identify a basic feasible solution to (LP) starting from this point.

3. (Optional) Note that the objective function  $c\mathbf{x}$  of (LP) equals  $-3$  at the given point. Modify the construction of Theorem 1 of the text to identify a basic feasible solution  $\mathbf{x}^*$  to (LP) with  $c\mathbf{x}^* \leq -3$  starting from the given point.

(iv) Using the fact that  $x_3, x_4$  and  $x_5$  are slack variables for the above linear program (LP) plot the feasible set in  $\mathbb{R}^2$  and interpret the various constructions of this exercise graphically.

---

(i) Since  $m = 3$  and  $n = 5$  (LP) has at most  $\binom{5}{3} = 10$  possible bases corresponding to the subsets of  $\{1, \dots, 5\}$  with 3 elements. Let  $I_1 = \{1, 2, 3\}$ ,  $I_2 = \{1, 2, 4\}$ ,  $I_3 = \{1, 2, 5\}$ ,  $I_4 = \{1, 3, 4\}$ ,  $I_5 = \{1, 3, 5\}$ ,  $I_6 = \{1, 4, 5\}$ ,  $I_7 = \{2, 3, 4\}$ ,  $I_8 = \{2, 3, 5\}$ ,  $I_9 = \{2, 4, 5\}$  and  $I_{10} = \{3, 4, 5\}$ . Checking the corresponding submatrices, we find that all except the set  $I_8$  define nonsingular submatrices of

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 4 & 0 & 0 & 1 \end{pmatrix}.$$

Of these we find 6 feasible bases corresponding to  $I_2, I_3, I_5, I_7, I_9$  and  $I_{10}$ .

(ii) Consider the point  $\mathbf{x} \in \mathbb{R}^5$  given by  $x_2 = 4, x_4 = 5, x_1 = x_3 = x_5 = 0$ .

1. The point  $\mathbf{x}$  satisfies all constraints of (LP) and thus is feasible with  $I_x = \{2, 4\}$ . Since  $\mathbf{x} \geq \mathbf{0}$  and  $|I_x| = r(\mathbf{A}_{I_x}) = 2$  the point  $\mathbf{x}$  is a basic feasible solution. Since  $|I_x| = 2 < m = 3 = r(\mathbf{A})$  it is a degenerate basic feasible solution.
2. Since  $\mathbf{B}$  contains already the column  $a_4$  we need to find out whether or column  $a_2$  can be exchanged for a basic column. Since

$$a_2 = \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix} = \mathbf{B}\lambda \quad \text{with} \quad \lambda = \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix}$$

we can replace either column  $a_3$  or  $a_5$  by  $a_2$  to get a basis that defines  $\mathbf{x}$ .

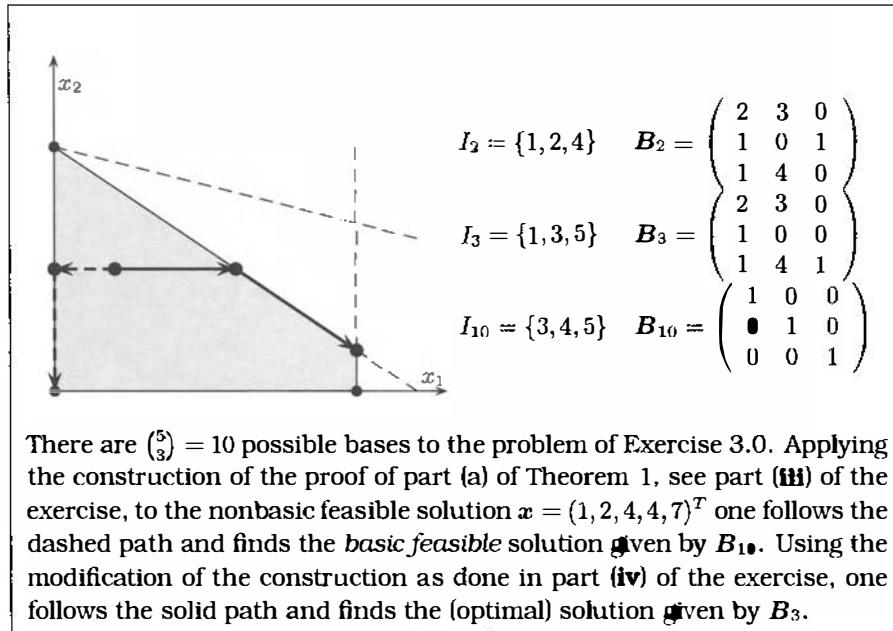
3. There are 3 different bases defining  $\mathbf{x}$ , namely those corresponding to  $I_2, I_7$  and  $I_9$ .
4. Corresponding to the proof of Remark 3.3 we set  $c_j = 0$  for all  $j \in I_x$  and  $c_j = 1$  for all  $j \in N - I_x$ . Thus  $c_2 = c_4 = 0$  and  $c_1 = c_3 = c_5 = 1$  works, i.e., after elimination of  $x_3$  and  $x_5$

$$x_1 + x_3 + x_5 = 38 - 2x_1 - 7x_2.$$

The function  $-2x_1 - 7x_2$  is uniquely minimized in the point  $\mathbf{x}$  as can be verified on the graph of the figure below (see part (iv)).

(iii) Consider the point  $\mathbf{x} \in \mathbb{R}^5$  given by  $x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 4, x_5 = 7$ .

1. The point  $\mathbf{x}$  satisfies all constraints of (LP) and thus is feasible with  $I_x = \{1, 2, 3, 4, 5\}$ . Since  $|I_x| = 5 > r(\mathbf{A}_{I_x}) = 3$  the point  $\mathbf{x}$  is not a basic feasible solution.



**Fig. 3.1.** Constructions for Exercise 3.0

2. Since  $|I_x| = 5$  and  $r(A_{I_x}) = 3$  there exists  $\lambda \in \mathbb{R}^5$ ,  $\lambda \neq 0$ , such that  $A_{I_x}\lambda = 0$ , e.g.  $\lambda^T = (-1 \ 0 \ 2 \ 1 \ 1)$ . From  $x + \theta\lambda \geq 0$  we get for  $\theta = 1$  a new point  $y \in \mathbb{R}^5$  with  $|I_y| = 4 < |I_x|$ , namely  $y^T = (0 \ 2 \ 6 \ 5 \ 8)$ , which by construction is feasible. Since, however,  $|I_y| = 4 > 3$  the point  $y$  is not basic and we have to iterate. Set  $x = y$ . Since again  $|I_x| > r(A_{I_x})$ , there exist  $\lambda \in \mathbb{R}^5$ ,  $\lambda \neq 0$ , such that  $\lambda_1 = 0$  and  $A_{I_x}\lambda = 0$ , e.g.  $\lambda^T = (0 \ -1 \ 3 \ 0 \ 4)$ . From  $x + \theta\lambda \geq 0$  we get for  $\theta = 2$  a new point  $y \in \mathbb{R}^5$  with  $|I_y| = 3 < |I_x|$ , namely  $y^T = (0 \ 0 \ 12 \ 5 \ 16)$ , which by construction is feasible and basic.
3. To ensure that in the construction of part (iii) the new points improve the objective function we need to evaluate  $c\lambda$  in the construction, since the new point is of the form  $x + \theta\lambda$ . In the first step we find  $c\lambda = (-1)(-1) = 1$ , thus the objective function gets worse. We remedy the situation by “flipping” the sign of  $\lambda$  and work with  $\lambda^T = (1 \ 0 \ -2 \ -1 \ -1)$  instead. Now from  $x + \theta\lambda \geq 0$  we get for  $\theta = 2$  a new point  $y \in \mathbb{R}^5$  with  $|I_y| = 4 < |I_x|$ , namely  $y^T = (3 \ 2 \ 0 \ 2 \ 5)$ , which by construction is feasible and  $cy = -5 < -3$ . Since, however,  $|I_y| = 4 > 3$  the point  $y$  is not basic and we have to iterate. Set  $x = y$ . Since again  $|I_x| > r(A_{I_x})$ , there exist  $\lambda \in \mathbb{R}^5$ ,  $\lambda \neq 0$ , such that  $\lambda_3 = 0$  and  $A_{I_x}\lambda = 0$ , e.g.  $\lambda^T = (-3 \ 2 \ 0 \ 3 \ -5)$ . Evaluating  $c\lambda = 1 > 0$  we see that we have to flip the sign of  $\lambda$  to move in the improving direction, i.e., we have to work with  $\lambda^T = (3 \ -2 \ 0 \ -3 \ 5)$  instead. From  $x + \theta\lambda \geq 0$  we get for  $\theta = \frac{2}{3}$  a new point  $y \in \mathbb{R}^5$  with  $|I_y| = 3 < |I_x|$ , namely  $y^T = (5 \ \frac{2}{3} \ 0 \ 0 \ 8\frac{1}{3})$ , which by construction is feasible and basic with  $cy = -5\frac{2}{3} < -5$ .

(iv) Various parts of the constructions of this exercise are displayed in Figure 3.1.

### Exercise 3.1

Show that  $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ , where  $A$  is an  $m \times n$  matrix and  $b \in \mathbb{R}^m$  is a column vector, is a convex set of  $\mathbb{R}^n$ , i.e., that for  $x_1, x_2 \in \mathcal{X}$  we have that for all  $0 \leq \mu \leq 1$   $\mu x_1 + (1 - \mu)x_2 \in \mathcal{X}$ .

Let  $x^1, x^2 \in \mathcal{X}$  and  $x = \mu x^1 + (1 - \mu)x^2$  where  $0 \leq \mu \leq 1$ . To show that  $x \in \mathcal{X}$  we have to show that (i)  $Ax = b$  and (ii)  $x \geq 0$ . To show (i) we calculate

$$Ax = A(\mu x^1 + (1 - \mu)x^2) = \mu Ax^1 + (1 - \mu)Ax^2.$$

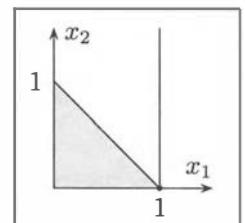
Since  $x^1 \in \mathcal{X}$  and  $x^2 \in \mathcal{X}$  we have  $Ax^1 = b$  and  $Ax^2 = b$  and thus  $Ax = \mu b + (1 - \mu)b = b$ . To prove (ii) we have that since  $x^1, x^2 \in \mathcal{X}$ ,  $x^1 \geq 0$  and  $x^2 \geq 0$  and since  $0 \leq \mu \leq 1$ ,  $1 - \mu \geq 0$ . Thus,  $\mu x^1 \geq 0$  and  $(1 - \mu)x^2 \geq 0$  and hence  $x = \mu x^1 + (1 - \mu)x^2 \geq 0$ .

### Exercise 3.2

Show (possibly by example) that if  $x$  is a degenerate basic feasible solution, then the extension procedure used in the proof of Lemma 1 need not be unique (i.e., in the case of degeneracy there exist in general several feasible bases which define the same solution vector).

On the other hand, let  $x \in \mathcal{X}$  be such that there exist two different bases that define  $x$ . Show that  $x$  is a degenerate basic feasible solution.

Suppose that  $x$  is a degenerate basic feasible solution, i.e.,  $r(A_{I_x}) = |I_x| = k_x < m$ . Let us index the columns of  $A_{I_x}$  by  $a_1, \dots, a_{k_x}$ . Following the procedure in the proof of Lemma 1 we bring all columns of  $A_{I_x}$  into the basis and WROG we write the basis as  $B = (a_1, \dots, a_{k_x}, b_{k_x+1}, \dots, b_m)$ . Repeating the same procedure but starting with a basis  $B'$  different from  $B$  we end up after bringing all the columns of  $A_{I_x}$  into the basis with



$$B' = (a_1, \dots, a_{k_x}, b'_{k_x+1}, \dots, b'_m)$$

which in general is different from  $B$ . Obviously, this cannot happen in the case  $k_x = m$  since in that case, the basis consists of all columns of  $A_{I_x}$ . As an example consider the LP with

$$\mathcal{X} = \{x \in \mathbb{R}^4 : x_1 + x_2 + x_3 = 1, x_1 + x_4 = 1, x \geq 0\}.$$

The point  $(1, 0, 0, 0)$  is a degenerate basic feasible solution and the bases  $B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$  and  $B' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$  correspond to this point as is easily verified; see also the figure.

On the other hand let  $x$  be a basic feasible solution and suppose that there exist two bases  $B_1$  and  $B_2$ ,  $B_1 \neq B_2$  that define  $x$ . Since  $x$  is a basic feasible solution, we have that  $r(A_{I_x}) = |I_x| \leq m$ . Let  $I_1$  and  $I_2$  be the column sets corresponding to the bases  $B_1$  and  $B_2$ , respectively. Since  $B_1 \neq B_2$

we have  $I_1 \neq I_2$ . Since  $I_x \subseteq I_1$  and  $I_x \subseteq I_2$  it follows that  $|I_x| < |I_1| = |I_2| = m$  and thus  $\mathbf{x}$  is degenerate.

---

### \*Exercise 3.3

*Write the problem  $\min\{\max\{\mathbf{c}\mathbf{x} - c_0, \mathbf{d}\mathbf{x} - d_0\} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  as a linear program.*

---

Let  $\mathbf{x}^T = (x_1, \dots, x_n)$  and introduce a new variable  $x_{n+1}$ . Then

$$\min\{x_{n+1} : \mathbf{c}\mathbf{x} - x_{n+1} \leq c_0, \mathbf{d}\mathbf{x} - x_{n+1} \leq d_0, \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

is a linear programming formulation of the problem. To prove it, we can assume that

$$\mathcal{X}^\leq = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \neq \emptyset,$$

since otherwise the LP formulation has no feasible solutions either. Suppose that the original problem has an unbounded optimum. Then there exist  $\mathbf{x}(\lambda) \in \mathcal{X}^\leq$  such that

$$\max\{\mathbf{c}\mathbf{x}(\lambda) - c_0, \mathbf{d}\mathbf{x}(\lambda) - d_0\} \rightarrow -\infty \quad \text{for } \lambda \rightarrow +\infty.$$

Setting  $x_{n+1}(\lambda) = \max\{\mathbf{c}\mathbf{x}(\lambda) - c_0, \mathbf{d}\mathbf{x}(\lambda) - d_0\}$  we have a family of feasible solutions for which the objective function of the linear programming problem is not bounded from below. Vice versa, when the linear program is not bounded from below, then neither is the original problem. The case of a finite optimum solution goes likewise.

---

### \*Exercise 3.4

(i) Consider the linear programming problem

$$\max\{x_1 + x_2 : sx_1 + x_2 \leq t, x_1 \geq 0, x_2 \geq 0\}.$$

Find values for  $s$  and  $t$  such that this linear program has (a) a finite optimum solution, (b) no feasible solution at all and (c) an unbounded optimum.

(ii) Prove or disprove: If the linear program  $\max\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  has an unbounded optimal solution, then the linear programming problem  $\max\{x_k : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  has an unbounded optimum solution for some subscript  $k$ . Does the reverse statement hold? Why or why not?

---

(i) To analyze the various possibilities we encourage you to graph the solution space. For  $s = t = 1$  we have a finite optimum solution, for  $s = 1$  and  $t = -1$  the solution space is empty and for  $s = -1$ ,  $t = 0$  we have an unbounded optimum.

**(iii)** If  $\max\{cx : Ax \leq b, x \geq 0\}$  has an unbounded optimum solution then there exist feasible solutions  $x(\lambda)$  such that  $cx(\lambda) \rightarrow +\infty$  for  $\lambda \rightarrow +\infty$ . Since  $x(\lambda) \geq 0$  it follows that  $x_k(\lambda) \rightarrow +\infty$  for some  $k$  and thus the assertion is correct. The reverse statement is wrong. The linear program

$$\max\{-x_1 - x_2 : x_1 \geq 0, x_2 \geq 0\}$$

has the unique solution  $x_1 = x_2 = 0$ , but there are feasible solutions where  $x_1, x_2$  or both can be arbitrarily large in value.

---

### \*Exercise 3.5

Let  $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ . Prove:  $x \in \mathcal{X}$  is a basic feasible solution if and only if

$$x = \mu x^1 + (1 - \mu)x^2 \text{ for } x^1, x^2 \in \mathcal{X} \text{ and } 0 \leq \mu \leq 1 \text{ implies that } x = x^1 = x^2,$$

i.e., that  $x$  is an “extreme point” of  $\mathcal{X}$  (see Definition EP of Chapter 7.2).

---

Let  $x$  be a basic feasible solution and  $I = \{j \in N : x_j > 0\}$ . Then for any  $x^1, x^2 \in \mathcal{X}$  such that  $x = \mu x^1 + (1 - \mu)x^2$  for some  $0 \leq \mu \leq 1$ ,  $x_j^1 = x_j^2 = 0$  for all  $j \in N - I$ . It follows that

$$A_I x_I = A_I x_I^1 = A_I x_I^2 = b$$

and thus  $A_I(x_I - x_I^1) = A_I(x_I - x_I^2) = 0$ . Since  $x$  is basic feasible,  $r(A_I) = |I|$ . Consequently,  $x_I = x_I^1 = x_I^2$  and thus  $x = x^1 = x^2$ .

On the other hand, suppose that  $x$  is feasible, but not basic feasible. As before let  $I = \{j \in N : x_j > 0\}$ . Then  $r(A_I) < |I|$  and there exists  $\lambda_I \neq 0_I$  such that  $A_I \lambda_I = 0$ . Since  $x_I > 0_I$  we can scale  $\lambda_I$  such that  $x_I + \lambda_I \geq 0_I$  and  $x_I - \lambda_I \geq 0_I$ . Let  $\lambda \in \mathbb{R}^n$  be the trivial extension of  $\lambda_I$  where  $\lambda_j = 0$  for all  $j \in N - I$ . It follows that  $x^1 = x + \lambda \geq 0$ ,  $x^2 = x - \lambda \geq 0$  and  $Ax^1 = Ax^2 = b$ , i.e.,  $x^1, x^2 \in \mathcal{X}$ . But then  $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$ , which is a contradiction.

## 4. Five Preliminaries

For every basic feasible solution  $\mathbf{x} \in \mathcal{X}$  we have by Lemma 1 a feasible basis  $\mathbf{B}$ . For every feasible basis  $\mathbf{B}$  with index set  $I$  we have the reduced system

$$\mathbf{x}_B + \mathbf{B}^{-1}\mathbf{R}\mathbf{x}_R = \bar{\mathbf{b}} \quad (4.1)$$

where  $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ . Hence a basic feasible solution  $\mathbf{x} \in \mathcal{X}$  is defined by

$$x_\ell = \bar{b}_{p_\ell} \quad \text{for all } \ell \in I, \quad x_\ell = 0 \quad \text{for all } \ell \in N - I,$$

where  $p_\ell$  is the position number of the variable  $\ell \in I$ . If  $I = \{k_1, \dots, k_m\}$ , we can write equivalently

$$x_{k_i} = \bar{b}_i \quad \text{for all } 1 \leq i \leq m, \quad x_\ell = 0 \quad \text{for all } \ell \in N - I.$$

For short, we write  $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ ,  $\mathbf{x}_R = \mathbf{0}$ .

**Sufficient Optimality Criterion:** Let  $\mathbf{B}$  be a feasible basis,  $\mathbf{c}_B$  be the subvector of the row vector  $\mathbf{c}$  corresponding to the basic variables,  $\mathbf{c}_R$  be the rest and  $\mathbf{z}_B = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}$ . If

$$\bar{\mathbf{c}}_R = \mathbf{c}_R - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{R} \geq \mathbf{0},$$

then the basic feasible solution  $\mathbf{x}$  defined by  $\mathbf{B}$  is optimal.

**Unboundedness Criterion:** Let  $\mathbf{B}$  be a feasible basis and  $I$  be the index set of basic variables. If there exists a  $j \in N - I$  such that

$$(i) \ c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j < 0 \quad \text{and} \quad (ii) \ \mathbf{B}^{-1} \mathbf{a}_j \leq \mathbf{0},$$

then the linear programming problem (LP) has an objective function value which is *not* bounded from below.

**Rank-One Update:** Let  $\mathbf{B}$  be a nonsingular  $m \times m$  matrix and  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  be any two column vectors with  $m$  components such that  $\mathbf{v}^T \mathbf{B}^{-1} \mathbf{u} \neq -1$ . Then

$$(\mathbf{B} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{B}^{-1} - \frac{1}{1 + \mathbf{v}^T \mathbf{B}^{-1} \mathbf{u}} (\mathbf{B}^{-1} \mathbf{u})(\mathbf{v}^T \mathbf{B}^{-1}).$$

Since the rank of the matrix  $\mathbf{u}\mathbf{v}^T$  for any  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  is at most one, the above formula for the inverse of the matrix  $\mathbf{B}$  when changed by  $\mathbf{u}\mathbf{v}^T$  is called a “rank-one” update.

**Basis Change:** Let  $\mathbf{B}$  be a feasible basis with index set  $I$ . If there exists a  $j \in N - I$  such that

$$(i) \ \bar{c}_j = c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j < 0 \quad \text{and} \quad (ii) \ \mathbf{B}^{-1} \mathbf{a}_j \not\leq \mathbf{0}$$

(i.e.,  $\mathbf{B}^{-1} \mathbf{a}_j$  has at least one positive entry!), then we obtain a new feasible basis  $\mathbf{B}'$  with objective function value  $\mathbf{z}_{B'}$  by replacing any column  $\mathbf{a}_\ell$  of  $\mathbf{B}$  by the column  $\mathbf{a}_j$  if the variable  $\ell \in I$  satisfies  $p_\ell = r$  where  $r$  is determined by

$$\frac{\bar{b}_r}{y_j^r} = \min \left\{ \frac{\bar{b}_i}{y_j^i} : y_j^i > 0, i = 1, 2, \dots, m \right\} \quad (4.2)$$

and  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j = (y_j^1, y_j^2, \dots, y_j^m)^T$ . The “entering” variable  $j$  gets the position  $p_j = r$  in the new basis while the variable  $\ell$  that “leaves” the basis loses its position number. Moreover,

$$z_{B'} = z_B + \theta(c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j) \leq z_B, \quad (4.3)$$

where  $\theta = \frac{\bar{b}_r}{y_j^r}$  is the minimum ratio (4.2).

We note that a basis change can be described schematically as follows where  $\rightsquigarrow$  means “changes to”:

Row	$I$	Col. $\ell$	Col. $j$	RHS	$I'$	Col. $\ell$	Col. $j$	RHS
0	*	0	$c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j$	*	*	$-(c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j)/y_j^r$	0	*
1	$k_1$	0	$y_j^1$	$\bar{b}_1$	$k_1$	$-y_j^1/y_j^r$	0	$\bar{b}_1 - \theta y_j^1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r	$k_r = \ell$	1	$y_j^r$	$\bar{b}_r$	$k_r = j$	$1/y_j^r$	1	$\theta = \bar{b}_r/y_j^r$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$k_m$	0	$y_j^m$	$\bar{b}_m$	$k_m$	$-y_j^m/y_j^r$	0	$\bar{b}_m - \theta y_j^m$

$$\begin{aligned} \mathbf{B} &= (\mathbf{a}_{k_1}, \dots, \mathbf{a}_{k_r} = \mathbf{a}_\ell, \dots, \mathbf{a}_{k_m}) \rightsquigarrow \mathbf{B}' = (\mathbf{a}_{k_1}, \dots, \mathbf{a}_{k_r} = \mathbf{a}_j, \dots, \mathbf{a}_{k_m}) \\ \mathbf{c}_B &= (c_{k_1}, \dots, c_{k_r} = c_\ell, \dots, c_{k_m}) \rightsquigarrow \mathbf{c}_{B'} = (c_{k_1}, \dots, c_{k_r} = c_j, \dots, c_{k_m}) \\ I &= \{k_1, \dots, k_r = \ell, \dots, k_m\} \rightsquigarrow I' = \{k_1, \dots, k_r = j, \dots, k_m\}. \end{aligned}$$

Algebraically, a basis change is summarized by

$$\mathbf{B}' = \mathbf{B} + (\mathbf{a}_j - \mathbf{a}_\ell)\mathbf{u}_r^T \quad \text{and} \quad \mathbf{c}_{B'} = \mathbf{c}_B + (c_j - c_\ell)\mathbf{u}_r^T \quad (4.4)$$

where  $\mathbf{u}_r^T = (0 \ \dots \ 1 \ \dots \ 0) \in \mathbb{R}^m$  is the  $r$ -th unit vector. From the rank-one update we compute

$$(\mathbf{B}')^{-1} = \mathbf{B}^{-1} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)(\mathbf{u}_r^T \mathbf{B}^{-1}). \quad (4.5)$$

Moreover, we compute the basic and nonbasic components of the new solution to be given by

$$\mathbf{x}_{B'} = \mathbf{B}^{-1}\mathbf{b} - \theta(\mathbf{y}_j - \mathbf{u}_r), \quad x_k = 0 \quad \text{for } k \in N - I', \quad (4.6)$$

where  $\theta$  as before is the value of the minimum ratio (4.2).

**Remark 4.1** If  $\theta > 0$ , then by (4.3) we have  $z_{B'} < z_B$  where  $\theta = \bar{b}_r/y_j^r$ . If the basic feasible solution defined by  $B$  is nondegenerate, then necessarily  $\theta > 0$ . If however  $\theta = 0$ , then the “old” as well as the “new” basic feasible solution are degenerate. Degeneracy of basic solutions comes about if the criterion (4.2) permits several choices. On the other hand, if  $z_{B'} = z_B$ , then by (4.3) necessarily  $\theta = 0$  since by assumption  $c_j - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{a}_j < 0$ .

**Remark 4.2** Given the mathematical correctness of the minimal ratio criterion (4.2), why do we want it, why do we need it? Since column  $j \in N - I$  satisfies  $\bar{c}_j < 0$  (see (i)) and we are minimizing, it “pays” to increase the variable  $x_j$  from its current value of zero to some (hopefully) positive value. Leaving the remaining nonbasic variables  $k \in N - I$ ,  $k \neq j$ , unchanged at their value of zero, we simply want to find the maximum value that  $x_j$  can assume while ensuring the (continued) nonnegativity

of the (currently) basic variables, i.e., we want to maximize  $x_j$  such that  $\mathbf{x}_B + x_j \mathbf{y}_j = \bar{\mathbf{b}}$ ,  $\mathbf{x}_B \geq \mathbf{0}$ . Consequently  $x_j \mathbf{y}_j \leq \bar{\mathbf{b}}$  since  $\mathbf{x}_B \geq \mathbf{0}$  or componentwise we get  $x_j \leq \min\{\bar{b}_i / y_j^i : y_j^i > 0, i = 1, \dots, m\}$ . The “simplex philosophy” is to “go all the way” and make  $x_j$  as large as possible if it pays to do so. But we also need to do so if we want to obtain a new basic feasible solution.

## 4.1 Exercises

---

### \*Exercise 4.0

You are given the following linear program (LP) in standard form:

$$(LP) \quad \begin{array}{lllll} \min & -x_1 & -x_2 & & \\ \text{s.t.} & 2x_1 & +3x_2 & +x_3 & = 12 \\ & x_1 & & & +x_4 = 5 \\ & x_1 & +4x_2 & & +x_5 = 16 \\ & x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_5 \geq 0. \end{array}$$

- (i) Write the problem in matrix/vector form  $\min\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  by supplying  $\mathbf{c}$ ,  $\mathbf{A}$ ,  $\mathbf{x}$  and  $\mathbf{b}$  explicitly. What are  $m$  and  $n$ ?
  - (ii) Consider the column index set  $I = \{3, 4, 2\}$  (in that order!). Verify that the corresponding submatrix  $\mathbf{A}_I$  is a basis  $B$  of  $\mathbf{A}$ . Write down the associated vectors/matrices  $\mathbf{x}_B$ ,  $\mathbf{x}_R$ ,  $\mathbf{c}_B$ ,  $\mathbf{c}_R$  and  $\mathbf{R}$  for the above (LP) explicitly. Find the position numbers  $p_\ell$  for all  $\ell \in I$ . Then do the following:
    - 1.) Determine the transformed right-hand-side  $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$  and the associated basic feasible solution (bfs)  $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ ,  $\mathbf{x}_R = \mathbf{0}$ .
    - 2.) Verify that the reduced costs for the basic variables are all equal to zero. Compute the reduced cost vector  $\bar{\mathbf{c}}_R = \mathbf{c}_R - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{R}$  for the nonbasic variables. Is the bfs defined by  $B$  optimal?
    - 3.) Compute the transformed column  $\mathbf{y}_j = \mathbf{B}^{-1} \mathbf{a}_j$  for  $j = 1$ . Is the (LP) unbounded? Show that the variable  $x_3$  must leave the basis  $B$  when variable  $x_1$  enters. What is the value of  $\theta$  and the value of the row index  $r$  of formula (4.2)?
    - 4.) Use formula (4.4) to compute the new basis  $B'$  and the new vector  $\mathbf{c}_{B'}$  that result when  $x_1$  enters the basis. Use formula (4.5) to compute  $(B')^{-1}$ . Update  $I$  and the position numbers of the basic variables.
    - 5.) Use formula (4.6) to compute the new bfs  $\mathbf{x}_{B'} = (B')^{-1}\mathbf{b}$  and repeat steps 2.), ..., 5.) until you can stop because of optimality or unboundedness.
    - 6.) Like in Exercise 3.0 interpret your “moves” graphically.
-

(i) In the case of problem (LP)  $m = 3$ ,  $n = 5$  and the problem in matrix form is

$$\min \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} 2 & 3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 4 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 5 \\ 16 \end{pmatrix},$$

where  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $x_3 \geq 0$ ,  $x_4 \geq 0$  and  $x_5 \geq 0$ .

(ii) For  $I = \{3, 4, 2\}$  we have  $A_I = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix}$  and  $\det A_I = 4$ , so  $A_I$  is a basis of  $A$ . So let  $B = A_I$ . The other quantities are

$$x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_2 \end{pmatrix}, x_R = \begin{pmatrix} x_1 \\ x_5 \end{pmatrix}, c_B = (0 \ 0 \ -1), c_R = (-1 \ 0), R = \begin{pmatrix} 2 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

The position numbers of the basic variables are  $p_2 = 3$ ,  $p_3 = 1$  and  $p_4 = 2$ .

1. We calculate  $B^{-1}$  and  $x_B = B^{-1}b$ :

$$B^{-1} = \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} \quad \text{and} \quad x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_2 \end{pmatrix} = B^{-1}b = \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 12 \\ 5 \\ 16 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \\ 4 \end{pmatrix}.$$

Consequently,  $x_2 = 4$ ,  $x_4 = 5$  and  $x_1 = x_3 = x_5 = 0$  is a (degenerate) basic feasible solution.

2. To verify the reduced cost of the basic variables we compute  $\bar{c}_B = c_B - c_B B^{-1} B = 0_B$  in full generality. Next we compute first  $c_B B^{-1} = (0 \ 0 \ -1) \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} = (0 \ 0 \ -\frac{1}{4})$ . Thus we get for the nonbasic variables

$$\bar{c}_R = c_R - c_B B^{-1} R = (-1 \ 0) - (0 \ 0 \ -\frac{1}{4}) \begin{pmatrix} 2 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} = (-\frac{3}{4} \ \frac{1}{4}),$$

i.e.,  $\bar{c}_1 = -\frac{3}{4}$  and  $\bar{c}_5 = \frac{1}{4}$ , which shows that  $B$  does not satisfy the sufficient optimality criterion.

3. For  $j = 1$  we compute  $y_1 = \begin{pmatrix} y_1^1 \\ y_1^2 \\ y_1^3 \end{pmatrix} = B^{-1} a_1 = \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{5}{4} \\ 1 \\ \frac{1}{4} \end{pmatrix}$ . Since  $y_1 \not\leq 0$ , i.e., because  $y_1$  has at least one positive component, the unboundedness test fails. From  $\bar{b}$  and  $y_1$  the least ratio calculation

$$\min\left\{\frac{0}{\frac{5}{4}}, \frac{5}{1}, \frac{4}{\frac{1}{4}}\right\} = 0$$

gives  $\theta = 0$ ,  $r = 1$ , when the variables  $x_1$  enters and  $x_3$  leaves the basis, because  $p_3 = 1$ .

4. From formula (4.4) the new basis  $B'$  and the vector  $c_{B'}$  are

$$\begin{aligned} B' &= B + (a_1 - a_3)u_1^T = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix} + \left( \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 1 & 1 & 0 \\ 1 & 0 & 4 \end{pmatrix}, \\ c_{B'} &= c_B + (c_1 - c_3)u_r^T = (0 \ 0 \ -1) + (-1 - 0)(1 \ 0 \ 0) = (-1 \ 0 \ -1). \end{aligned}$$

From formula (4.5) we calculate

$$\begin{aligned} (B')^{-1} &= B^{-1} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)(\mathbf{u}_r^T B^{-1}) \\ &= \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} - \frac{1}{\frac{5}{4}} \left( \begin{pmatrix} \frac{5}{4} \\ 1 \\ \frac{1}{4} \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) \begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix} = \begin{pmatrix} \frac{4}{5} & 0 & -\frac{3}{5} \\ -\frac{1}{5} & 1 & 0 \\ -\frac{1}{5} & 0 & \frac{1}{5} \end{pmatrix}. \end{aligned}$$

The new  $I = \{1, 4, 2\}$  and the new position numbers are  $p_1 = 1$ ,  $p_2 = 3$  and  $p_4 = 2$ .

5. From formula (4.6) we compute

$$\mathbf{x}_{B'} = \begin{pmatrix} x_1 \\ x_4 \\ x_2 \end{pmatrix} = B^{-1}\mathbf{b} - \theta(\mathbf{y}_j - \mathbf{u}_r) = B^{-1}\mathbf{b} = \begin{pmatrix} 0 \\ 5 \\ 4 \end{pmatrix},$$

i.e., because of the degeneracy of the solution, the solution did not change. We set next  $B = B'$  and note that now

$$\mathbf{x}_B = \begin{pmatrix} x_1 \\ x_4 \\ x_2 \end{pmatrix}, \mathbf{x}_R = \begin{pmatrix} x_3 \\ x_5 \end{pmatrix}, c_B = (-1 \ 0 \ -1), c_R = (0 \ 0), R = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

We compute next the new  $c_B B^{-1} = (-1 \ 0 \ -1) \begin{pmatrix} \frac{4}{5} & 0 & -\frac{3}{5} \\ -\frac{1}{5} & 1 & 0 \\ -\frac{1}{5} & 0 & \frac{1}{5} \end{pmatrix} = (-\frac{3}{5} \ 0 \ \frac{1}{5})$ . Thus we get for the reduced cost of the nonbasic variables

$$\bar{c}_R = c_R - c_B B^{-1} R = (0 \ 0) - (-\frac{3}{5} \ 0 \ \frac{1}{5}) \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = (\frac{3}{5} \ -\frac{1}{5}),$$

i.e.,  $\bar{c}_3 = \frac{3}{5}$  and  $\bar{c}_5 = -\frac{1}{5}$ , which shows that  $B$  does not satisfy the sufficient optimality criterion. Calculating  $y_5 = B^{-1}a_5$  and carrying out the least ratio calculation as before shows that  $x_4$  leaves when  $x_5$  enters the basis. The new basis with basic index set  $I = \{1, 5, 2\}$  satisfies the sufficient optimality criterion and the procedure stops with the optimal basic feasible solution  $x_1 = 5$ ,  $x_2 = \frac{2}{3}$ ,  $x_5 = 8\frac{1}{3}$ ,  $x_3 = x_4 = 0$ . The details of these calculations should be clear by now and are left to the reader.

6. The interpretation on the graph of Figure 3.1 is clear: We start at basic feasible solution  $x_1 = 0$  and  $x_2 = 4$ . In the first iteration we just change the basis and stay at that solution. In the second iteration we move to the basic feasible solution  $x_1 = 5$ ,  $x_2 = \frac{2}{3}$ , at which point we conclude optimality and stop.

**Exercise 4.1**

(i) Show that  $r(\mathbf{u}\mathbf{v}^T) \leq 1$  for any  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ .

(ii) For  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  such that  $\mathbf{v}^T \mathbf{u} \neq -1$  show  $(\mathbf{I}_m + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{I}_m - \frac{1}{1 + \mathbf{v}^T \mathbf{u}} \mathbf{u}\mathbf{v}^T$ .

---

**(i)** For  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  the product  $\mathbf{u}\mathbf{v}^T$  is an  $m \times m$  matrix whose rows are multiples of the vector  $\mathbf{v}^T$ , e.g. the  $i$ -th row of the matrix is given by  $u_i \mathbf{v}^T$ . Thus either  $\mathbf{u} = \mathbf{0}$  in which case the rank is zero, or, there exists  $1 \leq j \leq m$  with  $u_j \neq 0$  in which case each row of  $\mathbf{u}\mathbf{v}^T$  is a multiple of the  $j$ -th row, and hence the rank of the matrix is at most 1.

**(ii)** Since  $\mathbf{v}^T \mathbf{u} \neq -1$  we have that  $1 + \mathbf{v}^T \mathbf{u} \neq 0$  and thus the inverse is well defined. To prove that the inverse is indeed as given we apply the rank-one formula with  $B = \mathbf{I}_m$ . Then  $B^{-1} = \mathbf{I}_m$  and thus

$$(\mathbf{I}_m + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{I}_m - \frac{1}{1 + \mathbf{v}^T \mathbf{I}_m \mathbf{u}} (\mathbf{I}_m \mathbf{u})(\mathbf{v}^T \mathbf{I}_m) = \mathbf{I}_m - \frac{1}{1 + \mathbf{v}^T \mathbf{u}} \mathbf{u}\mathbf{v}^T.$$


---

**Exercise 4.2**

Show that if  $0 < \lambda < \theta$  then the vector  $\mathbf{x}(\lambda)$  given by  $\mathbf{x}_B(\lambda) = \bar{\mathbf{b}} - \lambda \mathbf{y}_j$ ,  $x_j(\lambda) = \lambda$  for some  $j \in N - I$ ,  $x_k(\lambda) = 0$  for all  $k \in N - I$ ,  $k \neq j$ , satisfies  $\mathbf{x}(\lambda) \in \mathcal{X}$ , but that  $\mathbf{x}(\lambda)$  is **not** a basic feasible solution to (LP) where  $\theta = \min\{\bar{b}_i/y_j^i : y_j^i > 0\}$  is the least ratio (4.2).

---

For  $\mathbf{x}(\lambda)$  to be in  $\mathcal{X}$  we have to show that  $A\mathbf{x}(\lambda) = \mathbf{b}$  and  $\mathbf{x}(\lambda) \geq \mathbf{0}$ . For the first we calculate

$$A\mathbf{x}(\lambda) = B\mathbf{x}_B(\lambda) + R\mathbf{x}_R(\lambda) = B(\bar{\mathbf{b}} - \lambda \mathbf{y}_j) + \lambda \mathbf{a}_j = BB^{-1}\mathbf{b} - \lambda BB^{-1}\mathbf{a}_j + \lambda \mathbf{a}_j = \mathbf{b}.$$

The second follows, since  $\lambda > 0$  and from  $\lambda < \theta$ ,  $\bar{b}_i - \lambda y_j^i \geq \bar{b}_i - \theta y_j^i \geq 0$  for all  $i$  with  $y_j^i > 0$ . Since  $\bar{b}_i - \lambda y_j^i \geq 0$  for all  $i$  with  $y_j^i \leq 0$  as well, we have  $\mathbf{x}(\lambda) \in \mathcal{X}$  for  $0 \leq \lambda \leq \theta$ .

To prove that  $\mathbf{x}(\lambda)$  is not basic for  $0 < \lambda < \theta$  we note that  $\mathbf{x}(\lambda) = \frac{\theta-\lambda}{\theta} \mathbf{x}(0) + \frac{\lambda}{\theta} \mathbf{x}(\theta)$ . Thus by Exercise 3.5  $\mathbf{x}(\lambda)$  is not basic. To give a different proof, suppose that  $\mathbf{x}(\lambda)$  is basic for some  $0 < \lambda < \theta$ . By Remark 3.2 there exists a vector  $\mathbf{c} \in \mathbb{R}^n$  such that  $\mathbf{c}\mathbf{x}(\lambda) > \mathbf{c}\mathbf{x}$  for all  $\mathbf{x} \in \mathcal{X}$ ,  $\mathbf{x} \neq \mathbf{x}(\lambda)$ . Thus in particular we have the inequalities  $\mathbf{c}\mathbf{x}(\lambda) > \mathbf{c}\mathbf{x}(0)$  and  $\mathbf{c}\mathbf{x}(\lambda) > \mathbf{c}\mathbf{x}(\theta)$ . Substituting  $\mathbf{c}\mathbf{x}(\lambda) = \frac{\theta-\lambda}{\theta} \mathbf{c}\mathbf{x}(0) + \frac{\lambda}{\theta} \mathbf{c}\mathbf{x}(\theta)$  in these inequalities we get  $\mathbf{c}\mathbf{x}(0) > \mathbf{c}\mathbf{x}(\theta)$  and  $\mathbf{c}\mathbf{x}(\theta) > \mathbf{c}\mathbf{x}(0)$ , which is a contradiction.

In the special case that  $\mathbf{x}(\lambda)$  is *nondegenerate* one proves that it is not basic also in a simpler way: since,  $\mathbf{x}_B(\lambda) > \mathbf{0}$  and  $x_j(\lambda) = \lambda > 0$  we have  $|I| = m + 1 > m$  and since  $r(A_I) \leq m$  it follows that  $\mathbf{x}(\lambda)$  is not a basic solution.

**Exercise 4.3**

The left box for a basis change shown above displays the index set  $I$  of the basis  $B$ , the column vectors  $B^{-1}\mathbf{a}_\ell$ ,  $B^{-1}\mathbf{a}_j$  and  $B^{-1}\mathbf{b}$  and the reduced cost  $\bar{c}_\ell$  and  $\bar{c}_j$ . Verify that the box on the right displays the same quantities in terms of the basis  $B'$ .

From (4.5) we have  $(B')^{-1} = B^{-1} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)(\mathbf{u}_r^T B^{-1})$  and thus

$$\mathbf{y}'_i = (B')^{-1}\mathbf{a}_i = B^{-1}\mathbf{a}_i - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\mathbf{u}_r^T B^{-1}\mathbf{a}_i = B^{-1}\mathbf{a}_i - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\mathbf{u}_r^T \mathbf{y}_i = \mathbf{y}_i - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)y_i^r .$$

For  $i = \ell$  since  $\mathbf{y}_\ell = \mathbf{u}_r$  we have that  $\mathbf{y}_\ell = \mathbf{u}_r - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)$ , and thus componentwise

$$y_\ell^k = \begin{cases} 0 - \frac{1}{y_j^r}y_j^k = -\frac{y_j^k}{y_j^r} & \text{if } k \neq r \\ 1 - \frac{1}{y_j^r}(y_j^r - 1) = \frac{1}{y_j^r} & \text{if } k = r \end{cases} .$$

For  $i = j$  we get  $\mathbf{y}'_j = \mathbf{y}_j - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)y_j^r = \mathbf{u}_r$ .

For the reduced costs we first calculate  $\mathbf{c}_{B'}(B')^{-1}$  as follows

$$\mathbf{c}_{B'}(B')^{-1} = (\mathbf{c}_B + (c_j - c_\ell)\mathbf{u}_r^T)(B^{-1} - \frac{1}{y_j^r}(B^{-1}\mathbf{a}_j - \mathbf{u}_r)\mathbf{u}_r^T B^{-1}) = \mathbf{c}_B B^{-1} + \frac{1}{y_j^r} \bar{c}_j \mathbf{u}_r^T B^{-1} .$$

Thus for  $\bar{c}'_\ell$  we calculate

$$\bar{c}'_\ell = c_\ell - (\mathbf{c}_B B^{-1} + \frac{1}{y_j^r} \bar{c}_j \mathbf{u}_r^T B^{-1})\mathbf{a}_\ell = c_\ell - \mathbf{c}_B B^{-1}\mathbf{a}_\ell + \frac{1}{y_j^r} \bar{c}_j \mathbf{u}_r^T \mathbf{u}_r = 0 + \frac{1}{y_j^r} \bar{c}_j = \frac{\bar{c}_j}{y_j^r} .$$

For  $\bar{c}'_j$  we have

$$\bar{c}'_j = c_j - (\mathbf{c}_B B^{-1} + \frac{1}{y_j^r} \bar{c}_j \mathbf{u}_r^T B^{-1})\mathbf{a}_j = c_j - \mathbf{c}_B B^{-1}\mathbf{a}_j - \frac{1}{y_j^r} \bar{c}_j \mathbf{u}_r^T \mathbf{y}_j = \bar{c}_j - \frac{1}{y_j^r} \bar{c}_j y_j^r = 0 .$$

Finally, for the RHS we calculate

$$(B')^{-1}\mathbf{b} = B^{-1}\mathbf{b} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\mathbf{u}_r^T B^{-1}\mathbf{b} = \bar{\mathbf{b}} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\mathbf{u}_r^T \bar{\mathbf{b}} = \bar{\mathbf{b}} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\bar{b}_r = \bar{\mathbf{b}} - \theta(\mathbf{y}_j - \mathbf{u}_r)$$

since  $\theta = \bar{b}_r/y_j^r$  and thus  $\bar{b}'_k = \begin{cases} \bar{b}_k - \theta y_j^k & \text{if } k \neq r \\ \bar{b}_r - \theta(y_j^r - 1) = \bar{b}_r - \bar{b}_r + \theta = \theta & \text{if } k = r \end{cases} .$

## 5. Simplex Algorithms

We can state now an iterative procedure for the resolution of the linear programming problem (LP) in standard form with descriptive “input data”  $m, n, A, b$  and  $c$ .

**Simplex Algorithm** ( $m, n, A, b, c$ )

**Step 0:** Find a feasible basis  $B$ , its index set  $I$  and initialize  $p_k$  for all  $k \in I$ .

**if** none exists **then**

**stop** “LP has no feasible solution”.

**else**

compute  $B^{-1}$ ,  $\bar{b} := B^{-1}b$  and initialize  $c_B$ .

**endif.**

**Step 1:** Compute  $\bar{c} := c - c_B B^{-1} A$ .

**if**  $\bar{c} \geq 0$  **then**

set  $x_B := \bar{b}$ ;  $x_R := 0$ ,

**stop** “ $x_B$  is an optimal basic feasible solution”.

**else**

(5.1) choose  $j \in \{k \in N : \bar{c}_k < 0\}$ .

**endif.**

**Step 2:** Compute  $y_j := B^{-1} a_j$ .

**if**  $y_j \leq 0$  **then**

**stop** “LP has an unbounded solution”.

**else**

compute the least ratio  $\theta := \min \left\{ \frac{\bar{b}_i}{y_j^i} : y_j^i > 0, 1 \leq i \leq m \right\}$ ,

(5.2) choose  $\ell \in I$  such that  $\frac{\bar{b}_{p_\ell}}{y_j^{p_\ell}} = \theta$  and set  $r := p_\ell$ .

**endif.**

**Step 3:** Set  $B := B + (a_j - a_\ell) u_r^T$ ,  $c_B := c_B + (c_j - c_\ell) u_r^T$ ,  $I := I - \{\ell\} \cup \{j\}$  and  $p_j := r$ .

**Step 4:** Compute  $B^{-1}$ ,  $\bar{b} := B^{-1}b$  and **go to** Step 1.

With the exception of Step 0 and the lines numbered (5.1) and (5.2), which involve “judgment”, the simplex algorithm is a deterministic computing mechanism and the question is whether or not an *answer* is found in a *finite* number of steps. By Theorem 1 finiteness and correctness of the algorithm are evidently assured

(i) if we can get started and (ii) if *no basis is repeated*.

We shall come back to the general question of finiteness below.

**Terminology:** A change of basis is called a *pivot* or a *pivot operation*. The nonbasic column selected in (5.1) is called the *pivot column*. The row selected in (5.2) is called a *pivot row*. The element  $y_j^r$  of (5.2) is called the *pivot element*. The calculation of the reduced cost vector  $\bar{c}$  is

called *pricing* or *pricing operation*. A variable  $j$  or  $x_j$  does *not price out correctly* if its reduced cost is negative.

**Reading Instructions:** In large-scale linear computation one does *not* calculate  $B^{-1}$  explicitly, because it is not necessary. The simplex algorithm really requires:

(A) Knowledge of  $\bar{b}$  for the solution vector  $x_B$  and for pivot row selection. To find  $\bar{b}$  we have to solve

$$B\bar{b} = b. \quad (5.3)$$

(B) Knowledge of  $\bar{c}$  to determine if  $x_B$  is optimal or not and to select a pivot column. To find  $\bar{c}$  we do the calculation in two steps. We first find a row vector  $u$ , say, that solves

$$uB = c_B. \quad (5.4)$$

Then we calculate  $\bar{c}_k = c_k - ua_k$  for  $k = 1, \dots, n$  which is *expensive* if  $n$  is very large, but unavoidable.

(C) Knowledge of  $y_j$  to determine unboundedness of (LP) or to select a pivot row. To find  $y_j$  we solve

$$By_j = a_j. \quad (5.5)$$

These systems of equations are solved using some form of *Gaussian elimination*; see also Chapter 7.

**Getting Started:** Multiplying the equations  $Ax = b$  by  $-1$  if necessary we can assume WROG that  $b \geq 0$ . In the *Big-M* method, we thus have a basic feasible start  $x = 0$  and  $s = b$  for the “enlarged” problem

$$\begin{aligned} (\text{Big M}) \quad & \min \sum_{j=1}^n c_j x_j + M \sum_{i=1}^m s_i \\ & \text{subject to } Ax + s = b \\ & \quad x \geq 0, s \geq 0. \end{aligned}$$

M is a “big” number e.g.  $M = 10^{30}$ , i.e. a number that is *sufficiently big to guarantee* that in any optimal solution to the original problem all artificial variables  $s_1, \dots, s_m$  assume the value zero. In the Two-Phase method we solve in Phase I the problem

$$\begin{aligned} (\text{Phase I}) \quad & \min \sum_{i=1}^m s_i \\ & \text{subject to } Ax + s = b \\ & \quad x \geq 0, s \geq 0, \end{aligned}$$

which we start as before. If at the end of (Phase I)  $\sum_{i=1}^m s_i > 0$ , then the original problem has no feasible solution. Otherwise, we switch to the original objective function and solve the linear program. In both approaches the artificial variables are simply “forgotten” as soon as they become nonbasic. In computational practice one frequently uses a “mixture” of the Big M-Method and the Two-Phase Method. Top quality commercial software also employs so-called “crash methods”, i.e., heuristic methods to find a “reasonably good” starting basis at low cost.

**Pivot Column Selection:** To make a deterministic (reproducible) choice in Step 1, line (5.1), of the algorithm we need a rule that uniquely identifies a column. The most commonly used choice rules for pivot column selection are:

- (c1) Choose  $j$  such that  $\bar{c}_j = \min\{\bar{c}_k : k \in N\}$  and break any remaining ties by choosing the one with the smallest index.
- (c2) Choose the first  $j$  for which  $\bar{c}_j < 0$ , i.e. the smallest column index that has a negative reduced cost.
- (c3) Create a stack of, say, the  $p$  most negatively priced columns, i.e. the column indices with the most negative  $\bar{c}_j$ 's. Then select the “next” column from the stack that (still) qualifies until the stack is empty at which point a “full” pricing of all columns is used to recreate the stack.  $p$  is, of course, a parameter set by the user.
- (c4) Calculate the least ratio  $\theta_k$ , say, in Step 2 for all  $k \in N$  such that  $\bar{c}_k < 0$  and choose  $j$  such that  $\bar{c}_j \theta_j = \min\{\bar{c}_k \theta_k : \bar{c}_k < 0, k \in N\}$ .

Define the norm of column  $j$  by  $n_j = \sqrt{1 + \sum_{i=1}^m (y_j^i)^2}$ . The following choice rules employ “steepest edge” criteria, since the normalization by  $n_j$  is geometrically just that; see Chapter 7.

- (c5) Use (c1), (c3) or (c4) with  $\bar{c}_j$  replaced by  $\frac{\bar{c}_j}{n_j}$ .

**Pivot Row Selection:** In Step 2, line (5.2), of the simplex algorithm, the *least ratio* need not be unique. To make the resulting selection unique, the most commonly used choice rules for pivot row selection are:

- (r1) Choose among the candidates the one with the biggest  $y_j^i$  and if the choice remains ambiguous choose the smallest row index  $r$  among these candidates.
- (r2) Choose among the candidates the one with smallest  $\ell$ , i.e. the one with  $\ell = \min\{k \in I : \frac{\bar{b}_{p_k}}{y_j^{p_k}} = \theta\}$  where  $\theta$  is the value of the least ratio.
- (r3) Choose among the candidates by randomized choice.
- (r4) Choose a row by so-called lexicographical choice rules.

Define the norm of row  $i$  by  $d_i = \sqrt{\sum_{j=1}^m (b_j^i)^2 + \sum_{j \in N-I} (y_j^i)^2}$ , where  $b_j^i$  are the elements of the basis inverse  $B^{-1}$ , and  $I$  is the basic index set.

- (r5) Choose among the rows with  $\frac{\bar{b}_r}{y_j^r} = \theta$  one for which  $\frac{\bar{b}_r}{d_r}$  is maximal, where  $\theta$  is the value of the least ratio.

The exact computation of the norms  $n_j$  and  $d_i$  is expensive. Thus approximate updating schemes must be used; see Exercise 5.11 and 6.12. The most commonly used pivot column and row selection criteria (c1) and (r1) do not guarantee the finiteness of the resulting algorithm. However, the choice rules (c2) and (r2) guarantee the finiteness of the algorithm.

**Theorem 2** Suppose that a feasible basis  $B$  exists. If the choice rules (c2) and (r2) are used for pivot column and pivot row selection, respectively, then the simplex algorithm repeats no basis and stops after a finite number of iterations.

In computational practice the choice rules (c2) and (r2) are used, however, only when “stalling” of the objective function prohibits the use of more promising choice rules, i.e., as “anti-cycling” devices.

**Upper Bounds:** Consider the linear program with *finite upper bounds*  $u_j > 0$  on all variables

$$(LP_u) \quad \min \{ \mathbf{c}x : A\mathbf{x} = \mathbf{b}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \},$$

where we assume that  $r(\mathbf{A}) = m$ . Introducing slack variables we have the standard form

$$\begin{array}{ll} \min & \mathbf{c}\mathbf{x} \\ \text{subject to} & \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} + \mathbf{s} = \mathbf{u} \\ \mathbf{x} \geq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0} \end{array} \end{array}$$

every basis  $\hat{\mathbf{B}}$  of which is  $(m+n) \times (m+n)$  and can be written in the form

$$\hat{\mathbf{B}} = \begin{pmatrix} \mathbf{B} & \mathbf{B}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_p & \mathbf{0} & \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_s \end{pmatrix}.$$

The  $p$  first columns of  $\hat{\mathbf{B}}$  are such that  $x_j$  and  $s_j$  are basic, the next  $q$  variables have  $x_j$  basic and  $s_j$  nonbasic and the last set has  $x_j$  nonbasic and  $s_j$  basic. From a column count we get  $2p+q+s = m+n$  and from a row count  $m+p+q+s = m+n$ . Consequently,  $p = m$ , the submatrix  $\mathbf{B}$  is of size  $m \times m$  and  $\det \hat{\mathbf{B}} = \pm \det \mathbf{B}$ , i.e.  $\mathbf{B}$  a basis of  $\mathbf{A}$ . We distinguish thus between nonbasic variables at their *lower bound* of zero (the usual concept) and nonbasic variables at their *upper bound* (the new concept), i.e.,

$$J_0 = \{k \in N : x_k = 0 \text{ and nonbasic}\}, \quad J_u = \{k \in N : x_k = u_k \text{ and nonbasic}\}.$$

The reduced system of Chapter 4 now reads

$$\begin{array}{ll} \min & z_B + \sum_{k \in J_0} \bar{c}_k x_k + \sum_{k \in J_u} \bar{c}_k x_k \\ \text{subject to} & \mathbf{x}_B + \sum_{k \in J_0} \mathbf{y}_k x_k + \sum_{k \in J_u} \mathbf{y}_k x_k = \bar{\mathbf{b}} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array}$$

and the basic feasible solution defined by  $\mathbf{B}$  and the partition  $(J_0, J_u)$  is

$$x_j = \bar{b}_{p_j} \text{ for } j \in I, \quad x_j = 0 \text{ for } j \in J_0, \quad x_j = u_j \text{ for } j \in J_u.$$

Note that  $\bar{\mathbf{b}} = \mathbf{B}^{-1} (\mathbf{b} - \sum_{j \in J_u} \mathbf{a}_j u_j) \geq 0$  by our choice of notation. While we could have “complemented” the variables  $x_j$  with index  $j \in J_u$  into their upper bounds  $u_j$  – e.g. by a substitution of the form  $s_j = u_j - x_j$  – we have *not* done that *explicitly* because we can do it “implicitly”. The optimality criterion becomes:

(i) If  $\bar{c}_k \geq 0$  for all  $k \in J_0$  and  $\bar{c}_k \leq 0$  for all  $k \in J_u$ , then the basis  $B$  together with the partition  $(J_0, J_u)$  defines an optimal solution.

Unboundedness cannot occur. To “change bases”, note that by the optimality criterion a basis does not “display” optimality if one of the following two situations occurs,

- (a) there exists  $j \in J_0$  with  $\bar{c}_j < 0$  or (b) there exists  $j \in J_u$  with  $\bar{c}_j > 0$ .

In case (a) we would like to *increase*  $x_j$  from its current level of zero to a positive level that, however, must also be less than or equal to its upper bound  $u_j$ . In case (b) we would like to *decrease* variable  $x_j$  from its current level  $u_j$  to a smaller value that, however, must be greater than or equal to zero. We have thus *two* types of changes to consider. To analyze case (a) we consider the problem

$$\begin{aligned} \max \quad & (-\bar{c}_j)x_j \\ \text{subject to} \quad & \mathbf{x}_B + \mathbf{y}_j x_j + \sum_{k \in J_u} \mathbf{y}_k x_k = \bar{\mathbf{b}} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}. \end{aligned}$$

We may leave the nonbasic variables  $k \in J_u$  at their respective upper bounds and thus using  $\mathbf{0} \leq \mathbf{x}_B \leq \mathbf{u}_B$  the problem reduces to

$$\begin{aligned} \max \quad & (-\bar{c}_j)x_j \\ \text{subject to} \quad & \bar{\mathbf{b}} - \mathbf{u}_B \leq \mathbf{y}_j x_j \leq \bar{\mathbf{b}} \\ & 0 \leq x_j \leq u_j, \end{aligned}$$

where  $\mathbf{u}_B$  is the upper bound vector corresponding to the variables in  $B$ . This problem has a feasible solution with  $x_j = 0$  by assumption. Thus either variable  $x_j$  can be increased to its upper bound  $u_j$  without violating the other inequalities in which case one recomputes  $\mathbf{x}_B$ , puts variable  $j$  into the “new” set  $J_u$  and one iterates. Or this is not the case. Then we have two possibilities: either a basic variable goes to zero first or a basic variable reaches its upper bound first. That is we need to consider *both*  $y_j^i > 0$  (for the first possibility) *and*  $y_j^i < 0$  (for the second possibility). The first of the two possibilities gives rise to a “normal” pivot, i.e. the basic variable leaves the basis and is put into the “new” set  $J_0$  while variable  $j$  enters the basis. In the second one, variable  $j$  enters the basis, a basic variable leaves it and enters the “new” set  $J_u$ . Then one iterates. In case (b) we need to take into account the “complementation” into the upper bound indicated above. Leaving all variables  $k \in J_u$ ,  $k \neq j$ , at their respective upper bounds the problem to be analyzed is given by

$$\begin{aligned} \max \quad & \bar{c}_j x'_j \\ \text{subject to} \quad & \bar{\mathbf{b}} - \mathbf{u}_B \leq -\mathbf{y}_j x'_j \leq \bar{\mathbf{b}} \\ & 0 \leq x'_j \leq u_j \end{aligned}$$

where  $x'_j = u_j - x_j$ . Note that we have changed the sign of  $\mathbf{y}_j$  and  $\bar{c}_j$ . Now either the variable  $x'_j$  can be increased all the way to  $u_j$  – which means the original variable  $x_j$  can be decreased to zero – without violating any of the other inequalities. In this case  $\mathbf{x}_B$  is recomputed and variable  $x_j$  leaves the set  $J_u$  and enters the set  $J_0$ . Or this is not the case and like above we have two possibilities that are analyzed in an analogous manner. See Exercise 5.12 for the details.

**Other Topics:** The other topics of this chapter discussed in the text concern explicit “updating” formulas for the basis inverse (see e.g. (4.5)), data structures, tolerances, the so-called product

product form of the basis, worst-case behavior, cycling and finiteness of simplex algorithms, direct algorithms for linear programs in canonical form, block pivots and the exploitation of special structure. Several of these topics are touched upon in the exercises.

## 5.1 Exercises

---

### Exercise 5.1

*Using precise pivot rules of your own choice for breaking the ties in (5.1) and (5.2) (which you write down explicitly!) solve the following linear programming problem for which a starting basis is evident after bringing the problem into the standard form:*

$$\begin{array}{ll} \max & 2x_1 + 3x_2 + 4x_3 + 2x_4 \\ \text{subject to} & \begin{array}{lllll} x_1 + x_2 + x_3 + x_4 & \leq 10 \\ 3x_1 + x_2 + 4x_3 + 2x_4 & \leq 12 \\ x_i & \geq 0 & \text{for } i = 1, \dots, 4. \end{array} \end{array}$$


---

The data of the problem in its standard (minimization) form are as follows

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 1 & 4 & 2 & 0 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 10 \\ 12 \end{pmatrix}, \quad \mathbf{c} = (-2, -3, -4, -2, 0, 0)$$

We choose  $j$  to be the first  $k$  such that  $\bar{c}_k < 0$  for (5.1) and  $\ell$  to be the first index such that  $\theta = \bar{b}_{p_\ell}/y_j^{p_\ell}$  for (5.2). Then we have

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad I = \{5, 6\} \text{ and } \mathbf{c}_B = (0, 0).$$

$$\mathbf{B}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} = \begin{pmatrix} 10 \\ 12 \end{pmatrix} \text{ and } \bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} = (-2, -3, -4, -2, 0, 0)$$

$$j = 1, \quad \mathbf{y}_1 = \mathbf{B}^{-1} \mathbf{a}_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \theta = \min\left\{\frac{10}{1}, \frac{12}{3}\right\} = 4, \quad \ell = 6.$$

$$I = \{5, 1\}, \quad \mathbf{B} = \mathbf{B} + (\mathbf{a}_1 - \mathbf{a}_6)(0, 1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix}(0, 1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 3 \end{pmatrix}.$$

$$\mathbf{c}_B = \mathbf{c}_B + (c_1 - c_6)(0, 1) = (0, 0) + (-2)(0, 1) = (0, -2), \quad \mathbf{B}^{-1} = \frac{1}{3} \begin{pmatrix} 3 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{pmatrix} \text{ and}$$

$$\bar{\mathbf{b}} = \mathbf{B}^{-1} \mathbf{b} = \begin{pmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 10 \\ 12 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}.$$

$$\bar{\mathbf{c}} = (-2, -3, -4, -2, 0, 0) - (0, -2) \begin{pmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 1 & 4 & 2 & 0 & 1 \end{pmatrix} = (0, -\frac{5}{3}, -\frac{4}{3}, -\frac{2}{3}, 0, \frac{2}{3})$$

$$j = 2, \mathbf{y}_2 = \begin{pmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}, \theta = \min\left\{\frac{6}{2}, \frac{4}{\frac{1}{3}}\right\} = \min\{9, 12\} = 9, \ell = 5.$$

$$I = \{2, 1\}, \mathbf{B} = \mathbf{B} + (\mathbf{a}_2 - \mathbf{a}_5)(1, 0) = \begin{pmatrix} 1 & 1 \\ 0 & 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} (1, 0) = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}.$$

$$\mathbf{c}_B = \mathbf{c}_B + (c_2 - c_5)(1, 0) = (0, -2) + (-3)(1, 0) = (-3, -2), \mathbf{B}^{-1} = \frac{1}{2} \begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

$$\bar{\mathbf{b}} = \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 10 \\ 12 \end{pmatrix} = \begin{pmatrix} 9 \\ 1 \end{pmatrix}.$$

$$\bar{\mathbf{c}} = (-2, -3, -4, -2, 0, 0) - (-3, -2) \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 1 & 4 & 2 & 0 & 1 \end{pmatrix} = (0, 0, -\frac{5}{2}, \frac{1}{2}, \frac{7}{2}, -\frac{1}{2}).$$

$$j = 3, \mathbf{y}_3 = \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{3}{2} \end{pmatrix}, \theta = \min\left\{\frac{1}{\frac{3}{2}}\right\} = \frac{2}{3}, \ell = 1.$$

$$I = \{2, 3\}, \mathbf{B} = \mathbf{B} + (\mathbf{a}_3 - \mathbf{a}_1)(0, 1) = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} (0, 1) = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}.$$

$$\mathbf{c}_B = \mathbf{c}_B + (c_3 - c_1)(0, 1) = (-3, -2) + (-2)(0, 1) = (-3, -4), \mathbf{B}^{-1} = \frac{1}{3} \begin{pmatrix} 4 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{4}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix},$$

$$\bar{\mathbf{b}} = \begin{pmatrix} \frac{4}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 10 \\ 12 \end{pmatrix} = \begin{pmatrix} 28/3 \\ 2/3 \end{pmatrix}.$$

$$\bar{\mathbf{c}} = (-2, -3, -4, -2, 0, 0) - (-3, -4) \begin{pmatrix} \frac{4}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 1 & 4 & 2 & 0 & 1 \end{pmatrix} = (\frac{5}{3}, 0, 0, \frac{4}{3}, \frac{8}{3}, \frac{1}{3}).$$

Since  $\bar{\mathbf{c}} \geq \mathbf{0}$ , the optimal solution has been found and it is  $x_1 = 0, x_2 = \frac{28}{3}, x_3 = \frac{2}{3}$  and  $x_4 = 0$ . The optimal objective function value is  $3(\frac{28}{3}) + 4(\frac{2}{3}) = \frac{92}{3} = 30\frac{2}{3}$ .

## Exercise 5.2

Write a computer program of the simplex algorithm in a computer language of your choice, using any “canned” inversion routine or the updating formulas based on (4.5) (see the text), for the linear programming problem in canonical form with  $\mathbf{b} \geq \mathbf{0}$ .

The following program is an implementation of the simplex algorithm as a MATLAB function.

```
% Primal simplex algorithm for LPs in canonical form
% max {cx: A~x <= b, x >= 0}
% where b >= 0. The identity matrix corresponding to the
% slack variables is assumed to be a~feasible initial basis.
%
% INPUT VARIABLES
```

```
% A,b,c          -> LP data
%
% RETURN VARIABLES
% sol_stat =  0 -> unbounded solution
%             -1 -> infeasible solution
%             1 -> finite optimal solution
% x            -> primal solution vector
% z            -> optimal objective function value

function [sol_stat,x,z] = psimplex(A,b,c)
[m,n]=size(A);
A=[A'; eye(m)]';
c=[-c zeros(1,m)];
I=eye(m);
B=I;
Binv=I;
cb=zeros(1,m);
for i=1:m, p(i)=n+i; end
bbar=Binv*b';
if bbar < 0
    fprintf('Initial basis is infeasible.');
    sol_stat = -1;
    return
end
iter=0;
sol_stat=-2;
while (sol_stat < 0) ,
    iter=iter+1;
    cbar=c-cb*Binv*A;
    if (cbar>=0)
        fprintf('Optimum found in %d iterations.\n',iter);
        z=-cb*bbar;
        x=zeros(1,n);
        for i=1:m, x(p(i))=bbar(i); end
        sol_stat=1;
        return;
    end
    k=1;
    while cbar(k) >=0, k=k+1; end
    j=k;
    yj=Binv*A(:,j);
    if (yj <= 0)
        fprintf('Problem is unbounded');
        sol_stat= 0;
        return;
    end
```

```

theta=2^30;
for i=1:m,
    if yj(i) > 0
        if bbar(i)/yj(i) < theta
            theta=bbar(i)/yj(i);
            r=i;
        end
    end
end
l=p(r);
ur=I(:,r);
B=B+(A(:,j)-A(:,l))*ur';
cb=cb+(c(j)-c(l))*ur';
p(r)=j;
Binv=inv(B);
bbar=Binv*b';
end

```

The input data for Exercise 5.1 are put as follows into a file called psdat.m:

```

c=[2 3 4 2];
b=[ 10 12];
A=[ 1 1 1 1 ; 3 1 4 2 ];

```

The following shows the function call from MATLAB and the output for the above data (assuming that they are in the file psdat.m):

```

>> psdat
>> [stat,x,z]=psimplex(A,b,c)
Optimum found in 4 iterations.

```

```

stat =
    1
x =
      0    9.3333    0.6667
z =
    30.6667

```

---

### Exercise 5.3

Consider the problem  $\max\{\sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_j x_j \leq a_0, \mathbf{x} \geq \mathbf{0}\}$  where  $c_j > 0$ ,  $a_j > 0$  for  $j = 1, 2, \dots, n$  and  $a_0 > 0$ . What is the optimal solution to this problem?

*Optional addendum:* Drop the sign assumption on the data and discuss all possible cases.

---

We bring the problem into standard form by introducing a slack variable  $x_{n+1} \geq 0$  and setting  $c_{n+1} = 0$ ,  $a_{n+1} = 1$ . Since  $0 \leq x_j \leq a_0/a_j$  for  $1 \leq j \leq n+1$  the solution set is nonempty and bounded.

Thus by the fundamental theorem of linear programming an optimal basis exists. Since  $m = 1$  a basis is an  $1 \times 1$  matrix, i.e. a scalar. Let  $a_k$  be the optimal basis. Then the reduced cost for the  $j$ -th variable is  $\bar{c}_j = c_j - c_k a_j / a_k$  and since  $a_k$  is an optimal basis and we have a maximization problem,  $\bar{c}_j \leq 0$  for all  $1 \leq j \leq n+1$ . It follows that  $\frac{c_k}{a_k} \geq \frac{c_j}{a_j}$  for all  $1 \leq j \leq n+1$  if  $a_k$  is an optimal basis. Thus if we select  $k \in \{1, \dots, n+1\}$  such that  $\frac{c_k}{a_k} = \max\{\frac{c_j}{a_j} : 1 \leq j \leq n+1\}$ , then the optimal solution is  $x_k = \frac{a_0}{a_k}$  and  $x_j = 0$  for all  $1 \leq j \leq n+1, j \neq k$  and the optimal value is  $c_k a_0 / a_k$ .

Suppose now that no sign restrictions are imposed on the data and let  $N = \{1, \dots, n\}$ . We distinguish the following cases:

(1) Assume  $a_0 \geq 0$ . Then if  $a_j \leq 0$  for all  $j \in N$  and there exists  $j \in N$  with  $c_j > 0$  the solution is unbounded. If  $c_j \leq 0$  for all  $j \in N$  then  $\mathbf{x} = \mathbf{0}$  is the optimal solution. Now let  $N_+ = \{j \in N : a_j > 0\}$  and  $N_- = \{j \in N : a_j < 0\}$  and assume that both sets are nonempty. The optimal solution if exists is  $x_k = \frac{a_0}{a_k}$  for some  $k \in N_+$ ,  $x_j = 0$  for all  $j \neq k$  or  $\mathbf{x} = \mathbf{0}$ . In the first case the variables  $j \in N_+, j \neq k$  price out correctly if  $\frac{c_j}{a_j} \leq \frac{c_k}{a_k}$ , the variables  $j \in N_-$  price out correctly if  $\frac{c_j}{a_j} \geq \frac{c_k}{a_k}$  and the slack variable prices out correctly if  $-\frac{c_k}{a_k} \leq 0$ , i.e. if  $c_k \geq 0$  since by assumption  $a_k > 0$ . In the second case the variables  $j \in N$  price out correctly if  $c_j \leq 0$ . We can eliminate from the LP variables  $j \in N_+$  with  $c_j \leq 0$ , so we assume that  $c_j > 0$  for all  $j \in N_+$ . With this information we distinguish the following cases:

- (i) if  $c_j \leq 0$  for all  $j \in N$ ,  $\mathbf{x} = \mathbf{0}$  is an optimal solution.
- (ii) if  $N_+ = \emptyset$  and there exists  $j \in N$  with  $c_j > 0$  the problem is unbounded.
- (iii) if  $N_+ \neq \emptyset$  let  $k$  be such that  $\frac{c_j}{a_j} \leq \frac{c_k}{a_k}$  for all  $j \in N_+$ 
  - (a) if there exists  $j \in N_-$  such that  $c_j > 0$  then  $x_j$  does not price out correctly and since  $\frac{a_j}{a_k} < 0$  the problem is unbounded.
  - (b) if  $c_j \leq 0$  for all  $j \in N_-$  let  $\ell$  be such that  $\frac{c_\ell}{a_\ell} \leq \frac{c_j}{a_j}$  for all  $j \in N_-$ . If  $\frac{c_\ell}{a_\ell} \geq \frac{c_k}{a_k}$ , then  $x_k = \frac{a_0}{a_k}$ ,  $x_j = 0$  for  $j \neq k$  is an optimal solution. Otherwise the problem is unbounded.
- (iv) if  $N_- = \emptyset$  the solution is found as in the first part of the exercise, unless  $c_j \leq 0$  for all  $j \in N$  in which case  $\mathbf{x} = \mathbf{0}$  is an optimal solution.

(2) Assume  $a_0 < 0$ . Then if  $a_j \geq 0$  for all  $j \in N$  the problem is infeasible since  $\frac{a_0}{a_j} < 0$  for all  $j$  with  $a_j > 0$ . Suppose now that  $N_- = \{j \in N : a_j < 0\} \neq \emptyset$ . Then if there exists an optimal solution it is  $x_k = \frac{a_0}{a_k}$  for some  $k \in N_-$ ,  $x_j = 0$  for  $j \neq k$ . The variables  $j \in N_-, j \neq k$  price out correctly if  $\frac{c_j}{a_j} \geq \frac{c_k}{a_k}$ , the variables  $j \in N_+$  price out correctly if  $\frac{c_j}{a_j} \leq \frac{c_k}{a_k}$ , and the slack prices out correctly if  $-\frac{c_k}{a_k} \leq 0$ , i.e. when  $c_k \leq 0$  since by assumption  $a_k < 0$ . With this information we distinguish the following cases:

- (i) if  $N_- = \emptyset$  the problem is infeasible.
- (ii) if  $N_- \neq \emptyset$  and there exists  $k \in N_-$  with  $c_k > 0$  the problem is unbounded.
- (iii) if  $N_- \neq \emptyset$  and  $c_j \leq 0$  for all  $j \in N_-$ ,  $c_j \leq 0$  for  $j \in N_+$ , then the optimal solution is  $x_k = \frac{a_0}{a_k}$ ,  $x_j = 0$  for  $j \neq k$ , where  $k$  is such that  $\frac{c_k}{a_k} \leq \frac{c_j}{a_j}$  for all  $j \in N_-$ .
- (iv) if  $N_- \neq \emptyset$ ,  $c_j \leq 0$  for all  $j \in N_-$  and  $C_+ = \{j \in N_+ : c_j > 0\} \neq \emptyset$ , let  $\ell \in C_+$  be such that  $\frac{c_j}{a_j} \leq \frac{c_\ell}{a_\ell}$  for all  $j \in C_+$ . If  $\frac{c_\ell}{a_\ell} \leq \frac{c_k}{a_k}$  where  $k$  is defined as in (iii), then  $x_k = \frac{a_0}{a_k}$ ,  $x_j = 0$  for  $j \neq k$  is an optimal solution. Otherwise the problem is unbounded, since  $\frac{a_0}{a_k} < 0$ .

**Exercise 5.4**

Consider the linear programming problem

$$\max \left\{ \sum_{i=1}^n s^i x_i : \sum_{i=1}^n r^{-i} x_i \leq t, \mathbf{x} \geq \mathbf{0} \right\}$$

where  $i$  means “to the power  $i$ ” and  $s$  and  $r$  are parameters satisfying  $1 > r > 0, s > 1, rs < 1$  and  $(1 - r)r < \frac{s-1}{s^2}$  (e.g.  $r = \frac{2}{5}, s = 2$ ). Show that if one starts with the solution  $\mathbf{x} = \mathbf{0}$  the simplex algorithm with the choice rule (c1) requires exactly  $n$  steps until the optimum is found, while the choice rule (c2) converges in one step.

Introducing a slack variable we bring the constraint into equality form, i.e.  $\sum_{i=1}^n \frac{1}{r^i} x_i + x_{n+1} = t$ . Starting with the solution  $\mathbf{x} = \mathbf{0}$ , i.e. with initial basis 1 (the coefficient of  $x_{n+1}$ ) we calculate the reduced cost for variable  $x_j$  to be  $\bar{c}_j = c_j - c_{n+1}a_j = s^i$  and since  $s > 0$  we have that  $\bar{c}_j > 0$  for all  $1 \leq j \leq n$ . Using choice rule (c1) for the entering variable, since we are maximizing, we have to pick up the variable with the maximum reduced cost among those with positive reduced cost. Since  $s > 1$  the entering variable is variable  $x_n$ . Now we claim that if variable  $x_k$  is in the basis in some iteration, then in the next iteration variable  $x_{k-1}$  will enter the basis. To prove the claim we first prove that variables  $x_1, \dots, x_{k-1}$  are the only candidates to enter the basis. Indeed, calculating the reduced cost for variable  $x_j$  we have  $\bar{c}_j = s^j - s^k r^k / r^j$  and thus since  $sr < 1$  we get  $\bar{c}_j > 0$  for  $1 \leq j < k$ ,  $\bar{c}_j < 0$  for  $k < j \leq n$ ,  $\bar{c}_{n+1} = -s^k r^k < 0$  and, of course,  $\bar{c}_k = 0$ . So to prove our claim we have to show that  $\bar{c}_{k-1} > \bar{c}_i$  for all  $1 \leq i \leq k-2$ . To this end we prove that  $\bar{c}_i < \bar{c}_{i+1}$  for all  $1 \leq i \leq k-2$ , i.e. that  $s^i - \frac{s^k r^k}{r^i} < s^{i+1} - \frac{s^k r^k}{r^{i+1}}$  which is equivalent to  $\frac{s^k r^k}{r^i} (\frac{1}{r} - 1) < s^i (s - 1)$ . After rearranging terms the inequality to show becomes  $(1 - r)r(rs)^{k-i-2} < \frac{s-1}{s^2}$ . Now, since  $rs < 1$  we have  $(1 - r)r(rs)^{k-i-2} < (1 - r)r < \frac{s-1}{s^2}$ , where the last inequality is given to be satisfied by  $r$  and  $s$  and the claim is proven. It follows that after  $n$  iterations, variable  $x_1$  will be in the basis and then  $\bar{c}_j = s^j - \frac{sr}{r^j} < 0$  for all  $1 < j \leq n$ , since  $sr < 1$  and  $\bar{c}_{n+1} = -sr < 0$ , i.e. the basis is optimal.

Using rule (c2) for the entering variable and starting again with the solution  $\mathbf{x} = \mathbf{0}$  we have already shown that  $\bar{c}_j > 0$  for  $1 \leq j \leq n$  and thus the smallest indexed variable with positive reduced cost is variable  $x_1$ . Bringing  $x_1$  into the basis, we have the optimal basis as shown above. Thus after one iteration the optimal solution is found.

**Exercise 5.5**

Denote by  $B_p$  the basis of the simplex algorithm at iteration  $p$  and assume  $B_0 = I_m$ . Show that  $\det B_p = \prod_{t=1}^p y_j^{r,t}$  where  $y_j^{r,t}$  is the pivot element  $y_j^r$  in iteration  $t$ .

We have from (4.4) that at any given iteration  $t$  the “next” basis is given by

$$B_t = B_{t-1}(I_m + (y_j - u_r)u_r^T)$$

where  $\mathbf{u}_r^T \mathbf{y}_j = y_j^r$  is the pivot element for iteration  $t$ . Taking the determinant of  $B_t$  we have

$$\det B_t = \det B_{t-1} \det(\mathbf{I}_m + (\mathbf{y}_j - \mathbf{u}_r) \mathbf{u}_r^T).$$

We calculate the second determinant as

$$\det(\mathbf{I}_m + (\mathbf{y}_j - \mathbf{u}_r) \mathbf{u}_r^T) = \det \begin{pmatrix} 1 & \cdots & y_j^1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & y_j^r & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & y_j^m & \cdots & 1 \end{pmatrix} = y_j^r$$

and thus  $\det B_t = \det B_{t-1} y_j^{r,t}$ . By backwards substitution since  $B_0 = \mathbf{I}_m$  we calculate for  $p \geq 1$

$$\det B_p = \prod_{t=1}^p y_j^{r,t} \det B_0 = \prod_{t=1}^p y_j^{r,t}.$$

### Exercise 5.6

Suppose that the original equation system has slack variables in positions  $n-m+1, \dots, n$ . Prove that at every iteration the current basis inverse is given by the transformed coefficients in the same columns of the equation format.

The transformed coefficient  $y_j^i$  for some  $n-m+1 \leq j \leq n$  and  $1 \leq i \leq m$  is the  $i$ -th component of the vector  $\mathbf{y}_j = B^{-1} \mathbf{a}_j$  where  $B^{-1}$  is the inverse of the current basis. Since by assumption the slack variables are in positions  $n-m+1, \dots, n$  we have that  $\mathbf{a}_j = \mathbf{u}_{j-n}$  for  $n-m+1 \leq j \leq n$ , i.e. the columns of the matrix  $A$  that correspond to the slack variables are unit vectors forming an identity matrix. But then  $\mathbf{y}_j = B^{-1} \mathbf{a}_j = B_j^{-1}$ , i.e. the vector  $\mathbf{y}_j$  is the  $j$ -th column of the inverse of the current basis.

### \*Exercise 5.7

(i) Show that the objective function value of  $(LP_C^-)$  is unbounded if and only if the same is true for the objective function value of the linear program  $(LP_C)$  in canonical form, where

$$(LP_C) \quad \max\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

$$(LP_C^-) \quad \max \left\{ \begin{pmatrix} \mathbf{c} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ s \end{pmatrix} : \mathbf{A}\mathbf{x} + \mathbf{s} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0} \right\}.$$

(ii) Let  $N = \{1, \dots, n\}$ ,  $M = \{1, \dots, m\}$  and  $\mathbf{d} = \begin{pmatrix} \mathbf{c} & \mathbf{0} \end{pmatrix}$ . For any basis  $B$  with index set  $I$  let

$$P = I \cap N, \quad L = \{i \in M : n+i \notin I\}, \quad S = M - L.$$

Rewriting  $(LP_C)$  in partitioned form with respect to  $P$  and  $L$  we get the equivalent form of  $(LP_C)$

$$\begin{array}{ll} \max & \mathbf{c}_P \mathbf{x}_P + \mathbf{c}_{N-P} \mathbf{x}_{N-P} \\ \text{subject to} & \mathbf{A}_P^L \mathbf{x}_P + \mathbf{A}_{N-P}^L \mathbf{x}_{N-P} \leq \mathbf{b}_L \\ & \mathbf{A}_P^{M-L} \mathbf{x}_P + \mathbf{A}_{N-P}^{M-L} \mathbf{x}_{N-P} \leq \mathbf{b}_{M-L} \\ & \mathbf{x}_P \geq \mathbf{0}, \quad \mathbf{x}_{N-P} \geq \mathbf{0}. \end{array}$$

Thus every basis  $B$  of  $(LP_C^\equiv)$  can be brought into the form

$$B = \begin{pmatrix} \mathbf{A}_P^L & \mathbf{0} \\ \mathbf{A}_P^{M-L} & \mathbf{I}_{m-p} \end{pmatrix},$$

where  $p = |P|$ ,  $|L| = |P|$  and  $\det \mathbf{A}_P^L = \det B$  if  $p \geq 1$ . Assume that  $B$  is an optimal basis for  $(LP_C^\equiv)$  that is found by the simplex algorithm and prove:

(i) If  $p = 0$ , then  $\mathbf{x} = \mathbf{0}$  is an optimal solution to  $(LP_C)$  and  $\mathbf{b} \geq \mathbf{0}$ ,  $\mathbf{c} \leq \mathbf{0}$ .

(ii) If  $1 \leq p \leq m$ , then an optimal solution to  $(LP_C)$  is given by

$$\mathbf{x}_P = (\mathbf{A}_P^L)^{-1} \mathbf{b}_L, \quad \mathbf{x}_{N-P} = \mathbf{0} \tag{5.6}$$

and moreover, we have the inequalities

$$\mathbf{c}_P (\mathbf{A}_P^L)^{-1} \geq \mathbf{0} \quad \text{and} \quad \mathbf{c}_P (\mathbf{A}_P^L)^{-1} \mathbf{A}_{N-P}^L \geq \mathbf{c}_{N-P}. \tag{5.7}$$

(i) Let  $\mathcal{X}^\equiv = \{(\mathbf{x}, \mathbf{s}) \in \mathbb{R}^{n+m} : \mathbf{A}\mathbf{x} + \mathbf{s} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}\}$ . First, we observe that  $(\mathbf{x}, \mathbf{s}) \in \mathcal{X}^\equiv$  if and only if  $\mathbf{x} \in \mathcal{X}^\leq$ . Assume that  $(LP_C^\equiv)$  has an unbounded solution and suppose that  $(LP_C)$  has a bounded optimal solution. Let  $\mathbf{x}^* \in \mathcal{X}^\leq$  be the optimizer. Then  $\mathbf{c}\mathbf{x}^* \geq \mathbf{c}\mathbf{x}$  for every  $\mathbf{x} \in \mathcal{X}^\leq$  and thus for every  $(\mathbf{x}, \mathbf{s}) \in \mathcal{X}^\equiv$  which contradicts the unboundedness of  $(LP_C^\equiv)$ . On the other hand if  $(LP_C)$  is unbounded, suppose that  $(LP_C^\equiv)$  is not. Then there exists an optimal solution  $(\mathbf{x}^*, \mathbf{s}^*) \in \mathcal{X}^\equiv$  such that  $\mathbf{c}\mathbf{x}^* \geq \mathbf{c}\mathbf{x}$  for all  $(\mathbf{x}, \mathbf{s}) \in \mathcal{X}^\equiv$  and thus for all  $\mathbf{x} \in \mathcal{X}^\leq$  which contradicts the unboundedness of  $(LP_C)$ .

(ii) Since  $p = 0$  we have  $P = \emptyset$  and thus the solution to  $(LP_C^\equiv)$  defined by  $B$  is  $\mathbf{x} = \mathbf{0}$ ,  $\mathbf{s} = \mathbf{b}$ . Consequently  $\mathbf{b} \geq \mathbf{0}$ . From the optimality criterion of the simplex algorithm we have  $\mathbf{d}_R - \mathbf{d}_B B^{-1} \mathbf{R} = \mathbf{c} \leq \mathbf{0}$  and thus (i) follows. If  $1 \leq p \leq m$  then we calculate

$$B^{-1} = \begin{pmatrix} (\mathbf{A}_P^L)^{-1} & \mathbf{0} \\ -\mathbf{A}_P^{M-L} (\mathbf{A}_P^L)^{-1} & \mathbf{I}_{m-p} \end{pmatrix},$$

and thus the solution to  $(LP_C^\equiv)$  defined by  $B$  is given by

$$\mathbf{x}_P = (\mathbf{A}_P^L)^{-1} \mathbf{b}_L, \quad \mathbf{x}_{N-P} = \mathbf{0}, \quad \mathbf{s}_P = \mathbf{0}, \quad \mathbf{s}_{M-P} = \mathbf{b}_{M-L} - \mathbf{A}_P^{M-L} \mathbf{x}_P.$$

Hence the solution (5.6) is feasible for  $(LP_C)$ . To derive (5.7) we note that

$$\mathbf{d}_B = \begin{pmatrix} \mathbf{c}_P & \mathbf{0} \end{pmatrix}, \quad \mathbf{d}_R = \begin{pmatrix} \mathbf{c}_{N-P} & \mathbf{0} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \mathbf{A}_{N-P}^L & \mathbf{I}_p \\ \mathbf{A}_{N-P}^{M-L} & \mathbf{0} \end{pmatrix},$$

where the first zero vector has  $m - p$  components, the second one  $p$  components and the zero matrix is of size  $(m - p) \times p$  since  $|L| = |P|$ . Consequently, the optimality criterion of the simplex algorithm becomes

$$\begin{aligned} \mathbf{d}_R - \mathbf{d}_B \mathbf{B}^{-1} \mathbf{R} &= (\mathbf{c}_{N-P} \mathbf{0}) - (\mathbf{c}_P \mathbf{0}) \begin{pmatrix} (\mathbf{A}_P^L)^{-1} & \mathbf{0} \\ -\mathbf{A}_P^{M-L} (\mathbf{A}_P^L)^{-1} \mathbf{I}_{m-p} & \end{pmatrix} \begin{pmatrix} \mathbf{A}_{N-P}^L \mathbf{I}_p \\ \mathbf{A}_{N-P}^{M-L} \mathbf{0} \end{pmatrix} \\ &= \left( \mathbf{c}_{N-P} - \mathbf{c}_P (\mathbf{A}_P^L)^{-1} \mathbf{A}_{N-P}^L, -\mathbf{c}_P (\mathbf{A}_P^L)^{-1} \right) \leq \mathbf{0} \end{aligned}$$

and (5.7) follows. Now suppose that the solution given by (5.6) is not optimal to  $(LP_C)$ . Then there exists  $\bar{\mathbf{x}} \in \mathcal{X}^\leq$  such that  $\mathbf{c}\bar{\mathbf{x}} > \mathbf{c}\mathbf{x}$ . But then  $\bar{\mathbf{s}} = (\bar{\mathbf{x}}, \bar{\mathbf{s}})$  where  $\bar{\mathbf{s}} = \mathbf{b} - \mathbf{A}\bar{\mathbf{x}} \geq \mathbf{0}$  is a feasible solution to  $(LP_{\bar{C}}^-)$  with  $d\bar{\mathbf{s}} > d\mathbf{z} = \mathbf{c}\mathbf{x}$ , which contradicts the optimality of the solution defined by the basis  $B$ .

---

### Exercise 5.8

Show that some optimal solution to the linear program

$$(LP_C^L) \quad \max\{\mathbf{c}\mathbf{x} : \mathbf{A}^L \mathbf{x} \leq \mathbf{b}_L, \mathbf{x} \geq \mathbf{0}\}$$

is optimal for  $(LP_C)$  where  $L = \{i \in M : n+i \notin I\}$ ,  $M = \{1, \dots, m\}$  and  $I$  is the index set of an optimal basis to  $(LP_{\bar{C}}^-)$ ; see Exercise 5.7.

---

Let  $B$  be an optimal basis for the problem  $(LP_{\bar{C}}^-)$  and  $\mathbf{x}_P = (\mathbf{A}_P^L)^{-1} \mathbf{b}_L$ ,  $\mathbf{x}_{N-P} = \mathbf{0}$  be the corresponding optimal solution. From (5.7) we have that  $\mathbf{c}_{N-P} - \mathbf{c}_P (\mathbf{A}_P^L)^{-1} \mathbf{A}_{N-P}^L \leq \mathbf{0}$ . We claim that this is an optimal solution to the problem  $(LP_C^L)$ . For suppose not. Then the basis  $\mathbf{A}_P^L$  is not optimal, i.e.  $\mathbf{c}_{N-P} - \mathbf{c}_P (\mathbf{A}_P^L)^{-1} \mathbf{A}_{N-P}^L \not\leq \mathbf{0}$  which contradicts the optimality of the solution for  $(LP_C)$ .

### Exercise 5.9

For any integers  $a$ ,  $b$  and  $c$  satisfying  $b \geq a \geq 2$  and  $c > ab$  consider the linear program in canonical form

$$\begin{aligned} \max \quad & \sum_{i=1}^n b^{n-i} x_i \\ \text{subject to } & \sum_{k=1}^{i-1} a b^{i-k} x_k + x_i \leq c^{i-1} \quad \text{for } i = 1, \dots, n \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Denote by  $s_i$  the slack variable of constraint  $i$  and prove that either  $x_i$  or  $s_i$  is in the basis in every basic feasible solution where  $1 \leq i \leq n$ . Let  $i_0 = 0 < i_1 < \dots < i_s < i_{s+1} = n+1$  be any set of indices and  $S = \{i_1, \dots, i_s\}$ , where  $0 \leq s \leq n$ .

- (i) Prove that  $x_{i_k} = c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1}$  for  $k = 1, \dots, s$ ,  $x_j = 0$  for  $j \notin S$  is a nondegenerate basic feasible solution where  $\mathbf{x} = \mathbf{0}$  if  $s = 0$ . Moreover, the linear program has exactly  $2^n$  basic feasible solutions.

- (ii) Prove that the objective function in reduced form is given by

$$\begin{aligned} \sum_{j=1}^n b^{n-j} x_j &= \sum_{k=1}^s (1-a)^{s-k} b^{n-i_k} c^{i_k-1} - \sum_{k=1}^s (1-a)^{s-k} b^{n-i_k} s_{i_k} \\ &\quad + \sum_{k=1}^s (1-a)^{s+1-k} \sum_{j=i_{k-1}+1}^{i_k-1} b^{n-j} x_j + \sum_{j=i_s+1}^n b^{n-j} x_j, \end{aligned}$$

i.e. the first term on the right-hand side of the equation is the objective function value of the basic feasible solution defined in (i) and the rest gives the reduced cost of the nonbasic variables.

- (iii) Prove that with pivot rule (c1) for column and (r1) for row selection the simplex algorithm iterates as follows: (a) If  $s = 1$  or  $s = 0$ , then if  $i_s = n$  stop; otherwise variable  $x_{i_s+1}$  is pivoted into the basis. (b) If  $s \geq 2$  and  $s$  even, then if  $i_1 > 1$  variable  $x_1$  is pivoted into, whereas if  $i_1 = 1$  variable  $x_1$  is pivoted out of the basis. (c) If  $s \geq 3$  and  $s$  odd, then if  $i_1 + 1 < i_2$  variable  $x_{i_1+1}$  is pivoted into, whereas if  $i_1 + 1 = i_2$  variable  $x_{i_2}$  is pivoted out of the basis.
- (iv) Let  $\mathbf{z}$  be a vector of length  $n$  and initially  $\mathbf{z} = \mathbf{0}$ . At any point of the following procedure denote by  $s$  the number of nonzero components and by  $i_k$  the position of the  $k^{\text{th}}$  nonzero component of  $\mathbf{z}$  where  $1 \leq k \leq s$ ,  $p$  and  $q$  are counters which are initially zero. The iterative step goes as follows:

Increment  $p$  by one. If  $p \geq 2^n$ , stop. If  $0 \leq s \leq 1$ , increment  $q$  by one, set  $z_q = 1$  and go to the iterative step. If  $s$  is even, then if  $z_1 = 1$  set  $z_1 = 0$ , else set  $z_1 = 1$  and go to the iterative step. If  $s$  is odd, then if  $i_1 + 1 < i_2$  set  $z_{i_1+1} = 1$ , else set  $z_{i_2} = 0$  and go to the iterative step.

Prove by induction on  $n$  that the procedure produces all  $2^n$  distinct zero-one vectors of length  $n$  and that the last vector produced is given by  $z_j = 0$  for  $j = 1, \dots, n-1$ ,  $z_n = 1$ . (Hint: Denote by  $z^{p+1}$  the zero-one vector produced in step  $p$  starting with  $z^1 = 0$ . In the step from  $n$  to  $n+1$  prove that  $z_j^{2^n+k} = z_j^{2^n+1-k}$  for  $j = 1, \dots, n$ ,  $z_{n+1}^{2^n+k} = 1$  for  $1 \leq k \leq 2^n$  using induction on  $k$ .)

- (v) Use (iv) to conclude that the simplex algorithm with the choice rules (c1) and (r1) requires  $2^n - 1$  iterations to solve the above linear program if it is started at  $\mathbf{x} = 0$ .
  - (vi) Consider a “reverse” rule (c2) and (r2) where “first” and “smallest” is replaced by “last” and “largest” in (c2) and likewise in (r2). Show that the simplex algorithm with the reverse choice rules (c2) and (r2) stops after one step if started at  $\mathbf{x} = 0$ . How about choice rules (c4) and (c1) as changed in (c5)?
- 

We claim that  $x_i + s_i > 0$  for all  $1 \leq i \leq n$ . This is clear for  $i = 1$  since  $x_1 + s_1 = c^0 = 1 > 0$ . So suppose the claim is true for  $1 \leq i \leq \ell < n$  and that  $x_{\ell+1} + s_{\ell+1} = 0$ . Then  $\sum_{k=1}^{\ell} a b^{\ell+1-k} x_k = c^{\ell}$  and thus

$$\sum_{k=1}^{\ell} b^{\ell-k} x_k = \frac{c^{\ell}}{ab} > c^{\ell-1} \geq \sum_{k=1}^{\ell-1} ab^{\ell-k} x_k + x_{\ell}.$$

Hence  $0 > \sum_{k=1}^{\ell-1} (a-1)b^{\ell-k} x_k \geq 0$ , which is a contradiction to  $\mathbf{x} \geq \mathbf{0}$ . So in every basic feasible solution either  $x_i$  or  $s_i$  or both variables belong to the basis. But every basic feasible solution has exactly  $n$  basic variables. Thus either  $x_i$  and  $s_i$  is basic and moreover, every basic feasible solution is nondegenerate.

(i) To prove this part we claim first that

$$\begin{aligned} x_{i_k} &= c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1} \\ &\quad + a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} s_{i_\ell} - s_{i_k} - a \sum_{\ell=1}^k (1-a)^{k-\ell} \sum_{h=i_{\ell-1}+1}^{i_\ell-1} b^{i_k-h} x_h \end{aligned} \tag{5.8}$$

for all  $1 \leq k \leq s$  and for any set  $S = \{i_1, \dots, i_s\} \subseteq \{1, \dots, n\}$  with  $i_1 < \dots < i_s$ . If  $s = 1$ , then from constraint  $i_1$  the claim follows. So suppose (5.8) is true for some  $s \geq 1$ . We show that formula (5.8) is true for any  $S \subseteq \{1, \dots, n\}$  with  $|S| = s+1 \leq n$ . From the induction hypothesis it follows

that (5.8) is correct for  $1 \leq k \leq s$ . Thus from constraint  $i_{s+1}$  we get:

$$\begin{aligned}
x_{i_{s+1}} &= c^{i_{s+1}-1} - s_{i_{s+1}} - \sum_{k=1}^{i_{s+1}-1} ab^{i_{s+1}-k} x_k \\
&= c^{i_{s+1}-1} - s_{i_{s+1}} - \sum_{k=1}^s ab^{i_{s+1}-i_k} x_{i_k} - \sum_{k=1}^{s+1} \sum_{h=i_{k-1}+1}^{i_k-1} ab^{i_{s+1}-h} x_h \\
&= c^{i_{s+1}-1} - \sum_{k=1}^s ab^{i_{s+1}-i_k} (c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1}) \\
&\quad - \sum_{k=1}^s a^2 \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_{s+1}-i_\ell} s_{i_\ell} + \sum_{k=1}^s ab^{i_{s+1}-i_k} s_{i_k} \\
&\quad - s_{i_{s+1}} + a^2 \sum_{k=1}^s \sum_{\ell=1}^k (1-a)^{k-\ell} \sum_{h=i_{\ell-1}+1}^{i_\ell-1} b^{i_{s+1}-i_h} x_h \\
&\quad - \sum_{k=1}^s \sum_{h=i_{k-1}+1}^{i_k-1} ab^{i_{s+1}-h} x_h - \sum_{h=i_s+1}^{i_{s+1}-1} ab^{i_{s+1}-h} x_h .
\end{aligned}$$

We calculate next

$$\begin{aligned}
\sum_{k=1}^s b^{i_{s+1}-i_k} (c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1}) &= \sum_{k=1}^s (b^{i_{s+1}-i_k} c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_{s+1}-i_\ell} c^{i_\ell-1}) \\
&= \sum_{k=1}^s b^{i_{s+1}-i_k} c^{i_k-1} - \sum_{k=1}^s a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_{s+1}-i_\ell} c^{i_\ell-1} \\
&= \sum_{k=1}^s b^{i_{s+1}-i_k} c^{i_k-1} + \sum_{k=1}^{s-1} ((1-a)^{s-k} - 1) b^{i_{s+1}-i_k} c^{i_k-1} = \sum_{k=1}^s (1-a)^{s-k} b^{i_{s+1}-i_k} c^{i_k-1} ,
\end{aligned}$$

where we have used that

$$\sum_{k=1}^s \sum_{\ell=1}^{k-1} \alpha^{k-1-\ell} \beta_\ell = \sum_{k=1}^{s-1} \frac{\alpha^{s-k} - 1}{\alpha - 1} \beta_k \tag{5.9}$$

for any real  $\beta_1, \dots, \beta_s$  and  $\alpha \neq 1$ . Likewise, we calculate

$$\begin{aligned}
\sum_{k=1}^s (-a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_{s+1}-i_\ell} s_{i_\ell} + b^{i_{s+1}-i_k} s_{i_k}) &= \sum_{k=1}^{s-1} ((1-a)^{s-k} - 1) b^{i_{s+1}-i_k} s_{i_k} + \sum_{k=1}^s b^{i_{s+1}-i_k} s_{i_k} \\
&= \sum_{k=1}^s (1-a)^{s-k} b^{i_{s+1}-i_k} s_{i_k} .
\end{aligned}$$

We calculate also

$$\sum_{k=1}^s \left( a \sum_{\ell=1}^k (1-a)^{k-\ell} \sum_{h=i_{\ell-1}+1}^{i_\ell-1} b^{i_{s+1}-i_h} x_h - \sum_{h=i_{k-1}+1}^{i_k-1} b^{i_{s+1}-h} x_h \right) = - \sum_{k=1}^s (1-a)^{s+1-k} \sum_{h=i_{k-1}+1}^{i_k-1} b^{i_{s+1}-h} x_h ,$$

where we have used that

$$\sum_{k=1}^s \sum_{\ell=1}^k \alpha^{k-\ell} \beta_\ell = \sum_{k=1}^s \frac{\alpha^{s+1-k} - 1}{\alpha - 1} \beta_k \quad (5.10)$$

for any real  $\beta_1, \dots, \beta_s$  and  $\alpha \neq 1$ . Combining the three terms, formula (5.8) follows for  $k = s + 1$ .

To show the positivity of  $x_{i_k}$  as defined in (5.8) we proceed as follows:

$$\begin{aligned} x_{i_k} &= c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1} \\ &= c^{i_k-1} - a b^{i_k-i_{k-1}} c^{i_{k-1}-1} + a(a-1) \sum_{\ell=1}^{k-2} (1-a)^{k-2-\ell} b^{i_k-i_\ell} c^{i_\ell-1} \\ &= c^{i_{k-1}-1} (c^{i_k-i_{k-1}} - a b^{i_k-i_{k-1}}) + a(a-1) \sum_{\ell=1}^{k-2} (1-a)^{k-2-\ell} b^{i_k-i_\ell} c^{i_\ell-1}. \end{aligned} \quad (5.11)$$

Now remember that  $i_1 < i_2 < \dots < i_k$ . For  $k < 3$  the last summation is empty and thus equal to zero. Since  $c > ab$  and  $a \geq 2$  imply  $c^\ell - ab^\ell > 0$  for all  $\ell \geq 1$  it follows that  $x_{i_1} > 0$  and  $x_{i_2} > 0$ . To prove that the last summation is nonnegative, we factor out  $b^{i_k-i_{k-2}} > 0$ , simplify the notation and claim that

$$\sum_{\ell=1}^k (1-a)^{k-\ell} b^{i_k-i_\ell} c^{i_\ell-1} > 0 \quad \text{for all } k \geq 1. \quad (5.12)$$

The claim is true for  $k = 1$  and  $k = 2$ . Suppose it is true for some  $k \geq 2$ . Then

$$\begin{aligned} \sum_{\ell=1}^{k+1} (1-a)^{k+1-\ell} b^{i_{k+1}-i_\ell} c^{i_\ell-1} &= (1-a)^2 b^{i_{k+1}-i_{k-1}} \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_{k-1}-i_\ell} c^{i_\ell-1} \\ &\quad + b^{i_{k+1}-i_k} c^{i_k-1} + c^{i_k-1} (c^{i_{k+1}-i_k} - ab^{i_{k+1}-i_k}) > 0 \end{aligned}$$

by the induction hypothesis and the previous reasoning. Consequently,  $x_{i_k} > 0$  for all  $1 \leq k \leq s$ .

To prove feasibility of  $x_{i_k}$  for  $1 \leq k \leq s$  we check the constraints. They are satisfied when  $i \leq i_1$ , with equality if  $i = i_1$ . Suppose  $i > i_1$  and let  $i_1 < i_2 < \dots < i_{p-1} < i \leq i_p$  for some  $p \geq 2$ . Define

$$\xi_i = c^{i-1} - a \sum_{\ell=1}^{p-1} (1-a)^{p-1-\ell} b^{i-i_\ell} c^{i_\ell-1}.$$

Thus  $\xi_i = x_{i_p}$  if  $i = i_p$ , whereas  $\xi_i > 0$  if  $i < i_p$ . (The last assertion follows from the previous argument using the partition  $\{i_1, \dots, i_{p-1}, i_p = i\}$  for the definition of the  $x_{i_k}$ .) Then we calculate

using (5.11)

$$\begin{aligned}
\sum_{k=1}^{i-1} ab^{i-k} x_k + \xi_i &= \sum_{k=1}^{p-1} ab^{i-i_k} x_{i_k} + \xi_i \\
&= \sum_{k=1}^{p-1} ab^{i-i_k} (c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1}) + \xi_i \\
&= a \left( \sum_{k=1}^{p-1} b^{i-i_k} c^{i_k-1} - a \sum_{k=1}^{p-1} \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i-i_\ell} c^{i_\ell-1} \right) + \xi_i \\
&= a \left( \sum_{k=1}^{p-1} b^{i-i_k} c^{i_k-1} + \sum_{k=1}^{p-2} ((1-a)^{p-1-k} - 1) b^{i-i_k} c^{i_k-1} \right) + \xi_i \\
&= a \sum_{k=1}^{p-1} (1-a)^{p-1-k} b^{i-i_k} c^{i_k-1} + \xi_i = c^{i-1},
\end{aligned}$$

where we have used (5.9) with  $s = p-1$ . It follows that the solution is feasible, that all inequalities with  $i \in \{i_1, \dots, i_s\}$  hold as equations whereas all others are strict inequalities. The submatrix of the constraint matrix corresponding to the columns and rows with indices  $i_1, \dots, i_s$  is lower triangular and has ones on the main diagonal. All other rows are strict inequalities. We must thus choose their corresponding slacks as the corresponding basic variables. Thus the solution  $(x, s)$  defined by the  $x_{i_k}$  for  $1 \leq i \leq s$  is a nondegenerate basic feasible solution. Since we can choose any subset  $S \subseteq \{1, \dots, n\}$  there are at least  $2^n$  basic feasible solutions. Since either  $x_i$  or  $s_i$  for  $1 \leq i \leq n$  is basic in any basic feasible solution there are exactly  $2^n$  such solutions.

**(ii)** Using (5.8) we calculate for the objective function

$$\begin{aligned}
\sum_{j=1}^n b^{n-j} x_j &= \sum_{k=1}^s b^{n-i_k} x_{i_k} + \sum_{k=1}^{s+1} \sum_{h=i_{k-1}+1}^{i_k-1} b^{n-h} x_h \\
&= \sum_{k=1}^s b^{n-i_k} (c^{i_k-1} - a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} c^{i_\ell-1}) + \sum_{k=1}^s b^{n-i_k} (a \sum_{\ell=1}^{k-1} (1-a)^{k-1-\ell} b^{i_k-i_\ell} s_{i_\ell} - s_{i_k}) \\
&\quad - a \sum_{k=1}^s \sum_{\ell=1}^k (1-a)^{k-\ell} \sum_{h=i_{\ell-1}+1}^{i_\ell-1} b^{n-h} x_h + \sum_{k=1}^{s+1} \sum_{h=i_{k-1}+1}^{i_k-1} b^{n-h} x_h \\
&= \sum_{k=1}^s (1-a)^{s-k} b^{n-i_k} c^{i_k-1} - \sum_{k=1}^s (1-a)^{s-k} b^{n-i_k} s_{i_k} \\
&\quad + \sum_{k=1}^s (1-a)^{s+1-k} \sum_{j=i_{k-1}+1}^{i_k-1} b^{n-j} x_j + \sum_{j=i_s+1}^n b^{n-j} x_j,
\end{aligned} \tag{5.13}$$

where we have used (5.9) and (5.10) to simplify the summations. So the reduced form of the objective function follows and we thus know all reduced costs for any basic solution to the problem.

**(iii)** To apply pivot rules we need to fix a sequential ordering of the rows and columns of the problem in standard form. The ordering of the rows is the natural one stated in the formulation. The variables  $x_1, \dots, x_n$  are as in the order of the formulation and  $x_{n+1}, \dots, x_{2n}$  with  $x_{n+i} = s_i$  for  $1 \leq i \leq n$  is the indexing of the slack variables as implied by the row order. From part (ii) we know the reduced cost  $\bar{c}_j$  for  $1 \leq j \leq 2n$ . Moreover, we know that in each basis that we encounter either  $x_i$  or  $s_i$  is basic. So if we pivot  $x_{n+i}$ , i.e. the slack variable  $s_i$ , into the basis then we pivot  $x_i$  out of the basis and vice versa. So we need only pivot rule (c1) which we repeat when stated for a maximization problem with  $2n$  variables.

(c1) Choose  $j$  such that  $\bar{c}_j = \max\{\bar{c}_k : 1 \leq k \leq 2n\} > 0$  and break any remaining ties by choosing the maximizer with smallest index.

To prove (a) assume first that  $s = 0$ . Then all slacks are basic. Thus from (c1) we pivot  $x_1$  into the basis because  $b^{n-1} > b^{n-j}$  for all  $j \geq 2$ . If  $s = 1$ , then from (5.13) we have

$$\sum_{j=1}^n b^{n-j} x_j = b^{n-i_1} c^{i_1-1} - b^{n-i_1} s_{i_1} + (1-a) \sum_{j=1}^{i_1-1} b^{n-j} x_j + \sum_{j=i_1+1}^n b^{n-j} x_j.$$

Since  $(1-a)b^{n-j} < 0$  for  $j \geq 1$  and  $-b^{n-i_1} < 0$ , it follows that  $x_{i_1+1}$  is pivoted into the basis according to (c1) if  $i_1 < n$ , because  $b^{n-i_1+1} > b^{n-j}$  for all  $j > i_1 + 1$ . Otherwise if  $i_1 = n$  the current solution is optimal and (a) follows.

To prove (b) assume  $s \geq 2$ ,  $s$  even and  $i_1 > 1$ . It follows that  $x_1$  is nonbasic with reduced cost  $\bar{c}_1 = (1-a)^s b^{n-1} > 0$  which is the unique maximum reduced cost because  $(1-a)^s \geq 1$ . Consequently,  $x_1$  is pivoted into the basis. Suppose now  $i_1 = 1$ . Then  $x_1$  is in the basis and  $s_1$  is nonbasic with reduced cost  $\bar{c}_{n+1} = -(1-a)^{s-1} b^{n-1} > 0$ . Again it follows that  $\bar{c}_{n+1}$  is the unique maximum reduced cost because  $-(1-a)^{s-1} \geq 1$ . Hence  $s_1 = x_{n+1}$  is pivoted into the basis which means that  $x_1$  leaves the basis and thus (b) is proven.

To prove (c) assume  $s \geq 3$ ,  $s$  odd and  $i_1 + 1 < i_2$ . It follows that  $x_{i_1+1}$  is a nonbasic variable with reduced cost  $\bar{c}_{i_1+1} = (1-a)^{s-1} b^{n-i_1-1} > 0$ . Since  $i_1 + 1 < i_2$  and  $(1-a)^{s-1} \geq -(1-a)^{s-2} \geq 1$  it follows that  $\bar{c}_{i_1+1}$  is the unique maximum reduced cost. Consequently, variable  $x_{i_1+1}$  is pivoted into the basis. Suppose now  $i_1 + 1 = i_2$ . Then  $x_{i_2}$  is in the basis and  $x_{n+i_2} = s_{i_2}$  is out of the basis with reduced cost  $\bar{c}_{n+i_2} = -(1-a)^{s-2} b^{n-i_2} > 0$ . Since  $i_1 + 1 = i_2$  it follows as before that  $\bar{c}_{n+i_2}$  is the unique maximum reduced cost and thus  $x_{i_2}$  is pivoted out of the basis which proves (c) and thus part (iii) follows.

**(iv)** Starting with  $z^1 = 0$  we have that the iterative scheme works for  $n = 1$  as follows:

$$s = 0 \rightarrow z_1 = 1 \rightarrow z^2 = (1)$$

and thus the iterative scheme produces the two possible 0-1 vectors of length one. For  $n = 2$  we have

$$s = 0; z_1 = 1 \rightarrow z^2 = (1, 0) \rightarrow s = 1; z_2 = 1 \rightarrow z^3 = (1, 1) \rightarrow s = 2; z_1 = 0 \rightarrow z^4 = (0, 1)$$

and thus again we get all 0-1 vectors of length two. Moreover, in both cases the last vector that we get has its only nonzero element in the  $n$ -th position. Suppose that the algorithm produces all 0-1 vectors of dimension  $\ell$  for  $\ell \leq n$  and that the last vector produced has only one nonzero component at the  $\ell$ -th position. To show that this is true for vectors of dimension  $n+1$  we show by induction on  $k$  that  $z_j^{2^n+k} = z_j^{2^n+1-k}$  for  $j = 1, \dots, n$ ,  $z_{n+1}^{2^n+k} = 1$  for  $1 \leq k \leq 2^n$ .

For  $k = 1$  we have that  $s = 1$  since  $z^{2^n} = (0, \dots, 0, 1)$  and thus  $z_{n+1}^{2^{n+1}} = 1$  and the assertion holds. Suppose that it holds for any  $t \leq k$ , i.e. we have  $z_j^{2^{n+t}} = z_j^{2^{n+1-t}}$  for  $j = 1, \dots, n$  and  $z_{n+1}^{2^{n+t}} = 1$  for all  $t \leq k$ . Suppose first that the number  $s$  of nonzeros in  $z^{2^n+k}$  is even and assume  $z_1^{2^n+k} = 1$ . Then  $s$  for  $z^{2^{n+1-k}}$  is odd and  $z_1^{2^{n+1-k}} = 1$ . Now according to the iterative scheme  $z^{2^n+k+1}$  will have  $s$  odd and  $z_1^{2^{n+k+1}} = 0$ , all other element remaining the same. The number  $s$  for  $z^{2^n-k}$  must be even since it differs only in one element from  $z^{2^{n+k+1}}$  and thus  $z_1^{2^n-k} = 0$ . Thus  $z_j^{2^{n+k+1}} = z_j^{2^{n-k}}$  for  $j = 1, \dots, n$  and  $z_{n+1}^{2^{n+k+1}} = 1$  since  $k \leq 2^n$ . Now suppose that  $s$  for  $z^{2^n+k}$  is odd and  $i_1 + 1 < i_2$ . Then  $z^{2^n+k+1}$  differs from  $z^{2^n+k}$  only at the  $i_1 + 1$ -st element, i.e.  $s$  for  $z^{2^n+k+1}$  is even and  $z_{i_1+1}^{2^{n+k+1}} = 1$ . Also,  $z^{2^n-k+1}$  has even  $s$  and  $z_{i_1+1}^{2^{n-k+1}} = 0$ . Then  $z^{2^n-k}$  has odd  $s$  and  $z_{i_1+1}^{2^n-k} = 1$ , because if  $z_{i_1+1}^{2^n-k} = 1$  then  $z_{i_1+1}^{2^{n-k+1}} = 0$  in the next iteration which is a contradiction. Thus the inductive proof is complete.

**(v)** Let  $z = (z_1, \dots, z_n)$  be a zero one vector with  $z_i = 1$  if variable  $x_i$  is in the basis,  $z_i = 0$  otherwise. As we start pivoting at the basic solution  $x = 0$  we have initially  $z = 0$ . From part (iii) it follows that the iterative application of pivoting rules (c1) and (r1) produce the same sequence of zero-one vectors as the iterative scheme of part (iv). Hence the simplex algorithm with choice rules (c1) and (r1) iterates exactly  $2^n - 1$  times before it comes to a stop.

**(vi)** The “reverse” rule (c2) for maximization, which is all that we need, goes as follows

*(c2) Choose the last  $j$  for which  $\bar{c}_j > 0$ , i.e. the largest column index that has a positive reduced cost.*

Since initially all of  $x_1, \dots, x_n$  are nonbasic and have reduced cost  $\bar{c}_j = b^{n-j}$  for  $1 \leq j \leq n$  it follows that we pivot variable  $x_n$  into the basic set. But then by the proof of part (a) of (iii) we have the optimal solution. We leave it to you to figure out what happens when normalized (or steepest edge) pivot criteria are used.

---

### Exercise 5.10

Apply the simplex algorithm with block pivots to the linear program of Exercise 5.9 where a block pivot consists of pivoting into the basis the largest possible subset of columns that do not price out correctly at the current iteration. Show that after at most  $n$  iterations the algorithm comes to a halt if it is started at  $x = 0$ .

(Hint: Show that all columns that do not price out correctly are exchanged at every iteration, that after the first iteration variable  $x_n$  remains in the basis and that – in the notation of Exercise 5.9 –  $i_{s-1} = n - 2$  in the second iteration. Moreover, show that in the remaining iterations the variables  $x_{i_{s-1}+1}, \dots, x_{n-1}$  remain nonbasic and that  $i_{s-1}$  decreases monotonically.)

---

We use the same notation as in the solution to Exercise 5.9, see in particular the solution to part (iii) for the indexing conventions, and assume  $n \geq 3$ , the other cases being trivial. From Exercise 5.9 we know that every basic index set to the problem in standard form is of the form  $I = S \cup \{n+i : i \in N - S\}$  where  $S \subseteq N = \{1, \dots, n\}$  is arbitrary. Consequently, we can block-pivot all nonbasic columns that do not price out correctly into the basis. Thus in block pivot 1 we pivot all variables  $x_1, \dots, x_n$  into the basis and all slacks leave the basis if we start at  $x = 0$ . Thus  $i_k = k$

for  $1 \leq k \leq n$  after the first block pivot and from Exercise 5.9(ii) we get the objective function in reduced form to be

$$\sum_{j=1}^n b^{n-j} x_j = \sum_{k=1}^n (1-a)^{n-k} b^{n-k} c^{k-1} - \sum_{k=1}^n (1-a)^{n-k} b^{n-k} s_k.$$

Now the reduced cost  $\bar{c}_{n+k} = -(1-a)^{n-k} b^{n-k}$  for  $1 \leq k \leq n$  and thus variable  $s_n = x_{2n}$  prices out correctly since  $\bar{c}_{2n} = -1$ , while  $s_{n-1} = x_{2n-1}$  has a reduced cost of  $\bar{c}_{2n-1} = -(1-a)b > 0$  and does not price out correctly. Moreover, variable  $s_{n-2} = x_{2n-2}$  has a reduced cost of  $\bar{c}_{2n-2} = -(1-a)^2 b^2 < 0$ , which means that it is not pivoted into the basis by the next block pivot. Consequently, the second block pivot produces a basis with  $i_s = n$ ,  $i_{s-1} = n-2$  and variable  $x_{n-1}$  is nonbasic. We claim that  $i_{s-1} = n-p$  and  $i_s = n$  after  $p$  block pivots. As we have just shown this is true for  $p = 1$  and  $p = 2$ . So suppose this to be true for some  $p \geq 2$ . Then the objective function in reduced form is given by

$$\begin{aligned} \sum_{j=1}^n b^{n-j} x_j &= \text{const} - \sum_{k=1}^{s-2} (1-a)^{s-k} b^{n-i_k} s_{i_k} + (a-1)b^p s_{n-p} - s_n \\ &\quad + \sum_{k=1}^{s-2} (1-a)^{s+1-k} \sum_{j=i_{k-1}+1}^{i_k-1} b^{n-j} x_j + (1-a)^2 \sum_{j=i_{s-2}+1}^{n-p-1} b^{n-j} x_j + (1-a) \sum_{j=n-p+1}^{n-1} b^{n-j} x_j. \end{aligned}$$

Consequently, in the next block pivot  $x_n$  remains in the basis, the variables  $x_{n-p+1}, \dots, x_{n-1}$  remain nonbasic,  $x_{n-p-1}$  enters the basis and  $x_{n-p}$  leaves the basis. Consequently, after  $p+1$  pivots we have  $i_{s-1} = n-p-1$  and  $i_s = n$ . The claim follows and thus at most  $n$  block pivots are used in an iterative application.

---

### \*Exercise 5.11

Consider the normed pivot column selection criterion (c5) using the column norms  $n_j$  given by  $n_j^2 = 1 + \|\mathbf{y}_j\|^2$  for  $j \in J$ , where  $\mathbf{y}_j = \mathbf{B}^{-1} \mathbf{a}_j$  and  $J$  is the index set of the nonbasic columns. Let  $\mathbf{B}_{new}$  be a new basis obtained from  $\mathbf{B}$  by pivoting in column  $j$  and row  $r$ , see (4.4). Show that

$$(n_k^{new})^2 = 1 + \Theta_k^2 + \|\mathbf{y}_k - \Theta_k \mathbf{y}_j\|^2 \quad k \in J - j, \quad (n_\ell^{new})^2 = (\mathbf{y}_j^r)^{-2} n_j^2,$$

where  $\ell$  is the index of the variable that leaves the basis  $\mathbf{B}$ ,  $\mathbf{y}_k^r = \mathbf{u}_r^T \mathbf{B}^{-1} \mathbf{a}_k$  for all  $k \in J - j$  and  $\Theta_k = \mathbf{y}_k^r / \mathbf{y}_j^r$ . Discuss various ways of using these relations to "update" the new norms from the "old" norms. (See Chapter 7 for a geometric interpretation of the normed pivot column selection criterion.)

---

From formula (4.5) of Chapter 4 we have

$$\mathbf{B}_{new}^{-1} = \mathbf{B}^{-1} - \frac{1}{\mathbf{y}_j^r} (\mathbf{y}_j - \mathbf{u}_r) \mathbf{u}_r^T \mathbf{B}^{-1}$$

and denoting  $(B^{-1})^T$  simply by  $B^{-T}$  we calculate for  $k \in J - j$

$$\begin{aligned}(n_k^{new})^2 &= 1 + (\mathbf{y}_k^{new})^T \mathbf{y}_k^{new} \\&= 1 + \mathbf{a}_k^T (B^{-T} - \frac{1}{y_j^r} B^{-T} \mathbf{u}_r (\mathbf{y}_j - \mathbf{u}_r)^T) (B^{-1} - \frac{1}{y_j^r} (\mathbf{y}_j - \mathbf{u}_r) \mathbf{u}_r^T B^{-1}) \mathbf{a}_k \\&= 1 + (\mathbf{y}_k^T - \Theta_k (\mathbf{y}_j - \mathbf{u}_r)^T) (\mathbf{y}_k - \Theta_k (\mathbf{y}_j - \mathbf{u}_r)) \\&= 1 + \|\mathbf{y}_k - \Theta_k \mathbf{y}_j\|^2 + 2y_k^r \Theta_k - 2y_j^r \Theta_k^2 + \Theta_k^2\end{aligned}$$

as claimed. For the leaving variable  $\ell$  we get likewise

$$(n_\ell^{new})^2 = 1 + (\mathbf{y}_\ell^{new})^T \mathbf{y}_\ell^{new} = (y_j^r)^{-2} (1 + \|\mathbf{y}_j\|^2)$$

and thus the assertion follows. To get a recursion formula for  $n_k^{new}$  when  $k \neq \ell$  we calculate the norm on the right-hand side of the formula and get

$$(n_k^{new})^2 = n_k^2 + \Theta_k^2 n_j^2 - 2\Theta_k \mathbf{y}_k^T \mathbf{y}_j \text{ for } k \in J - j.$$

We can thus update the norms from the old ones at the expense of calculating the inner products  $\mathbf{y}_k^T \mathbf{y}_j$  for all  $k \in J - j$ . To see how this calculation can be sped up write  $\mathbf{y}_k^T \mathbf{y}_j = \mathbf{a}_k^T B^{-T} \mathbf{y}_j = \mathbf{a}_k^T \mathbf{w}$  where  $\mathbf{w} = B^{-T} \mathbf{y}_j$ , i.e.,  $\mathbf{w}$  is a solution to the system of equations  $\mathbf{y}_j^T = \mathbf{w}^T B$ . Besides the inner products  $\mathbf{y}_k^T \mathbf{y}_j$  we need the coefficients  $y_k^r = \mathbf{u}_r^T B^{-T} \mathbf{a}_k$  which necessitates taking the inner product of the pivot row of the basis with the nonbasic column  $\mathbf{a}_k$ . All other quantities are needed for the simplex algorithm as well. Thus with the additional work just described the norms can be updated while carrying out the pricing step of the simplex algorithm. From a numerical point of view it is desirable to ensure that the norms remain positive during the iterative calculations. From the first formula for  $n_k^{new}$  it follows that  $(n_k^{new})^2 \geq 1 + \Theta_k^2 \geq 1$  and thus one can use the following updating formulas for the norms  $(n_\ell^{new})^2 = (y_j^r)^{-2} n_j^2$ ,  $(n_k^{new})^2 = \max\{1 + \Theta_k^2, n_k^2 + \Theta_k^2 n_j - 2\Theta_k \mathbf{a}_k^T \mathbf{w}\}$  for  $k \in J - j$ .

---

**\*Exercise 5.12** Consider the linear program in  $ns \geq 1$  variables  $\xi_j$

$$\max\{\hat{\mathbf{c}}\xi : \mathbf{A}_1\xi \leq \mathbf{b}_1, \mathbf{A}_2\xi \geq \mathbf{b}_2, \mathbf{A}_3\xi = \mathbf{b}_3, \ell \leq \xi \leq \mathbf{u}\}$$

where  $-\infty \leq \ell_j \leq u_j \leq \infty$  for  $j \in NS = \{1, \dots, ns\}$  and  $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$  are  $m_1 \times ns, m_2 \times ns$  and  $m_3 \times ns$  matrices of reals, respectively. Let  $m = m_1 + m_2 + m_3$ ,  $n = ns + m$  and denote by  $ns+1, \dots, ns+m$  the indices of the slack, surplus and artificial variables corresponding to the inequalities/equations. We define  $c_{ns+j} = \ell_{ns+j} = 0$ ,  $u_{ns+j} = +\infty$  if  $ns+j$  corresponds to a slack or surplus variable and  $c_{ns+j} = \ell_{ns+j} = u_{ns+j} = 0$  if  $ns+j$  indexes an artificial variable. We thus consider the bounded variable linear program in standard form

$$(BVLP) \quad \max\{\mathbf{c}\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \ell \leq \mathbf{x} \leq \mathbf{u}\},$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the vector of  $n$  real variables,  $x_j = \xi_j$  for  $1 \leq j \leq ns$  are the **structural** variables and  $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_j, \dots, \mathbf{a}_n)$  is a matrix of size  $m \times n$  satisfying  $r(\mathbf{A}) = m$  by construction. We denote by

1.  $F = \{j \in NS : \ell_j = -\infty, u_j = +\infty\}$  the **free** variables,
2.  $D = \{j \in N : -\infty < \ell_j < +\infty, u_j = +\infty\}$  the **lower bounded** variables,
3.  $H = \{j \in NS : \ell_j = -\infty, -\infty < u_j < +\infty\}$  the **upper bounded** variables,
4.  $C = \{j \in N : -\infty < \ell_j \leq u_j < +\infty\}$  the **bounded** variables,

where  $N = \{1, \dots, n\}$ . Note that the index set of the slack and the surplus variables is in this notation precisely  $D - NS$ , that  $C - NS$  is the index set of the artificial variables, and that  $F, D, H, C$  is a partitioning of  $N$  into four sets some of which may be empty.

Let  $B$  be any basis of  $A$  with index set  $I \subseteq N$ ,  $|I| = m$  and let  $J = N - I$ . We partition  $J$  into  $G$ ,  $L$  and  $U$  where

5.  $G = F \cap J$  is the set of nonbasic free variables,
6.  $L = (D \cup C) \cap J$  is the set of nonbasic lower-bounded and bounded variables at their lower bounds and
7.  $U = (H \cup C) \cap J$  is the set of nonbasic upper-bounded and bounded variables at their upper bounds.

Correspondingly, we partition the vector  $\mathbf{x} \in \mathbb{R}^n$  into  $\mathbf{x}_B$ ,  $\mathbf{x}_G$ ,  $\mathbf{x}_L$  and  $\mathbf{x}_U$ . For any basis  $B$  cum partitioning  $(G, L, U)$  we define a basic solution to (BVLP), as follows:

$$\mathbf{x}_B = B^{-1}(\mathbf{b} - R_L \ell_L - R_U u_U), \quad \mathbf{x}_G = \mathbf{0}_G, \quad \mathbf{x}_L = \ell_L, \quad \mathbf{x}_U = u_U,$$

where  $R_L = (\mathbf{a}_j)_{j \in L}$ ,  $R_U = (\mathbf{a}_j)_{j \in U}$ ,  $\ell_L = (\ell_j)_{j \in L}$ ,  $u_U = (u_j)_{j \in U}$  and  $\mathbf{0}_G$  is a vector with  $|G|$  zeros. We define  $R_G$ ,  $\ell_B$ , and  $u_B$ , likewise. The objective function value of a basic solution to (BVLP) is given by

$$\begin{aligned} z_B(G, L, U) &= c_B B^{-1}(\mathbf{b} - R_L \ell_L - R_U u_U) + c_L \ell_L + c_U u_U \\ &= c_B B^{-1}\mathbf{b} + (c_L - c_B B^{-1} R_L) \ell_L + (c_U - c_B B^{-1} R_U) u_U. \end{aligned}$$

A basic solution to (BVLP) is a **basic feasible** solution if  $\ell_B \leq \mathbf{x}_B \leq u_B$ . For short, we denote any basic (feasible) solution to (BVLP) by  $(I, G, L, U)$  and call  $B$  (cum  $(G, L, U)$ ) a (feasible) basis.

(i) Justify the use of the term “basic feasible solution” for the notion defined above.

(ii) Prove or disprove: If  $(I, G, L, U)$  satisfies

$$c_G - c_B B^{-1} R_G = \mathbf{0}_G, \quad c_L - c_B B^{-1} R_L \leq \mathbf{0}_L, \quad c_U - c_B B^{-1} R_U \geq \mathbf{0}_U,$$

then the basic solution defined by  $(I, G, L, U)$  is an optimal solution to (BVLP) if it is feasible.

(iii) Suppose that  $(I, G, L, U)$  defines a nonoptimal basic feasible solution to (BVLP). Discuss the possible ways of changing the basis cum partitioning by pivoting in a single nonbasic column and characterize the detection of an unbounded optimum solution to (BVLP).

(iv) Derive updating formulas for  $\bar{\mathbf{b}} = B^{-1}(\mathbf{b} - R_L \ell_L - R_U u_U)$ ,  $\bar{c} = c - c_B B^{-1} A$  and  $z_B(G, L, U)$ .

- (v) State the simplex algorithm for (BVLP) and prove its correctness.
- (vi) Give a procedure that lets you find an initial feasible basis for (BVLP) or conclude that none exists.
- 

**(i)** To justify the use of the term “basic feasible solution” one brings (BVLP) into the usual standard form. To do so we substitute  $x_j = x_j^+ - x_j^-$  for  $j \in F$  and require that  $x_j^+ \geq 0$  and  $x_j^- \geq 0$  for  $j \in F$ . Moreover, we introduce new variables  $x'_j = x_j - \ell_j \geq 0$  for all  $j \in D \cup C$  and  $x'_j = u_j - x_j \geq 0$  for all  $j \in H$ . Introducing slack variables the inequalities corresponding to the finite upper bounds are then converted into equations. Assuming WROG that the matrix  $A$  of (BVLP) has full rank, it follows by a counting argument that to every basis  $\hat{B}$  of the augmented matrix there corresponds a basis  $B$  of  $A$  with two types of nonbasic variables: those at their lower bounds and those at their upper bounds with all free variables either basic or nonbasic at value zero. In other words, arguments similar to those used in the upper bounds section above establish the claim. We leave the messy details of this exercise for the reader to carry out on a rainy Sunday afternoon.

**(ii)** Every feasible  $x \in \mathbb{R}^n$  satisfies  $x_B = B^{-1}(b - R_G x_G - R_L x_L - R_U x_U)$  and  $x_L \geq \ell_L$ ,  $x_U \leq u_U$ . Using the assumptions we calculate and estimate

$$\begin{aligned} cx &= c_B x_B + c_G x_G + c_L x_L + c_U x_U \\ &= c_B B^{-1}b + (c_G - c_B B^{-1}R_G)x_G + (c_L - c_B B^{-1}R_L)x_L + (c_U - c_B B^{-1}R_U)x_U \\ &\leq c_B B^{-1}b + 0_G x_G + (c_L - c_B B^{-1}R_L)\ell_L + (c_U - c_B B^{-1}R_U)u_U \leq z_B(G, L, U), \end{aligned}$$

for all feasible  $x \in \mathbb{R}^n$ . Consequently, if  $\ell_B \leq x_B \leq u_B$  then the basic feasible solution defined by  $(I, G, L, U)$  is optimal for (BVLP).

**(iii)** Suppose that  $(I, G, L, U)$  defines a basic feasible solution that satisfies the optimality criterion proven in (ii) then we are done and a pivot is not required. So suppose the contrary. There are thus three cases to be distinguished:

- (a)  $\bar{c}_j \neq 0$  for some  $j \in G$ ,   (b)  $\bar{c}_j > 0$  for some  $j \in L$ , or   (c)  $\bar{c}_j < 0$  for some  $j \in U$ ,

where  $\bar{c} = c - c_B B^{-1}A$  are the reduced objective function coefficients. In all cases let  $y_j = B^{-1}a_j$  be the corresponding transformed column for  $j \in J$ .

(a) If  $\bar{c}_j > 0$  for some  $j \in G$  we calculate

(a1)

$$\theta = \min\{u_j - \ell_j, \min\left\{\frac{\bar{b}_i - \ell_{k(i)}}{y_j^i} : y_j^i > 0, k(i) \in I \cap (D \cup C)\right\}, \min\left\{\frac{\bar{b}_i - u_{k(i)}}{y_j^i} : y_j^i < 0, k(i) \in I \cap (H \cup C)\right\}\},$$

where  $k(i)$  is the index of the variable that is basic in column  $i$  of  $B$ , i.e., the variable  $\ell \in I$  with position number  $p_\ell = i$ . Note, that basic free variables are ignored and that the minimum ratio is determined such that  $\theta$  is the largest nonnegative change for  $x_j$  that is possible so as to make sure that  $\ell_B \leq \bar{b} - \theta y_j \leq u_B$ . If  $\bar{c}_j < 0$  for some  $j \in G$  we calculate likewise

(a2)

$$\theta = \max\{\ell_j - u_j, \max\left\{\frac{\bar{b}_i - \ell_{k(i)}}{y_j^i} : y_j^i < 0, k(i) \in I \cap (D \cup C)\right\}, \max\left\{\frac{\bar{b}_i - u_{k(i)}}{y_j^i} : y_j^i > 0, k(i) \in I \cap (H \cup C)\right\}\},$$

i.e.,  $\theta$  is the largest nonpositive change for  $x_j$  so as to make sure that  $\ell_B \leq \bar{b} - \theta y_j \leq u_B$ . Both possibilities (a1) and (a2) must be considered because  $x_j$  is a free variable. Since  $u_j = +\infty$  and  $\ell_j = -\infty$  we have  $\theta = +\infty$  if  $\theta = u_j - \ell_j$  in the first case and likewise,  $\theta = -\infty$  in the second case. Suppose first that  $\theta$  is a finite number. If  $\theta = (\bar{b}_r - \ell_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (D \cup C)$  then in either case (a1) or (a2), variable  $x_j$  enters the basis and leaves the set  $G$ , i.e., the new set  $G'$  is  $G - j$ , whereas the variable  $x_{k(r)}$  leaves the basis and enters into the set  $L$ , i.e., the new set  $L'$  is  $L \cup \{k(r)\}$ . If  $\theta = (\bar{b}_r - u_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (H \cup C)$  then in either case (a1) or (a2), variable  $x_j$  leaves the set  $G$  and enters the basis, whereas the variable  $x_{k(r)}$  leaves the basis and enters the set  $U$ , i.e., we have  $G' = G - j$  and  $U' = U \cup \{k(r)\}$ . Suppose now that  $\theta = \pm\infty$ . Then  $\ell_B \leq \bar{b} - \theta y_j \leq u_B$  for all  $\theta \geq 0$  in case (a1) and all  $\theta \leq 0$  in case (a2). Consequently, defining  $y \in \mathbb{R}^n$  by  $y_{k(i)} = -y_j^i$  for  $i \in I$ ,  $y_j = +1$ ,  $y_k = 0$  otherwise, it follows that  $x^* + \theta y$  is feasible for (BVLP) for all  $\theta$  where  $x^*$  is the current basic feasible solution. But

$$c(x^* + \theta y) = z_B(G, L, U) + \bar{c}_j \theta \rightarrow +\infty$$

for  $\theta \rightarrow +\infty$  in case (a1) and for  $\theta \rightarrow -\infty$  in case (a2), i.e., (BVLP) has an unbounded optimum. Note that the above minimum ratio calculations (a1) and (a2) are restricted to all **nonfree** basic variables and that  $\theta = \pm\infty$  can arise only if both of the two “inner” maxima or minima in the calculation of  $\theta$  are undefined.

(b) If  $\bar{c}_j > 0$  for some  $j \in L$  we calculate  $\theta \geq 0$  by formula (a1) where  $u_j = +\infty$  if  $j \notin C$ . Thus if  $\theta = u_j - \ell_j$  then no variable leaves the basis, variable  $x_j$  leaves  $L$  and enters  $U$ , i.e.,  $L' = L - j$  and  $U' = U \cup \{j\}$ , while  $I$  and  $G$  remain unchanged. If in this case  $j \notin C$ , then by the same argument as in case (a) it follows that (BVLP) has an unbounded optimum. If  $\theta = (\bar{b}_r - \ell_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (D \cup C)$  then  $I' = I - k(r) \cup \{j\}$ ,  $L' = L - j \cup \{k(r)\}$  and  $G$  and  $U$  remain unchanged. If  $\theta = (\bar{b}_r - u_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (H \cup C)$ , then  $I' = I - k(r) \cup \{j\}$ ,  $L' = L - j$ ,  $G' = G$  and  $U' = U \cup \{k(r)\}$  are the resulting new basic set and partition of the nonbasic variables, respectively.

(c) If  $\bar{c}_j < 0$  for some  $j \in U$  we calculate  $\theta \leq 0$  by formula (a2) where  $\ell_j = -\infty$  if  $j \notin C$ . If  $\theta = \ell_j - u_j$  then like in case (b) a variable switches bounds and the basis remains unchanged. If this happens when  $j \notin C$ , then we argue like in case (a) that (BVLP) has an unbounded optimum. If  $\theta = (\bar{b}_r - \ell_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (D \cup C)$ , then  $I' = I - k(r) \cup \{j\}$ ,  $G' = G$ ,  $L' = L \cup \{k(r)\}$ ,  $U' = U - j$ , whereas if  $\theta = (\bar{b}_r - u_{k(r)})/y_j^r$  for some  $k(r) \in I \cap (H \cup C)$  then  $I' = I - k(r) \cup \{j\}$ ,  $G' = G$ ,  $L' = L$ ,  $U' = U - j \cup \{k(r)\}$  are the resulting new basic set and partitioning of the nonbasic variables, respectively.

We summarize the basis change as follows. Whenever a leaving variable  $k(r) \in I$  is undefined, the basis does not change. In case (a) where  $j \in G$  this means that unboundedness of (BVLP) is detected, in cases (b) and (c) this means that the nonbasic variable  $j$  switches its bound unless  $j \notin C$  in which case we again have unboundedness of (BVLP). In the case when the minimum ratio test defines a leaving variable  $k(r) \in I$  the index set of the basic variables changes in all cases to  $I' = I - k(r) \cup \{j\}$ . In Table 5.1 we have summarized the changes in the partitioning of the nonbasic variables. Sets in the partitioning that do not change are not shown and  $\ell = k(r) \in I$  denotes the variable that leaves the basis. “ $\ell \in I \cap (D \cup C) \downarrow$ ” means the leaving variable  $\ell$  decreases to its lower bound, etc.

**(iv)** We assume first that the basis  $B$  is changed. We denote the entering variable by  $j$  and the variable that leaves the basis by  $\ell$ . Variable  $\ell$  corresponds to column  $r$  of  $B$ . All changed quantities get a subscript or superscript *new*. Thus  $B_{new} = B + (a_j - a_\ell)u_r^T$  is the new basis

**Table 5.1.** Changing bases  $(I, G, L, U)$  in (BVLP)

	$\ell \in I \cap (D \cup C) \downarrow$	$\ell \in I \cap (H \cup C) \uparrow$
$j \in G$	$G - j, L \cup \{\ell\}$	$G - j, U \cup \{\ell\}$
$j \in L$	$L - j \cup \{\ell\}$	$L - j, U \cup \{\ell\}$
$j \in U$	$L \cup \{\ell\}, U - j$	$U - j \cup \{\ell\}$

where  $\mathbf{u}_r \in \mathbb{R}^n$  is the  $r$ -th unit vector and likewise  $\mathbf{c}_{B_{new}} = \mathbf{c}_B + (c_j - c_\ell)\mathbf{u}_r^T$ . We use the formula

$$\mathbf{B}_{new}^{-1} = \mathbf{B}^{-1} - \frac{1}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r)\mathbf{u}_r^T \mathbf{B}^{-1}$$

repeatedly where  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$  and  $y_j^r = \mathbf{u}_r^T \mathbf{y}_j$ . We claim that  $\bar{\mathbf{b}}$  is updated as follows

$$\bar{b}_i^{new} = \bar{b}_i - \theta y_j^i \quad \text{for } 1 \leq i \neq r \leq m, \quad \bar{b}_r^{new} = \begin{cases} \theta & \text{if } j \in G, \\ \theta + \ell_j & \text{if } j \in L, \\ \theta + u_j & \text{if } j \in U. \end{cases}$$

To prove it we must calculate  $\bar{\mathbf{b}}^{new}$  according to the six possible cases that we have discussed in part (iii) of this exercise. We denote  $\mathbf{b}^* = \mathbf{b} - \mathbf{R}_L \ell_L - \mathbf{R}_U \mathbf{u}_U$  and thus  $\bar{\mathbf{b}} = \mathbf{B}^{-1} \mathbf{b}^*$ . Suppose  $j \in G$  and  $\ell \in I \cap (D \cup C)$ . Using  $\theta$  defined in (a1) of part (iii) we calculate

$$\begin{aligned} \bar{\mathbf{b}}^{new} &= \mathbf{B}_{new}^{-1}(\mathbf{b} - \mathbf{R}_L^{new} \ell_L^{new} - \mathbf{R}_U \mathbf{u}_U) = \mathbf{B}_{new}^{-1}(\mathbf{b}^* - \ell_\ell \mathbf{a}_\ell) \\ &= \bar{\mathbf{b}} - \frac{\bar{b}_r}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r) - \ell_\ell \mathbf{u}_r + \frac{\ell_\ell}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r) = \bar{\mathbf{b}} - \theta(\mathbf{y}_j - \mathbf{u}_r) - \ell_\ell \mathbf{u}_r, \end{aligned}$$

and thus the formula is correct. If  $j \in G$  and  $\ell \in I \cap (H \cup C)$  we have

$$\begin{aligned} \bar{\mathbf{b}}^{new} &= \mathbf{B}_{new}^{-1}(\mathbf{b} - \mathbf{R}_L \ell_L - \mathbf{R}_U^{new} \mathbf{u}_U^{new}) = \mathbf{B}_{new}^{-1}(\mathbf{b}^* - u_\ell \mathbf{a}_\ell) \\ &= \bar{\mathbf{b}} - \frac{\bar{b}_r}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r) - u_\ell \mathbf{u}_r + \frac{u_\ell}{y_j^r}(\mathbf{y}_j - \mathbf{u}_r) = \bar{\mathbf{b}} - \theta(\mathbf{y}_j - \mathbf{u}_r) - u_\ell \mathbf{u}_r, \end{aligned}$$

and thus the formula is correct. For the remaining four cases we indicate the necessary calculations and leave the details to the reader. If  $j \in L$  and  $\ell \in I \cap (D \cup C)$  we have

$$\bar{\mathbf{b}}^{new} = \mathbf{B}_{new}^{-1}(\mathbf{b} - \mathbf{R}_L^{new} \ell_L^{new} - \mathbf{R}_U \mathbf{u}_U) = \mathbf{B}_{new}^{-1}(\mathbf{b}^* - \ell_\ell \mathbf{a}_\ell + \ell_j \mathbf{a}_j),$$

and if  $j \in L$  and  $\ell \in I \cap (H \cup C)$  we calculate

$$\bar{\mathbf{b}}^{new} = \mathbf{B}_{new}^{-1}(\mathbf{b} - \mathbf{R}_L^{new} \ell_L^{new} - \mathbf{R}_U^{new} \mathbf{u}_U^{new}) = \mathbf{B}_{new}^{-1}(\mathbf{b}^* - u_\ell \mathbf{a}_\ell + \ell_j \mathbf{a}_j).$$

If  $j \in U$  and  $\ell \in I \cap (D \cup C)$  we have

$$\bar{\mathbf{b}}^{new} = \mathbf{B}_{new}^{-1}(\mathbf{b} - \mathbf{R}_L^{new} \ell_L^{new} - \mathbf{R}_U^{new} \mathbf{u}_U^{new}) = \mathbf{B}_{new}^{-1}(\mathbf{b}^* - \ell_\ell \mathbf{a}_\ell + u_j \mathbf{a}_j),$$

and if  $j \in L$  and  $\ell \in I \cap (H \cup C)$  we find

$$\bar{b}^{new} = B_{new}^{-1}(b - R_L \ell_L - R_U^{new} u_U^{new}) = B_{new}^{-1}(b^* - u_\ell a_\ell + u_j a_j).$$

Substituting the expression for  $B_{new}^{-1}$  and carrying out the necessary algebra you find that the updating formula for  $\bar{b}^{new}$  simplifies as claimed above.

It remains to derive the updating formula for  $b$  when in case (b) or (c) of part (iii) a nonbasic variable switches its bound. So suppose  $j \in L$  and  $\theta = u_j - \ell_j$  as in case (b). Then

$$\bar{b}^{new} = B_{new}^{-1}(b - R_L^{new} \ell_L^{new} - R_U^{new} u_U^{new}) = B_{new}^{-1}(b^* + \ell_j a_j - u_j a_j) = \bar{b} - \theta y_j$$

and this formula remains correct if  $j \in U$  and  $\theta = \ell_j - u_j$  as in case (c).

The updating formula for  $v = c_B B^{-1}$  is computed by

$$v_{new} = c_{B_{new}} B_{new}^{-1} = (c_B + (c_j - c_\ell) u_r^T) B_{new}^{-1} = c_B B^{-1} + \lambda u_r^T B^{-1} = v + \lambda u_r^T B^{-1},$$

where  $\lambda = \bar{c}_j/y_j^r$  and  $\bar{c}_j = c_j - c_B B^{-1} a_j$  is the reduced cost of the entering variable. Consequently, denoting by  $y^r = u_r^T B^{-1} A$  the transformed pivot row we get the following updating formulas for the reduced objective function coefficients

$$\bar{c}^{new} = c - c_{B_{new}} B_{new}^{-1} A = c - v_{new} A = \bar{c} - \lambda y^r,$$

which give two different ways of updating  $\bar{c}$ . Finally, the objective function value is updated by

$$z_{B_{new}}(G^{new}, L^{new}, U^{new}) = z_B(G, L, U) + \bar{c}_j \theta,$$

where  $\bar{c}_j$  is the reduced cost of the entering (or switched) variable and  $\theta$  is determined as in part (iii) of this exercise. The details of the derivation of this formula follow the same format that we have employed so far, i.e., you have to distinguish the eight cases involved in changing a basis or switching the bounds of a variable in (BVLP).

(v) The simplex algorithm for (BVLP) can now be stated as follows

### **BVSimplex Algorithm** ( $m, n, \ell, u, A, b, c$ )

**Step 0:** Find a feasible basis  $B$  cum partitioning  $(G, L, U)$  of the nonbasic variables. Let  $I$  be the index set of  $B$  and initialize  $p_k$  for all  $k \in I$ .

**if** none exists **then stop** “BVLP has no feasible solution”.

Compute  $B^{-1}$ ,  $\bar{b} := B^{-1}(b - R_L \ell_L - R_U u_U)$  and initialize  $c_B$ .

**Step 1:** Compute  $\bar{c} := c - c_B B^{-1} A$ .

**if**  $\bar{c}_G = 0_G$ ,  $\bar{c}_L \leq 0_L$  and  $\bar{c}_U \geq 0_U$  **then**

set  $x_B := \bar{b}$ ;  $x_G := 0$ ,  $x_L = \ell_L$ ,  $x_U = u_U$ ,

**stop** “ $x_B, x_G, x_L, x_U$  is an optimal basic feasible solution”.

**else**

(1) choose  $j \in \{k \in G : \bar{c}_k \neq 0\} \cup \{k \in L : \bar{c}_k > 0\} \cup \{k \in U : \bar{c}_k < 0\}$ ,

**endif.**

**Step 2:** Compute  $y_j := B^{-1} a_j$ .

**if**  $\bar{c}_j > 0$  **then** compute the least ratio

$$\theta = \min\{u_j - \ell_j, \min\{\frac{\bar{b}_i - \ell_{k(i)}}{y_j^i} : y_j^i > 0, k(i) \in I \cap (D \cup C)\}\},$$

$$\min\left\{\frac{\bar{b}_i - u_{k(i)}}{y_j^i} : y_j^i < 0, k(i) \in I \cap (H \cup C)\right\}$$

**else** compute the least ratio

$$\theta = \max\{\ell_j - u_j, \max\left\{\frac{\bar{b}_i - \ell_{k(i)}}{y_j^i} : y_j^i < 0, k(i) \in I \cap (D \cup C)\right\}\},$$

$$\max\left\{\frac{\bar{b}_i - u_{k(i)}}{y_j^i} : y_j^i > 0, k(i) \in I \cap (H \cup C)\right\}\}$$

**endif.**

**if**  $\theta = \pm\infty$  **then stop** “BVLP has an unbounded solution”.

(2) Let  $\ell \in I \cup \{j\}$  be any index for which the least ratio  $\theta$  is attained.

**if**  $\ell = j$  **then go to** Step 4 **else** set  $r := p_\ell$  **endif.**

Step 3: Set  $B := B + (a_j - a_\ell)u_r^T$ ,  $c_B := c_B + (c_j - c_\ell)u_r^T$ ,  $I := I - \ell \cup \{j\}$ ,  $p_j := r$ ,  $p_\ell := 0$  and update  $(G, L, U)$  according to Table 5.1.

Recompute  $B^{-1}$  and  $\bar{b}$  and **go to** Step 1.

Step 4: **if**  $j \in L$  **then**

set  $L := L - j$ ,  $U = U \cup \{j\}$

**else**

set  $L := L \cup \{j\}$ ,  $U = U - j$

**endif.**

Set  $\bar{b} := \bar{b} - \theta y_j$  and **go to** Step 1.

The correctness of the BVSimplex Algorithm follows from the discussion preceding it. The sequence of objective function values increases monotonically since  $z_{new} = z_{old} + \bar{c}_j \theta$  and  $\bar{c}_j \theta \geq 0$ . Thus if no basis  $B$  is repeated then the algorithm terminates in a finite number of steps no matter what choice rules are used in the pivot column selection (1) and the selection (2) of the leaving variable. In the case of degenerate pivots, i.e., when  $\theta = 0$ , cycling can occur and thus to assure finite termination, anti-cycling strategies – such as a least-index rule – must be utilized. The algorithm can be sped up (in general) substantially by use of the normed pivot column selection criteria discussed in Exercise 5.11. The efficient organization of the calculations in the BVSimplex Algorithm is left to the reader.

(vi) The easiest way to start the BVSimplex Algorithm is a two-phase procedure using (additional) artificial variables sparingly. Consider the **original** problem in the variables  $\xi$  and introduce slack and/or surplus variables as required. We start by setting all free variables equal to zero, i.e., we put  $G = F$ . If  $|\ell_j| \leq |u_j|$  we put the variable (including slack and surplus variables) at its lower bound, i.e., it enters into the set  $L$ . If  $|\ell_j| > |u_j|$  we put the variable at its upper bound, i.e., it enters the set  $U$ . (Note that this selection criterion for membership in  $L$  and  $U$  is quite arbitrary. We may as well set a variable with a finite upper (lower) bound to its upper bound if  $c_j > 0$  and to its lower bound if  $c_j \leq 0$ , etc.) We thus have now an initial partition  $(G, L, U)$  of all variables that we need and by construction all slack and surplus variables have the value zero. We compute  $b^* = b - \sum_{j \in L} \ell_j a_j - \sum_{j \in U} u_j a_j$  and proceed as follows. If the right-hand side  $b_i^*$  of a less-than-or-equal-to constraint is nonnegative, then the corresponding slack variable leaves the set  $L$  and is put into the initial basis; otherwise,  $b_i^* < 0$  and we append row  $i$  by  $-x'_{n+i}$  where  $x'_{n+i} \geq 0$  is a new artificial variable that we put into the basis. Likewise, if the right-hand side  $b_i^*$  of a greater-than-or-equal-to constraint is nonpositive then the corresponding surplus variable leaves the set  $L$  and is put into the basis; otherwise  $b_i^* > 0$  and we append row  $i$  by  $+x'_{n+i}$  where  $x'_{n+i} \geq 0$  is an artificial variable which is put into the basis. If for an equation  $i$  the right-hand-side  $b_i^* \geq 0$  we append row  $i$  by  $+x'_{n+i}$  and put the artificial variable  $x'_{n+i} \geq 0$  into the basis; otherwise

$b_i^* < 0$  and we append row  $i$  by  $-x'_{n+i}$  and put the artificial variable  $x'_{n+i} \geq 0$  into the basis. Now we have a feasible basis *cum* partition  $(G, L, U)$  for the (possibly) enlarged problem and we can start the BVSimplex Algorithm. In Phase I we maximize the objective function  $-\sum x'_{n+i}$ , where the summation is over all artificial variables that we have introduced. If at termination we have an objective function value of zero for the Phase I problem we have a feasible basis  $B$  *cum* partitioning  $(G, L, U)$  for (BVLP), we change to the original objective function and solve in Phase II the original problem to optimality. If at termination of Phase I the objective function value is negative, then we can stop; the problem (BVLP) does not have any feasible solution. There are other ways to start the algorithm which rely on “guessing” a good basis. It is clear that if you guess “right”, i.e., a basis  $B$  *cum* partitioning  $(G, L, U)$  that satisfies the optimality criterion, then there is no need to iterate and the algorithm terminates right away. However, guessing is a speculative business and may lead to unexpectedly bad results. For further methods to initialize the BVSimplex Algorithm see e.g. R.E. Bixby “Implementing the Simplex Method: The Initial Basis”, *ORSA Journal on Computing*, **4**, 267-284, 1992.

## 6. Primal-Dual Pairs

Every linear programming problem or *primal* linear program has an associated *dual* linear program like follows, where any of the  $m_i \times n_j$  submatrices  $A_{ij}$  for  $1 \leq i, j \leq 3$  may be empty:

PRIMAL	DUAL
$\min c_1x + c_2y + c_3z$	$\max ub_1 + vb_2 + wb_3$
s.t. $A_{11}x + A_{12}y + A_{13}z = b_1$	s.t. $u$ free
$A_{21}x + A_{22}y + A_{23}z \geq b_2$	$v \geq 0$
$A_{31}x + A_{32}y + A_{33}z \leq b_3$	$w \leq 0$
$x \geq 0$	$uA_{11} + vA_{21} + wA_{31} \leq c_1$
$y \leq 0$	$uA_{12} + vA_{22} + wA_{32} \geq c_2$
$z$ free	$uA_{13} + vA_{23} + wA_{33} = c_3$

The primal-dual mechanism for a primal *minimization* problem is summarized as follows:

- The dual is a maximization problem.
- Equations of the primal give rise to “free” variables in the dual.
- Inequalities of the  $\geq$  type correspond to nonnegative dual variables.
- Inequalities of the  $\leq$  type correspond to nonpositive dual variables.
- Nonnegative primal variables give rise to inequalities of the type  $\leq$  in the dual problem, nonpositive primal variables to inequalities of the type  $\geq$  and free primal variables to equations in the dual.

The dual linear program (dLP) to the primal linear program (pLP) in canonical form is

$$(pLP) \quad \max\{cx : Ax \leq b, x \geq 0\} \quad (dLP) \quad \min\{ub : uA \geq c, u \geq 0\}.$$

The dual linear program (dLP) to the primal linear program (pLP) in standard form is

$$(pLP) \quad \min\{cx : Ax = b, x \geq 0\} \quad (dLP) \quad \max\{ub : uA \leq c, u \text{ free}\}.$$

### Weak Duality and Complementary Slackness:

**Remark 6.1** (i) For every primal solution  $(x \ y \ z)$  and every dual solution  $(u \ v \ w)$  we have

$$ub_1 + vb_2 + wb_3 \leq c_1x + c_2y + c_3z. \quad (6.1)$$

(ii) If  $(x \ y \ z)$  is a primal solution and  $(u \ v \ w)$  a dual solution such that we have equality in (6.1), then  $(x \ y \ z)$  and  $(u \ v \ w)$  is an optimal solution to PRIMAL and DUAL, respectively.

**Remark 6.2** If the objective function value of the linear program DUAL is not bounded from above, then the linear program PRIMAL has no feasible solution. On the other hand, if the objective function value of the linear program PRIMAL is not bounded from below; then DUAL has no feasible solution.

Denote by  $\mathbf{a}_{kj}^i$  the  $i^{th}$  row of the matrix  $\mathbf{A}_{kj}$ , where  $1 \leq i \leq m_k$  and  $1 \leq k, j \leq 3$ , by  $a_j$  the  $j^{th}$  column of

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \\ \mathbf{A}_{31} & \mathbf{A}_{32} \end{pmatrix},$$

where  $1 \leq j \leq n_1 + n_2$ , by  $b_j^i$  the components of  $\mathbf{b}_j$  and by  $c_i^j$  the components of  $\mathbf{c}_i$ .

**Remark 6.3** Let  $(\mathbf{x} \ \mathbf{y} \ \mathbf{z})$  and  $(\mathbf{u} \ \mathbf{v} \ \mathbf{w})$  be primal and dual solutions such that inequality (6.1) holds with equality. Then we have the following implications:

- (i)  $v_i > 0$  implies  $\mathbf{a}_{21}^i \mathbf{x} + \mathbf{a}_{22}^i \mathbf{y} + \mathbf{a}_{23}^i \mathbf{z} = b_2^i$ , for  $1 \leq i \leq m_2$ .
- (ii)  $\mathbf{a}_{21}^i \mathbf{x} + \mathbf{a}_{22}^i \mathbf{y} + \mathbf{a}_{23}^i \mathbf{z} > b_2^i$  implies  $v_i = 0$ , for  $1 \leq i \leq m_2$ .
- (iii)  $w_i < 0$  implies  $\mathbf{a}_{31}^i \mathbf{x} + \mathbf{a}_{32}^i \mathbf{y} + \mathbf{a}_{33}^i \mathbf{z} = b_3^i$ , for  $1 \leq i \leq m_3$ .
- (iv)  $\mathbf{a}_{31}^i \mathbf{x} + \mathbf{a}_{32}^i \mathbf{y} + \mathbf{a}_{33}^i \mathbf{z} < b_3^i$  implies  $w_i = 0$ , for  $1 \leq i \leq m_3$ .
- (v)  $x_j > 0$  implies  $c_1^j = (\mathbf{u} \ \mathbf{v} \ \mathbf{w}) \mathbf{a}_j$ , for  $1 \leq j \leq n_1$ .
- (vi)  $c_1^j > (\mathbf{u} \ \mathbf{v} \ \mathbf{w}) \mathbf{a}_j$  implies  $x_j = 0$ , for  $1 \leq j \leq n_1$ .
- (vii)  $y_j < 0$  implies  $c_2^j = (\mathbf{u} \ \mathbf{v} \ \mathbf{w}) \mathbf{a}_j$ , for  $n_1 + 1 \leq j \leq n_2$ .
- (viii)  $c_2^j < (\mathbf{u} \ \mathbf{v} \ \mathbf{w}) \mathbf{a}_j$  implies  $y_j = 0$ , for  $n_1 + 1 \leq j \leq n_2$ .

**Strong Duality:** WROG we will state the strong duality theorem for linear programs in canonical form.

*Notational conventions.* Denote the set of primal (dual) solutions by

$$\mathcal{X}^\leq = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \quad \mathcal{U} = \{\mathbf{u} \in \mathbb{R}^m : \mathbf{uA} \geq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\}.$$

Let  $z_{PRIM} = \max \{ \mathbf{cx} : \mathbf{x} \in \mathcal{X}^\leq \}$ , and  $z_{DUAL} = \min \{ \mathbf{ub} : \mathbf{u} \in \mathcal{U} \}$ . If  $\mathcal{X}^\leq = \emptyset$ , define  $z_{PRIM} = -\infty$  and  $z_{PRIM} = +\infty$  if the primal objective function  $\mathbf{cx}$  is unbounded over  $\mathcal{X}^\leq$ . Likewise,  $z_{DUAL} = +\infty$  if  $\mathcal{U} = \emptyset$  and  $z_{DUAL} = -\infty$  if the dual objective function  $\mathbf{ub}$  is unbounded over the set  $\mathcal{U}$ .

By weak duality and using the above conventions we thus have always  $z_{PRIM} \leq z_{DUAL}$ .

**Theorem 3** The primal linear program (pLP) has a finite optimal solution if and only if the dual linear program (dLP) has a finite optimal solution. Moreover, in either case  $z_{PRIM} = z_{DUAL}$  and the simplex algorithm stops with an optimal solution  $\mathbf{x}$  to (pLP) and an optimal solution  $\mathbf{u}$  to (dLP).

**Remark 6.4** (Complementary slackness) If  $\mathbf{x} \in \mathcal{X}^\leq$  and  $\mathbf{u} \in \mathcal{U}$  are optimal solutions to (pLP) and (dLP), respectively, then

$$\mathbf{u}(\mathbf{Ax} - \mathbf{b}) = \mathbf{0} \quad \text{and} \quad (\mathbf{uA} - \mathbf{c})\mathbf{x} = \mathbf{0}. \quad (6.2)$$

**Remark 6.5** The linear programs (pLP) and (dLP) have both finite optimal solutions if and only if there exist a column vector  $x \in \mathbb{R}^n$  and a row vector  $u \in \mathbb{R}^m$  such that

$$Ax \leq b, uA \geq c, cx - ub \geq 0, x \geq 0, u \geq 0. \quad (6.3)$$

Moreover, every pair of vectors  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$  satisfying (6.3) is a pair of optimal solutions to (pLP) and (dLP), respectively.

**Solvability, Redundancy, Separability:** Call  $m$  linear inequalities in  $n$  unknowns *solvable* if

$$Ax \leq b, x \geq 0 \quad (6.4)$$

has a feasible solution, i.e., if  $\mathcal{X}^{\leq} \neq \emptyset$ , and *nonsolvable* otherwise. An inequality  $dx \leq d_0$  where  $d \in \mathbb{R}^n$  is *redundant relative to* (6.4) if  $\mathcal{X}^{\leq} \cap \{x \in \mathbb{R}^n : dx \leq d_0\} = \mathcal{X}^{\leq}$ . Suppose that there exists  $u \in \mathbb{R}^m$  such that

$$u \geq 0, d \leq uA, ub \leq d_0. \quad (6.5)$$

It follows that *if there exists  $u \in \mathbb{R}^m$  satisfying (6.5), then  $dx \leq d_0$  is redundant relative to (6.4)*. If the reverse statement is true as well, then applying it to  $\mathcal{X}^{\leq}$  and ( $d = 0, d_0 = -1$ ) asserts the existence of  $u \in \mathbb{R}^m$  with

$$u \geq 0, 0 \leq uA \text{ and } ub \leq -1, \quad (6.6)$$

i.e. (6.6) is solvable if  $\mathcal{X}^{\leq} = \emptyset$ . It thus seems *plausible* that the question of the *solvability* of (6.4) can be reduced to the question of the *nonsolvability* of the associated (or *alternative*) inequality system (6.6) and vice versa. This is indeed the case. Denote  $\mathcal{U}^0 = \{u \in \mathbb{R}^m : u \geq 0, uA \geq 0, ub < 0\}$ .

**Lemma 2 (Farkas' Lemma)**  $\mathcal{X}^{\leq} \neq \emptyset$  if and only if  $\mathcal{U}^0 = \emptyset$ .

Let  $d \in \mathbb{R}^n$  be arbitrary,  $\beta$  be any scalar and define

$$\mathcal{X}_{\beta}^> = \{x \in \mathcal{X}^{\leq} : dx > \beta\}, \quad \mathcal{U} = \{u \in \mathbb{R}^m : u \geq 0, uA \geq d\}, \quad \mathcal{U}_{\beta}^{\leq} = \{u \in \mathcal{U} : ub \leq \beta\}.$$

Then  $\mathcal{X}_{\beta}^> \subseteq \mathcal{X}^{\leq}$  and  $\mathcal{U}_{\beta}^{\leq} \subseteq \mathcal{U}$  and the following is an *inhomogeneous* form of Farkas' lemma.

**Theorem 4**  $\mathcal{X}_{\beta}^> \neq \emptyset$  if and only if  $\mathcal{U}^0 = \mathcal{U}_{\beta}^{\leq} = \emptyset$ .

**Corollary 1** An inequality  $dx \leq d_0$  is redundant relative to (6.4) if and only if (6.4) is nonsolvable or there exists a row vector  $u \in \mathbb{R}^m$  satisfying (6.5).

Based on the corollary one can devise criteria for the removal of constraints from linear programming problems without changing the solution set.

The following *strict separation theorem for convex polyhedral sets* is another consequence of linear programming duality and related to the “separation” problem in combinatorial optimization. Let

$$\mathcal{Y}^{\leq} = \{x \in \mathbb{R}^n : Dx \leq g, x \geq 0\},$$

where  $D$  is a  $p \times n$  matrix and  $g \in \mathbb{R}^p$ . The question is: when does there exist a linear inequality  $fx \leq f_0$  that separates  $\mathcal{X}^{\leq}$  and  $\mathcal{Y}^{\leq}$  strictly, i.e. such that  $fx < f_0$  for all  $x \in \mathcal{X}^{\leq}$  and  $fx > f_0$  for all  $x \in \mathcal{Y}^{\leq}$ .

**Theorem 5** Let  $\mathcal{X}^{\leq}$  and  $\mathcal{Y}^{\leq}$  be defined as above and assume  $\mathcal{X}^{\leq} \neq \emptyset \neq \mathcal{Y}^{\leq}$ . Then either  $\mathcal{X}^{\leq} \cap \mathcal{Y}^{\leq} \neq \emptyset$  or there exist a row vector  $f \in \mathbb{R}^n$  and a scalar  $f_0$  such that

$$\mathcal{X}^{\leq} \subseteq \{x \in \mathbb{R}^n : f x < f_0\} \quad \text{and} \quad \mathcal{Y}^{\leq} \subseteq \{x \in \mathbb{R}^n : f x > f_0\}.$$

**Dual Simplex Algorithm:** Consider the primal linear program in standard form and its dual

$$(pLP) \quad \min\{cx : Ax = b, x \geq 0\} \quad (dLP) \quad \max\{ub : uA \leq c, u \text{ free}\}.$$

A basis  $B$  of  $A$  is called a *dual basis* for (pLP) if the reduced cost vector  $\bar{c} = c - c_B B^{-1} A \geq 0$ .

**Remark 6.6** For any dual basis  $B$  for (pLP) the vector  $u = c_B B^{-1}$  is a (basic) feasible solution to (dLP). If a dual basis  $B$  is also a feasible basis for (pLP) then  $x_B = B^{-1}b$ ,  $x_R = 0$  and  $u = c_B B^{-1}$  are optimal solutions to (pLP) and (dLP), respectively. Theorem 3 applies to the primal-dual pair (pLP) and (dLP).

The dual simplex algorithm works directly on the primal linear program (pLP) for input data  $m$ ,  $n$ ,  $A$ ,  $b$  and  $c$ . Remember that  $u_r$  denotes the  $r^{th}$  unit vector.

**Dual Simplex Algorithm ( $m, n, A, b, c$ )**

Step 0: Find a dual basis  $B$ , its index set  $I$  and initialize  $p_k$  for all  $k \in I$ .

**if** none exists **then**

**stop** “(pLP) is either infeasible or unbounded”.

**else**

compute  $B^{-1}$  and  $\bar{c} := c - c_B B^{-1} A$ .

**endif.**

Step 1: Compute  $\bar{b} := B^{-1}b$ .

**if**  $\bar{b} \geq 0$  **then**

set  $x_B := B^{-1}b$ ;  $x_R := 0$ , **stop** “ $x_B$  is an optimal solution to (pLP)”.

**else**

(6.7) choose  $\ell \in I$  such that  $\bar{b}_{p_\ell} < 0$  and set  $r := p_\ell$ .

**endif.**

Step 2: Compute  $y^r := u_r^T B^{-1} R$  and set  $J := N - I$ .

**if**  $y^r \geq 0$  **then**

**stop** “(pLP) has no feasible solution”.

**else**

compute the least ratio  $\gamma := \min \left\{ \frac{\bar{c}_k}{|y_k^r|} : y_k^r < 0, k \in J \right\}$ ,

(6.8) choose  $j \in J$  such that  $\frac{\bar{c}_j}{|y_j^r|} = \gamma$  and  $y_j^r < 0$ .

**endif.**

Step 3: Set  $B := B + (a_j - a_\ell)u_r^T$ ,  $c_B := c_B + (c_j - c_\ell)u_r^T$ ,  $I := I - \{\ell\} \cup \{j\}$  and  $p_j := r$ .

Step 4: Compute  $B^{-1}$ ,  $\bar{c} := c - c_B B^{-1} A$  and **go to** Step 1.

**Reading Instructions:** Like in the (primal) simplex algorithm of Chapter 5, one does not calculate  $B^{-1}$  in actual computation. The dual simplex algorithm requires:

**(D)** Knowledge of  $\bar{c}$ , which is calculated like in Chapter 5.

(E) Knowledge of  $\bar{b}$ , which is calculated like in Chapter 5.

(F) Knowledge of  $y^r$  of  $R$ , which is calculated in two steps: First, one solves  $vB = u_r^T$  and then one computes  $y_k^r = va_k$  for  $k \in J$ .

**Remark 6.7** (Correctness) (i) If no dual basis exists, then (pLP) either has no feasible solution or an unbounded solution. (ii) If  $y^r \geq 0$  in Step 2, then (pLP) has no feasible solution. (iii) The new basis  $B'$ , say, defined in Step 3 is a dual basis for (pLP) with an objective function value  $z_{B'} = z_B - \gamma\bar{b}_r \geq z_B$  where  $\gamma$  is the least ratio of Step 2.

The pivot column and pivot row selection rules of Chapter 5 are adapted to the dual simplex algorithm with the necessary changes. Finiteness of the dual algorithm follows if “least index rules” are used as the choice rules for (6.8) and (6.7):

(c2\*) Choose the smallest  $j \in J$  with  $\frac{c_j}{|y_j^r|} = \gamma$  and  $y_j^r < 0$ . (r2\*) Choose the smallest  $\ell \in I$  s. t.  $\bar{b}_{p_\ell} < 0$ .

**Remark 6.8** (Finiteness) Suppose that a dual basis exists. If the choice rules (c2\*) and (r2\*) are used for pivot column (6.8) and pivot row (6.7) selection, respectively, then the dual simplex algorithm repeats no basis and stops after a finite number of iterations.

If the problem to be solved is originally in standard form with  $c \leq 0$ , then the matrix  $B = I_m$  corresponding to the slack variables is a dual basis and we can get started. Otherwise, we add

$$\sum_{j=1}^n x_j + x_{n+1} = M \quad (6.9)$$

where  $x_{n+1} \geq 0$  is a “new” variable with  $c_{n+1} = 0$  in the objective function vector and  $M = 10^{30}$  a “big” number. If  $x_{n+1} = 0$  in an optimal solution to the enlarged problem, then the dual is unbounded and a dual basis to (pLP) does not exist. The details are in the text.

**Post-Optimality:** Several ways of computing the effect of changing the data of the linear program are explored. In the case of a *parametric* right-hand side, we are interested in the solutions  $x(\theta)$  to

$$(LP_\theta) \quad z(\theta) = \min\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b} + \theta\mathbf{g}, \mathbf{x} \geq 0\} .$$

for all  $\theta$  or for  $\theta$  in some interval, where  $\mathbf{g} \in \mathbb{R}^m$  is a vector of *changes* in  $\mathbf{b}$ . If  $x(\theta_a)$  and  $x(\theta_b)$  are the respective optimal basic solutions for two distinct values  $\theta_a < \theta_b$ , then for all  $0 \leq \mu \leq 1$

$$\mathbf{x}_\mu = \mu x(\theta_a) + (1 - \mu)x(\theta_b)$$

is feasible for  $(LP_{\theta_\mu})$  where  $\theta_\mu = \mu\theta_a + (1 - \mu)\theta_b$ . Thus  $(LP_{\theta_\mu})$  has a feasible solution and the dual linear program  $\max\{\mathbf{u}(\mathbf{b} + \theta\mathbf{g}) : \mathbf{u}\mathbf{A} \leq \mathbf{c}\}$  has a *feasible* solution no matter what  $\theta$ , since by assumption e.g.  $(LP_{\theta_a})$  has a finite optimal solution. Thus  $(LP_\theta)$  is bounded from below. Hence,  $z(\theta_\mu)$  exists and is finite. From the feasibility of  $\mathbf{x}_\mu$

$$z(\mu\theta_a + (1 - \mu)\theta_b) \leq \mu z(\theta_a) + (1 - \mu)z(\theta_b) \quad \text{for all } 0 \leq \mu \leq 1 ,$$

since we are minimizing, i.e.,  $z(\theta)$  is a convex function of  $\theta$ . Thus if  $z(\theta)$  is defined at all then it is defined over an interval of the real line (which might be a single point). Assume that  $(LP_\theta)$  has a finite optimal solution for  $\theta = 0$ , let  $B$  be an optimal basis found by a simplex method for  $\theta = 0$  and consider  $\mathbf{x}(\theta)$ :

$$\mathbf{x}_B(\theta) = \mathbf{B}^{-1}\mathbf{b} + \theta\mathbf{B}^{-1}\mathbf{g}, \quad \mathbf{x}_R(\theta) = \mathbf{0}.$$

The reduced cost vector given by  $B$  is not, but the feasibility of the solution vector  $\mathbf{x}(\theta)$  is affected if we vary  $\theta$ . Thus we have to ensure that  $\mathbf{x}_B(\theta) \geq \mathbf{0}$ . So let  $\bar{\mathbf{g}} = \mathbf{B}^{-1}\mathbf{g}$  and  $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ . From the condition that  $\bar{\mathbf{b}} + \theta\bar{\mathbf{g}} \geq \mathbf{0}$  we find that  $\mathbf{x}_B(\theta) \geq \mathbf{0}$  for  $\theta$  in the interval

$$\max \left\{ -\frac{\bar{b}_i}{\bar{g}_i} : \bar{g}_i > 0, i = 1, \dots, m \right\} \leq \theta \leq \min \left\{ \frac{\bar{b}_i}{|\bar{g}_i|} : \bar{g}_i < 0, i = 1, \dots, m \right\}.$$

If either quantity on the left or the right is undefined, then it is replaced by  $-\infty$  for the maximum, by  $+\infty$  for the minimum, respectively. We thus have locally, i.e. in the “vicinity” of  $\theta = 0$ ,  $z(\theta) = z_B + \theta c_B \mathbf{B}^{-1}\mathbf{g}$ , and hence  $z(\theta)$  is locally a linear function of  $\theta$ . If one increases or decreases  $\theta$  beyond the bounds just stated, one loses primal feasibility of the solution  $\mathbf{x}(\theta)$ , but the reduced cost vector still displays optimality of the basis  $B$ . Using an  $\varepsilon$ -argument we can thus apply the dual simplex algorithm to reoptimize the changed linear program as we have a dual basis for  $(LP_\theta)$ . The reoptimization produces a new basis that displays optimality and one repeats the process. As there are only finitely many bases, there are only a finite number of “break points”, i.e.,  $z(\theta)$  is a piecewise linear convex function of  $\theta$ .

The problem of parametrizing the objective function coefficients is treated in Exercise 6.10 below. The analysis of changing elements of the matrix  $A$  is treated in the text.

The problem of adding one or more “new” variables or one or more “new” constraints to a linear program that we have optimized leads to reoptimization problems using a primal and a dual simplex algorithm, respectively, and can be carried out efficiently – barring unlikely data configurations, of course.

**A Dynamic Simplex Algorithm:** The combination of the primal and dual simplex algorithm constitutes a powerful tool to solve large-scale linear programs of the form

$$(LP_H) \quad \min\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{H}\mathbf{x} \leq \mathbf{h}, \mathbf{x} \geq \mathbf{0}\},$$

where  $A$  is an  $m \times n$  matrix,  $H$  is an  $f \times n$  matrix and the vectors are dimensioned accordingly. Let  $N$  be the set of variables,  $n = |N|$  a truly large number,  $F$  the set of inequality constraints and  $f = |F|$  a truly large number as well. In the analysis of  $(LP_H)$  we make the **assumption** that all variables are bounded from above so that if  $(LP_H)$  is feasible then it has a finite optimal solution.

For any nonempty subset  $P \subseteq N$  and  $L \subseteq F$  define  $(LP_P^L)$  to be the subproblem of  $(LP_H)$  given by

$$(LP_P^L) \quad \min\{\mathbf{c}_P \mathbf{x}_P : \mathbf{A}_P \mathbf{x}_P = \mathbf{b}, \mathbf{H}_P^L \mathbf{x}_P \leq \mathbf{h}_L, \mathbf{x}_P \geq \mathbf{0}\}.$$

That is we assume that all original equations are in the problem, but that among the inequalities we have “activated” only a small subset  $L$  and that only a small subset  $P$  of all variables has been activated as well. As usually we denote by  $a_j$  the column  $j$  of  $A$ , by  $h_P^i$  the  $i^{th}$  row of  $H_P^L$  and by  $h_j^L$  the column  $j$  of the matrix  $H^L$  which comprises all the columns with index in  $N$ .

We denote by  $\mathbf{x}_P$  an optimal solution to  $(LP_P^L)$  and by  $\mathbf{x}$  the vector with components  $\mathbf{x}_P$  and  $\mathbf{x}_{N-P} = \mathbf{0}$ . We denote by  $\mathbf{u}$  the row vector of the optimal dual variables corresponding to the

equations and by  $v_L$  the row vector of the optimal dual variables corresponding to the “active” linear inequalities.

The following *dynamic simplex algorithm* solves the problem  $(LP_H)$  by solving a sequence of problems  $(LP_P^L)$ . We assume in the algorithm that the original problem data  $A$ ,  $H$ ,  $b$ ,  $h$  and  $c$  are stored *separately* from the working arrays  $A_P$ ,  $H_P^L$ ,  $b$ ,  $h_L$  and  $c_P$  used in the simplex algorithms.

### **Dynamic Simplex Algorithm ( $n, m, f, A, H, b, h, c$ )**

- Step 0:** Select a subset  $P \subseteq N$  with  $1 \leq |P| \ll n$  and  $L \subseteq F$  with  $0 \leq |L| \ll f$ . Set  $z_{LOW} := -\infty$ .  
 Solve  $(LP_P^L)$  by the primal simplex algorithm and let  $x$  be the optimal solution with objective function value  $z$  and  $u, v_L$  be the optimal dual solution found by the algorithm.  
**go to** Step 2.
- Step 1:** Reoptimize  $(LP_P^L)$  by the primal simplex algorithm, update  $x, z, u, v_L$ .
- Step 2:** Compute  $\bar{c}_j := c_j - ua_j - v_L h_j^L$  for all  $j \in N - P$ , set  $Q := \{j \in N - P : \bar{c}_j < 0\}$ .  
**if**  $Q = \emptyset$  **then go to** Step 3. Replace  $P$  by  $P \cup Q$ ; **go to** Step 1.
- Step 3:** **if**  $z > z_{LOW}$  **then**  
     set  $z_{LOW} := z$ ; find  $S := \{i \in L : \text{the slack of constraint } i \text{ is basic}\}$ ;  
     replace  $L$  by  $L - S$  and reset  $v_L, H_P^L, h_L$  etc. accordingly.  
**endif.**
- Step 4:** Find a subset  $K \subseteq \{i \in F - L : h_P^i x_P > h_i\}$ .  
**if** none exists **then**  
     **stop** “ $x$  is an optimal solution to  $(LP_H)$ ”.  
**else**  
     replace  $L$  by  $L \cup K$ ; **go to** Step 5.  
**endif.**
- Step 5:** Reoptimize  $(LP_P^L)$  by the dual simplex algorithm, update  $x, z, u, v_L$ ; **go to** Step 2.

Step 2 of the dynamic simplex algorithm is called the **column generation** step. Step 4 is the **row generation** step of the algorithm. Step 3 is called **purging**. Here the inequality constraints for which the associated slack variables are in the optimal basis of  $(LP_P^L)$  are “deactivated” or purged from the actual *working arrays* if the objective function value  $z$  increased strictly with respect to the objective function value  $z_{LOW}$  when purging was performed last.

$z_{LOW}$  is the objective function value of the linear program  $(LP_N^L)$  and thus a lower bound on the “true” minimum, i.e. the objective function value of  $(LP_H)$ . If purging is done when  $z$  “equals”  $z_{LOW}$  then the algorithm may cycle between Steps 2 and 5 and such cycling has been observed by the author in computational practice. When this possibility is ruled out as it is in the dynamic simplex algorithm, then the finiteness of the algorithm follows from the finiteness of the primal and dual simplex algorithms and the finiteness of  $|N|$  and  $|F|$  if an exceptional case does not occur; the general case is discussed in the text.

## 6.1 Exercises

---

### Exercise 6.1

Show that the dual linear program of the linear program DUAL is the linear program PRIMAL.

---

First we bring DUAL into the same form as the problem PRIMAL, i.e. in the form

$$\begin{array}{lll} \min & -\mathbf{u}\mathbf{b}_1 & -\mathbf{v}\mathbf{b}_2 & -\mathbf{w}\mathbf{b}_3 \\ \text{s.t.} & -\mathbf{u}\mathbf{A}_{13} & -\mathbf{v}\mathbf{A}_{23} & -\mathbf{w}\mathbf{A}_{33} = -\mathbf{c}_3 \\ & -\mathbf{u}\mathbf{A}_{11} & -\mathbf{v}\mathbf{A}_{21} & -\mathbf{w}\mathbf{A}_{31} \geq -\mathbf{c}_1 \\ & -\mathbf{u}\mathbf{A}_{12} & -\mathbf{v}\mathbf{A}_{22} & -\mathbf{w}\mathbf{A}_{32} \leq -\mathbf{c}_2 \\ & -\mathbf{u} & & \text{free} \\ & \mathbf{v} & & \geq \mathbf{0} \\ & \mathbf{w} & & \leq \mathbf{0} \end{array}$$

We transpose the (in)equalities and substitute the variables  $-\mathbf{u}^T = \mathbf{u}'$ ,  $-\mathbf{v}^T = \mathbf{v}'$  and  $-\mathbf{w}^T = \mathbf{w}'$  and get

$$\begin{array}{lll} \min & \mathbf{b}_1^T \mathbf{u}' & + \mathbf{b}_2^T \mathbf{v}' & + \mathbf{b}_3^T \mathbf{w}' \\ \text{s.t.} & \mathbf{A}_{13}^T \mathbf{u}' & + \mathbf{A}_{23}^T \mathbf{v}' & + \mathbf{A}_{33}^T \mathbf{w}' = -\mathbf{c}_3^T \\ & \mathbf{A}_{11}^T \mathbf{u}' & + \mathbf{A}_{21}^T \mathbf{v}' & + \mathbf{A}_{31}^T \mathbf{w}' \geq -\mathbf{c}_1^T \\ & \mathbf{A}_{12}^T \mathbf{u}' & + \mathbf{A}_{22}^T \mathbf{v}' & + \mathbf{A}_{32}^T \mathbf{w}' \leq -\mathbf{c}_2^T \\ & \mathbf{u}' & & \text{free} \\ & \mathbf{v}' & & \leq \mathbf{0} \\ & \mathbf{w}' & & \geq \mathbf{0} \end{array}$$

Now we write the dual of it where we denote the dual variables by  $\mathbf{z}'$ ,  $\mathbf{x}'$  and  $\mathbf{y}'$ .

$$\begin{array}{lll} \max & -\mathbf{z}' \mathbf{c}_3^T & -\mathbf{x}' \mathbf{c}_1^T & -\mathbf{y}' \mathbf{c}_2^T \\ \text{s.t.} & \mathbf{z}' \mathbf{A}_{13}^T & + \mathbf{x}' \mathbf{A}_{11}^T & + \mathbf{y}' \mathbf{A}_{12}^T = \mathbf{b}_1^T \\ & \mathbf{z}' \mathbf{A}_{23}^T & + \mathbf{x}' \mathbf{A}_{21}^T & + \mathbf{y}' \mathbf{A}_{22}^T \geq \mathbf{b}_2^T \\ & \mathbf{z}' \mathbf{A}_{33}^T & + \mathbf{x}' \mathbf{A}_{31}^T & + \mathbf{y}' \mathbf{A}_{32}^T \leq \mathbf{b}_3^T \\ & \mathbf{z}' & & \text{free} \\ & \mathbf{x}' & & \geq \mathbf{0} \\ & \mathbf{y}' & & \leq \mathbf{0} \end{array}$$

After transposing the (in)equalities, substituting  $\mathbf{z}^T = \mathbf{z}'$ ,  $\mathbf{x}^T = \mathbf{x}'$  and  $\mathbf{y}^T = \mathbf{y}'$  and rearranging we get

$$\begin{aligned}
 & \max -c_1x - c_2y - c_3z \\
 \text{s.t. } & A_{11}x + A_{12}y + A_{13}z = b_1 \\
 & A_{21}x + A_{22}y + A_{23}z \geq b_2 \\
 & A_{31}x + A_{32}y + A_{33}z \leq b_3 \\
 & x \geq 0 \\
 & y \leq 0 \\
 & z \text{ free}
 \end{aligned}$$

which is exactly the problem PRIMAL, since  $\max(-c_1x - c_2y - c_3z) = -\min(c_1x + c_2y + c_3z)$ .

---

### Exercise 6.2

*Consider the linear programming problem*

$$\max \left\{ \sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_j x_j \leq a_0, 0 \leq x_j \leq 1, \text{ for } j = 1, \dots, n \right\}$$

where  $c_j > 0$ ,  $a_j > 0$  for all  $j$  and  $a_0 > 0$ . Find an optimal solution. (Hint: Note the upper bounds.)

---

We bring the problem into the form  $(LP_u)$  by introducing a slack variable  $x_{n+1} \geq 0$  and  $c_{n+1} = 0$ ,  $a_{n+1} = 1$ ,  $u_j = 1$  for  $1 \leq j \leq n$  and  $u_{n+1} = a_0$ . If  $\sum_{j=1}^n a_j \leq a_0$  then  $x_j = 1$  for  $1 \leq j \leq n$  is optimal and there is nothing to prove. So suppose that  $\sum_{j=1}^n a_j > a_0$ . The solution set is nonempty and bounded. Thus an optimal solution exists. From Chapter 5 we have that the optimal solution is of the following form. If  $a_k$  is a basis and  $J_0$ ,  $J_1$  is a partition of the variable set except  $x_k$  such that  $x_j = 0$  for all  $j \in J_0$  and  $x_j = u_j$  for all  $j \in J_1$ , with  $\bar{c}_j \leq 0$  for all  $j \in J_0$  and  $\bar{c}_j \geq 0$  for all  $j \in J_1$ , then the basis is optimal. (Note that we have changed the direction of the inequalities for the reduced costs since here we have a maximization problem.) Thus for all  $j \in J_0$  we have  $\bar{c}_j = c_j - \frac{c_k}{a_k} a_j \leq 0$ , i.e.  $\frac{c_k}{a_k} \geq \frac{c_j}{a_j}$ . Similarly, for all  $j \in J_1$  we get  $\frac{c_k}{a_k} \leq \frac{c_j}{a_j}$ . So suppose that we sort the variables with respect to the ratio  $\frac{c_j}{a_j}$  in descending order and find  $k$  such that  $\sum_{j=1}^{k-1} a_j \leq a_0$  but  $\sum_{j=1}^k a_j > a_0$ . Since  $\sum_{j=1}^n a_j > a_0$  by assumption, we have  $k \leq n$ . Then we claim that the optimal solution is given by  $x_j = 1$  for all  $j \in J_1 = \{1, \dots, k-1\}$ ,  $x_k = (a_0 - \sum_{j=1}^{k-1} a_j)/a_k$  and  $x_j = 0$  for all  $j \in J_0 = \{k+1, \dots, n+1\}$ . The objective function of this solution is

$$z_P = \sum_{j=1}^{k-1} c_j + \frac{c_k}{a_k} (a_0 - \sum_{j=1}^{k-1} a_j).$$

We prove our claim by showing that the dual of the original linear program has a feasible solution with an objective function value equal to the above value. The dual program is:

$$\min \{a_0 u + \sum_{j=1}^n v_j : u a_j + v_j \geq c_j \text{ for all } 1 \leq j \leq n, u \geq 0, \mathbf{v} \geq \mathbf{0}\}.$$

From the complementary slackness conditions we have

$$v_j = 0 \text{ for all } j \in J_0 \quad \text{and} \quad ua_j + v_j = c_j \text{ for all } j \in J_1 .$$

Adding all the equations for  $j \in J_1$  we get  $\sum_{j \in J_1} v_j = \sum_{j \in J_1} c_j - u \sum_{j \in J_1} a_j$ . The dual objective value is  $z_D = ua_0 + \sum_{j=1}^n v_j$  or after substituting the values  $v_j$  for  $j \in J_0$  and  $\sum_{j \in J_1} v_j$  for  $j \in J_1$  we get

$$z_D = ua_0 + \sum_{j \in J_1} c_j - u \sum_{j \in J_1} a_j + v_k .$$

We distinguish the following two cases for the value of  $x_k$  (note that from the definition of  $k$ ,  $x_k$  cannot be one).

- (i) If  $0 < x_k < 1$ , then from the complementary slackness condition we have  $v_k = 0$  and  $ua_k + v_k = c_k$ , i.e.  $u = \frac{c_k}{a_k}$ . Then we have  $z_D = \frac{c_k}{a_k}(a_0 - \sum_{j \in J_1} a_j) + \sum_{j \in J_1} c_j = z_P$ .
- (ii) If  $x_k = 0$ , then  $a_0 = \sum_{j \in J_1} a_j$ ,  $v_k = 0$  and thus  $z_D = \sum_{j \in J_1} c_j = z_P$ .

Thus in both cases  $z_D = z_P$ , which proves the claim.

---

### Exercise 6.3

Consider the following capital budgeting model (which assumes a perfect capital market since the rates for borrowing and lending money are equal to a single “market rate”  $r$  and there is no limitation on the borrowing/lending activities):

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j + y_T \\ \text{subject to} \quad & - \sum_{j=1}^n a_j^1 x_j + y_1 \leq s_1 \\ & - \sum_{j=1}^n a_j^i x_j - (1+r)y_{i-1} + y_i \leq s_i \quad \text{for } i = 2, \dots, T \\ & 0 \leq x_j \leq 1 \quad \text{for all } j = 1, \dots, n , \end{aligned}$$

where  $n$  is the number of possible projects,  $T$  is the number of time periods considered,  $s_i$  are (exogenous) investment funds for period  $i$ ,  $a_j^i$  is the cash flow associated with project  $j$  at the end of period  $i$ ,  $c_j$  is the net present value in year  $T$  for project  $j$  of all cash flows subsequent to the year  $T$  and discounted at the interest rate  $r$ ,  $r$  is the market rate of interest,  $y_i$  is the amount borrowed (if negative) or lent (if positive) in period  $i$  and  $x_j$  is the fraction of project  $j$  undertaken.

Show that this problem always has an optimal solution with **no** fractional projects. Find an optimal solution and give an economic interpretation of the resulting decision rule. (Hint: The net present value  $NPV_j$  evaluated at the interest rate  $r$  of an (infinite) stream of cash flows  $a_j^i$  for  $i = 1, 2, \dots$  equals  $\sum_{i=1}^{\infty} (1+r)^{-i} a_j^i$ , where  $j \in \{1, \dots, n\}$  is any project. Moreover,  $a_j^i > 0$  means that project  $j$  generates an inflow of cash in period  $i$ , while  $a_j^i < 0$  is an outflow of cash.)

---

To solve the problem we start from its dual linear program:

$$\begin{aligned} \min \quad & \sum_{i=1}^T s_i u_i + \sum_{j=1}^n v_j \\ \text{s.t.} \quad & -\sum_{i=1}^T a_j^i u_i + v_j \geq c_j \quad \text{for } 1 \leq j \leq n \\ & u_i - (1+r)u_{i+1} = 0 \quad \text{for } 1 \leq i \leq T-1 \\ & u_T = 1 \\ & \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

From the last  $T$  constraints of the dual problem we solve for the variables  $u_i$ , for  $i = 1, \dots, T$  by backward substitution, i.e.  $u_T = 1$ ,  $u_{T-1} = 1 + r$ ,  $u_{T-2} = (1 + r)^2, \dots, u_1 = (1 + r)^{T-1}$ . Thus for every feasible solution to the dual we have  $u_i = (1 + r)^{T-i}$ ,  $1 \leq i \leq T$ . Now the first  $n$  constraints of the dual become

$$v_j \geq c_j + \sum_{i=1}^T a_j^i (1 + r)^{T-i} \quad 1 \leq j \leq n.$$

Since we want to minimize  $\sum_{j=1}^n v_j$  the optimal values for  $v_1, \dots, v_n$  are given by

$$v_j = \begin{cases} 0 & \text{if } c_j + \sum_{i=1}^T (1 + r)^{T-i} a_j^i \leq 0, \\ c_j + \sum_{i=1}^T (1 + r)^{T-i} a_j^i & \text{otherwise,} \end{cases}$$

and hence, we calculate the optimal dual objective value

$$z_D = \sum_{i=1}^T s_i (1 + r)^{T-i} + \sum_{j \in N_+} c_j + \sum_{j \in N_+} \sum_{i=1}^T a_j^i (1 + r)^{T-i},$$

where  $N_+ = \{j \in \{1, \dots, n\} : c_j + \sum_{i=1}^T (1 + r)^{T-i} a_j^i > 0\}$ . Using the complementary slackness conditions we have the following conclusions for the primal constraints and variables. Since all  $u_i > 0$  for  $1 \leq i \leq T$  the first  $T$  constraints of the primal hold as equalities which we write as

$$y_1 = s_1 + \sum_{j=1}^n a_j^1 x_j \tag{6.10}$$

$$y_i = s_i + (1 + r)y_{i-1} + \sum_{j=1}^n a_j^i x_j \quad 2 \leq i \leq T \tag{6.11}$$

Multiplying the equation for  $y_i$  by  $(1 + r)^{T-i}$  for  $1 \leq i \leq T$  and adding all the equations we eliminate variables  $y_i$  for  $1 \leq i < T$  and get

$$y_T = \sum_{i=1}^T s_i (1 + r)^{T-i} + \sum_{i=1}^T (1 + r)^{T-i} \sum_{j=1}^n a_j^i x_j. \tag{6.12}$$

Since the variables  $y_1, \dots, y_T$  are not restricted in sign, every choice of  $x_j \in \{0, 1\}$  for  $1 \leq j \leq n$  gives a primal feasible solution. From the complementary slackness conditions we get that  $x_j = 1$  for all  $j \in N_+$  and  $x_j = 0$  for all  $j$  such that  $c_j + \sum_{i=1}^T a_j^i (1 + r)^{T-i} < 0$ . Setting  $x_j = 0$  for all  $j \notin N_+$

we have a primal feasible solution. To show that the solution to the primal obtained this way is optimal we have to compute its objective function value  $z_P$ . Using (6.12) we calculate

$$z_P = \sum_{j \in N_+} c_j + \sum_{i=1}^T s_i (1+r)^{T-i} + \sum_{j \in N_+} \sum_{i=1}^T (1+r)^{T-i} a_j^i = z_D ,$$

which proves the optimality of the solution because  $z_P = z_D$ . Since in this solution all  $x_j$ 's are either zero or one we have proven that there exists an optimal solution with no fractional projects.

To summarize, the optimal solution to the primal problem is given by

$$x_j = \begin{cases} 1 & \text{if } j \in N_+, \\ 0 & \text{otherwise,} \end{cases} \quad (6.13)$$

and the values of the  $y$  variables are given by (6.10) and (6.11) using (6.13). Thus the selection rule is that we accept project  $j$  if  $j \in N_+$ , i.e. if  $c_j + \sum_{i=1}^T a_j^i (1+r)^{T-i} > 0$ . By definition,  $c_j = \sum_{i=T+1}^{\infty} a_j^i (1+r)^{T-i}$ . Consequently,  $j \in N_+$  if and only if  $\text{NPV}_j > 0$ , i.e. project  $j$  is accepted if and only if the net present value  $\text{NPV}_j$  of all cash flows associated with the project is positive. By the way, this shows that the *net present value rule* of Financial Economics is consistent with the assumption that the capital markets are perfect (which, alas, they are not). See also Exercise 10.9.

---

#### Exercise 6.4

*A corn silo with given capacity  $K$  is operated for the next  $n$  years. Per unit buying and selling prices for corn as well as the inventory cost per unit stored are known, but change from year to year. Linearity of cost and revenue from the quantity is assumed. You can buy corn at the beginning of each year but you can sell corn only at the end of the year, i.e. in order to sell corn it has to be stored at least one year. The company that operates the silo has unlimited funds for the purchase of corn and can sell any amount of it, i.e. the market absorbs any amount at the given selling price. The silo is initially empty and has to be empty at the end of the last year.*

- (a) *Formulate the problem so as to maximize profit (= revenue – cost).*
  - (b) *What can you prove about the dependence of the optimal solution and profit on the capacity  $K$  of the silo?*
- 

We shall give two seemingly different, but equivalent formulations of this problem as a linear program. In the *first formulation* we choose the decision variables as follows.

For each year  $j$  the quantity  $z_j$  to be stored equals the quantity  $z_{j-1}$  stored in year  $j-1$  plus the quantity  $y_j$  bought at the beginning of the year minus the quantity  $x_{j-1}$  sold at the end of year  $j-1$ , i.e.,

$$z_j = z_{j-1} + y_j - x_{j-1}$$

for  $1 \leq j \leq n$ , where  $z_0 = x_0 = 0$  because initially the silo is empty. Since the silo must be empty at the end of the last year, we get the additional constraint

$$-z_n + x_n = 0.$$

Since the corn needs to be stored one year it follows that  $x_j \leq z_j$  for  $1 \leq j \leq n$ . Because the silo has a finite capacity  $K > 0$  we get the constraints  $z_j \leq K$  for  $1 \leq j \leq n$  and all quantities are nonnegative. Letting  $r_j$  be the unit selling price,  $c_j$  the unit buying price and  $h_j$  the unit inventory cost for corn in  $1, \dots, n$  we thus have the following linear programming formulation of the problem with  $n+1$  equations,  $2n-1$  inequalities and  $3n$  variables, of which  $2n$  are required to be nonnegative:

$$\begin{aligned} \max \quad & \sum_{j=1}^n (r_j x_j - c_j y_j - h_j z_j) \\ \text{s.t.} \quad & y_1 - z_1 = 0 \\ & z_j - z_{j-1} - y_j + x_{j-1} = 0 \quad 2 \leq j \leq n \\ & -z_n + x_n = 0 \\ & -z_j + x_j \leq 0 \quad 1 \leq j \leq n-1 \\ & z_j \leq K \quad 1 \leq j \leq n \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Since  $\mathbf{x} = \mathbf{0}$ ,  $\mathbf{y} = \mathbf{0}$ ,  $\mathbf{z} = \mathbf{0}$  is a feasible solution the maximum profit from operating the silo is nonnegative, no matter what the data are. Moreover, since  $K > 0$  is finite, the solution set is bounded and thus a maximum profit solution exists. Suppose that the data are such that the optimum profit is positive. Since for any feasible solution  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  the vector  $(\lambda \mathbf{x}, \lambda \mathbf{y}, \lambda \mathbf{z})$  for  $0 \leq \lambda \leq 1$  is feasible as well, it follows that in every optimal solution  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ ,  $z_j = K$  for some  $j \in \{1, \dots, n\}$ , i.e., the silo is operated at capacity for at least one of the time periods. Moreover, consider the dual linear program. The silo capacity  $K$  appears only in the right-hand side of the  $n$  "capacity" constraints and all other right-hand side coefficients are zero. Denoting the dual variables of the capacity constraints  $z_j \leq K$  by  $v_j$  the objective function of the dual becomes  $K \sum_{j=1}^n v_j$ . Since  $K$  is a constant, the optimal dual solution is independent of  $K$ , i.e., it suffices to minimize  $\sum_{j=1}^n v_j$  over the dual feasible region to get the optimal solution. Then the optimal value of the problem is  $K$  times the optimal value of  $\sum_{j=1}^n v_j$ . Thus the optimal profit is either zero or proportional to the size of  $K$  of the silo. Of course, in our scenario we make the assumption that the storage costs  $h_j$  are independent of the silo size which may be unrealistic.

In the *second formulation* we choose the quantities  $x_{ij}$  of corn to be bought at the beginning of year  $i$ , to be stored in the silo, and to be sold at the end of the year  $j$  as our decision variables where  $1 \leq i \leq j \leq n$ . The profit that accrues from buying, storing and selling  $x_{ij}$  units of corn is given by  $r_j - c_i - \sum_{\ell=i}^j h_\ell$ . We thus have the following linear program with  $n$  constraints and  $n(n+1)/2$  nonnegative variables:

$$\begin{aligned} \max \quad & \sum_{j=1}^n \sum_{i=1}^j (r_j - c_i - \sum_{\ell=i}^j h_\ell) x_{ij} \\ \text{s.t.} \quad & \sum_{\ell=1}^i \sum_{j=i}^n x_{\ell j} \leq K \quad \text{for } 1 \leq i \leq n \\ & x_{ij} \geq 0 \quad \text{for } 1 \leq i \leq j \leq n. \end{aligned}$$

Here the linear constraints express the fact that for each year the silo capacity must not be exceeded and we can derive the same conclusions as above for the optimal policy of buying,

storing and selling corn. The two formulations are interrelated by the variable transformation

$$x_i = \sum_{\ell=1}^i x_{\ell i}, \quad y_i = \sum_{\ell=i}^n x_{i\ell}, \quad z_i = \sum_{\ell=1}^i \sum_{j=i}^n x_{\ell j},$$

which is a linear transformation. In Chapter 7 we give a general, algebraic method to compute e.g. the first LP formulation from the second one, which is an important tool in *computing* different formulations of the same problem especially in connection with combinatorial optimization problem –see Chapter 10. Moreover, it can be shown using elementary row operations that the constraint matrix of e.g. the second formulation is *totally unimodular* –see Chapter 10– and thus the optimal policy of buying, storing and selling corn is of the “bang-bang” variety: whenever corn is bought, the silo is filled to capacity and whenever corn is sold, the entire silo is emptied. The corresponding decision rule, however, is data-dependent and (apparently) not easy to state in explicit form.

---

### Exercise 6.5

Let  $A, D$  be matrices of size  $m \times n$  and  $p \times n$ , respectively,  $b \in \mathbb{R}^m$ ,  $d \in \mathbb{R}^p$  column vectors,  $c \in \mathbb{R}^n$  a row vector and  $z \in \mathbb{R}$ . Using Lemma 2 and Theorem 4 show the following statements:

- (i)  $\{x \in \mathbb{R}^n : Ax \leq b\} = \emptyset$  if and only if  $\{u \in \mathbb{R}^m : uA = 0, ub < 0, u \geq 0\} \neq \emptyset$ .
  - (ii)  $\{x \in \mathbb{R}^n : Ax = b, Dx \leq d\} = \emptyset$  if and only if  $\{(u, v) \in \mathbb{R}^{m+p} : uA + vD = 0, ub + vd < 0, v \geq 0\} \neq \emptyset$ .
  - (iii)  $\{x \in \mathbb{R}^n : Ax = b, Dx \leq d, cx > z\} \neq \emptyset$  if and only if  $\{(u, v) \in \mathbb{R}^{m+p} : uA + vD = 0, ub + vd < 0, v \geq 0\} = \emptyset$  and  $\{(u, v) \in \mathbb{R}^{m+p} : uA + vD = c, ub + vd \leq z, v \geq 0\} = \emptyset$ .
  - (iv)  $\{x \in \mathbb{R}^n : Ax = 0, x \geq 0, x \neq 0\} \neq \emptyset$  if and only if  $\{u \in \mathbb{R}^m : uA > 0\} = \emptyset$ .
- 

**(i)** Suppose that  $\mathcal{X}_1 = \{x \in \mathbb{R}^n : Ax \leq b\} = \emptyset$  and  $\mathcal{U}_1 = \{u \in \mathbb{R}^m : uA = 0, ub < 0, u \geq 0\} = \emptyset$ . Since  $\mathcal{U}_1 = \emptyset$  and  $u = 0$  satisfies  $uA = 0$  and  $u \geq 0$  it follows that the linear program  $\min\{ub : uA = 0, u \geq 0\}$  has an optimal solution of value 0, and thus by strong duality (Theorem 3) its dual  $\max\{0x : Ax \leq b\}$  is feasible, which contradicts that  $\mathcal{X}_1 = \emptyset$ .

On the other hand assume that  $\mathcal{X}_1 \neq \emptyset$  and  $\mathcal{U}_1 \neq \emptyset$ , i.e. that there exists  $u \in \mathbb{R}^m$  such that  $uA = 0$ ,  $ub < 0$ ,  $u \geq 0$  and  $x \in \mathbb{R}^m$  such that  $Ax \leq b$ . Consider the (LP)  $\max\{0x : Ax \leq b\}$ . Its dual is  $\min\{ub : uA = 0, u \geq 0\}$ . Since both primal and dual are feasible, from weak duality we have that  $ub \geq 0$  which is a contradiction.

**(ii)** Replacing the set of equations  $Ax = b$  by the set of inequalities  $Ax \leq b$  and  $-Ax \leq -b$  we have that  $\{x \in \mathbb{R}^n : Ax = b, Dx \leq d\} = \{x \in \mathbb{R}^n : A'x \leq b'\}$  where  $A' = \begin{pmatrix} A \\ -A \\ D \end{pmatrix}$  and  $b' = \begin{pmatrix} b \\ -b \\ d \end{pmatrix}$ .

By part (i) we have that  $\{x \in \mathbb{R}^n : A'x \leq b'\} = \emptyset$  if and only if  $\mathcal{U} = \{u' \in \mathbb{R}^{2m+p} : u'A' = 0, u'b < 0, u' \geq 0\} \neq \emptyset$ . Breaking  $u'$  into three subvectors  $y \in \mathbb{R}^m$ ,  $w \in \mathbb{R}^m$  and  $v \in \mathbb{R}^p$  we have

$$u'A' = yA - wA + vD, \quad u'b = yb - wb + vd, \quad y, w, v \geq 0.$$

Replacing  $y - w$  by a vector  $u \in \mathbb{R}^m$  we have that  $\mathcal{U} = \{(u, v) \in \mathbb{R}^{m+p} : uA + vD = 0, ub + vd < 0, v \geq 0\}$  and the assertion follows.

**(iii)** Let  $x \in \mathbb{R}^n$  be such that  $Ax = b$ ,  $Dx \leq d$  and  $cx > z$ . First suppose that there exist  $u \in \mathbb{R}^m$  and  $v \in \mathbb{R}^p$  such that  $uA + vD = 0$ ,  $ub + vd < 0$  and  $v \geq 0$ . Then we have  $0 = uAx + vDx \leq ub + vd$  which contradicts  $ub + vd < 0$ . Next, suppose that there exist  $u \in \mathbb{R}^m$  and  $v \in \mathbb{R}^p$  such that  $uA + vD = c$ ,  $ub + vd \leq z$  and  $v \geq 0$ . Then we have  $cx = uAx + vDx \leq ub + vd \leq z$  which contradicts  $cx > z$ . Thus the “only if” part is proven. To prove the “if” part, suppose that  $\mathcal{U}_1 = \{(u, v) \in \mathbb{R}^{m+p} : uA + vD = 0, ub + vd < 0, v \geq 0\} = \emptyset$  and  $\mathcal{U}_2 = \{(u, v) \in \mathbb{R}^{m+p} : uA + vD = c, ub + vd \leq z, v \geq 0\} = \emptyset$ . Since  $\mathcal{U}_1 = \emptyset$  we get from part (ii) that  $\mathcal{X}_1 = \{x \in \mathbb{R}^n : Ax = b, Dx \leq d\} \neq \emptyset$ . Suppose that  $cx \leq z$  for all  $x \in \mathcal{X}_1$ . Then the LP  $\max\{cx : x \in \mathcal{X}_1\}$  has a bounded objective function from above. The dual of this problem is  $\min\{ub + vd : uA + vD = c, v \geq 0\}$ . Since the primal is feasible and bounded, the dual is feasible and by strong duality we have that  $cx = ub + vd$  at the optimum. But since  $\mathcal{U}_2 = \emptyset$  we necessarily have that  $ub + vd > z$  and thus  $cx > z$  which contradicts  $cx \leq z$ .

This part can be also be proven like we proved part (ii) using Theorem 4.

**(iv)** Assume that both sets  $\{u \in \mathbb{R}^m : uA > 0\}$  and  $\{x \in \mathbb{R}^n : Ax = 0, x \geq 0, x \neq 0\}$  are nonempty, i.e. there exist  $u \in \mathbb{R}^m$  and  $x \in \mathbb{R}^n$  such that  $uA > 0$ ,  $Ax = 0$ ,  $x \geq 0$  and  $x \neq 0$ . Without restriction of generality we can assume that  $uA \geq e^T = (1, \dots, 1)$ . To see this let  $uA \geq (\alpha_1, \dots, \alpha_n) > 0$  and  $D = \text{diag}(\alpha_1, \dots, \alpha_n)$ . Then  $uAD^{-1} \geq e^T$  and  $AD^{-1}(Dx) = 0$ ,  $Dx \geq 0$ ,  $Dx \neq 0$ , i.e. the scaled matrix  $AD^{-1}$  has the assumed properties. Consider the (LP)  $\max\{e^T x : Ax = 0, x \geq 0\}$  and its dual  $\min\{u0 : uA \geq e^T\}$ , both of which by assumption have feasible solutions. But by Remark 3.1 the primal (LP) has an unbounded optimum, which contradicts Remark 6.2. Consequently, both sets cannot be nonempty.

On the other hand suppose  $\{x \in \mathbb{R}^n : Ax = 0, x \geq 0, x \neq 0\} = \emptyset$ , i.e.  $x = 0$  is the unique solution to  $Ax = 0$ ,  $x \geq 0$ . It follows that the linear program (LP) has the unique solution  $x = 0$  and thus from strong duality theorem (Theorem 4) we have that  $\min\{u0 : uA \geq e^T\} = 0$  and thus there exists  $u \in \mathbb{R}^m$  such that  $uA \geq e^T > 0$ .

### Exercise 6.6

Let  $\mathcal{X}^\leq = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$  and  $\mathcal{Y}^\leq = \{x \in \mathbb{R}^n : Dx \leq g, x \geq 0\}$ .

- (i) Suppose  $\emptyset \neq \mathcal{Y}^\leq \subseteq \mathcal{X}^\leq$ . Then either  $\mathcal{X}^\leq = \mathcal{Y}^\leq$  or there exist a row vector  $f \in \mathbb{R}^n$  and a scalar  $f_0$  such that  $\mathcal{Y}^\leq \subseteq \{x \in \mathbb{R}^n : fx \leq f_0\}$  and  $\mathcal{X}^\leq \cap \{x \in \mathbb{R}^n : fx > f_0\} \neq \emptyset$ .
- (ii) Given any linear program in  $n$  nonnegative variables suppose that there is an equation of the form  $\sum_{j=1}^n a_j^i x_j = b_i$  such that for some  $k \in \{1, \dots, n\}$  we have  $a_k^i > 0$  and  $a_j^i \leq 0$  for all  $j \neq k$ . Show that if  $b_i \geq 0$  then variable  $x_k$  can be eliminated and after substitution equation  $i$  can be dropped from the problem.
- (iii) With the same notation as before suppose  $a_k^i > 0$  and  $a_j^i \geq 0$  for all  $j \neq k$ . Show that  $x_k = 0$  in every feasible solution if  $b_i = 0$  and that the program has no feasible solution if  $b_i < 0$ .

**(i)** We want to show that  $\mathcal{X}^{\leq} = \mathcal{Y}^{\leq}$  if and only if  $\mathcal{X}^{\leq} \cap \{x \in \mathbb{R}^n : f\mathbf{x} > f_0\} = \emptyset$ , where  $f \in \mathbb{R}^n$  and  $f_0 \in \mathbb{R}$  are such that  $\mathcal{Y} \subseteq \{x \in \mathbb{R}^n : f\mathbf{x} \leq f_0\}$ . (Note that such a pair  $(f, f_0)$  exists because  $\mathcal{Y}^{\leq}$  is contained in the nonnegative orthant and thus, in particular,  $\mathcal{Y}^{\leq} \neq \mathbb{R}^n$ .) If  $\mathcal{X}^{\leq} = \mathcal{Y}^{\leq} \subseteq \{x \in \mathbb{R}^n : f\mathbf{x} \leq f_0\}$  then  $f\mathbf{x} \leq f_0$  for all  $x \in \mathcal{X}^{\leq}$  and thus  $\mathcal{X}^{\leq} \cap \{x \in \mathbb{R}^n : f\mathbf{x} > f_0\} = \emptyset$ . On the other hand let  $f \in \mathbb{R}^n$ ,  $f_0 \in \mathbb{R}$  be such that  $\mathcal{Y}^{\leq} \subseteq \{x \in \mathbb{R}^n : f\mathbf{x} \leq f_0\}$  and suppose that  $\mathcal{X}^{\leq} \cap \{x \in \mathbb{R}^n : f\mathbf{x} > f_0\} \neq \emptyset$ , i.e. there exists  $x \in \mathcal{X}^{\leq}$  such that  $f\mathbf{x} > f_0$ . Thus  $x \notin \mathcal{Y}^{\leq}$  and since  $\mathcal{Y}^{\leq} \subseteq \mathcal{X}^{\leq}$  it follows that  $\mathcal{Y}^{\leq} \subset \mathcal{X}^{\leq}$ , i.e.  $\mathcal{Y}^{\leq} \neq \mathcal{X}^{\leq}$ .

**(ii)** Solving for  $x_k$  we get  $x_k = \frac{b_i}{a_k^i} - \sum_{\substack{j=1 \\ j \neq k}}^n \frac{a_j^i}{a_k^i} x_j \geq 0$ , since  $b_i/a_k^i > 0$  and  $a_j^i/a_k^i < 0$ . Thus after

substituting the variable  $x_k$  wherever it appears, we can drop the equation from the formulation since it no longer affects the solution. Its numerical value is computed from the equation based on the values of the other variables.

**(iii)** If  $b_i = 0$  we have  $\sum_{j=1}^n a_j^i x_j = 0$ . Suppose that there exists a solution with  $x_k > 0$ . Then we have  $\sum_{j=1}^n a_j^i x_j \geq a_k^i x_k > 0$  where the first inequality follows from the fact that  $a_j^i \geq 0$  and  $x_j \geq 0$  for all  $j \neq k$  and the second from  $a_k^i > 0$  and the assumption that  $x_k > 0$ . Thus we get a contradiction.

If  $b_i < 0$ , then we have  $\sum_{j=1}^n a_j^i x_j \geq 0$  since  $a_j^i \geq 0$  and  $x_j \geq 0$  for all  $j$ . Thus the equality  $\sum_{j=1}^n a_j^i x_j = b_i < 0$  is violated by all feasible points and thus the linear program is infeasible.

---

### Exercise 6.7

Write a computer program of the dual simplex algorithm in a computer language of your choice for problems in canonical form satisfying  $c \leq 0$  and using the updating formulas like in Chapter 5 or a canned subroutine for inverting a nonsingular matrix.

---

The following program is an implementation of the algorithm as a MATLAB function. We pick the smallest index that satisfies (6.7) and (6.8) as  $\ell$  and  $j$  respectively.

```
% Dual simplex algorithm for LPs in canonical form
% max {cx: A~x <= b, x >= 0}
% where c <= 0.
%
% INPUT VARIABLES
% A,b,c          -> LP data
%
% RETURN VARIABLES
% sol_stat =  0 -> unbounded solution
%             -1 -> infeasible solution
%             1 -> finite optimal solution
% x              -> primal solution vector
% z              -> optimal objective function value
```

```

function [sol_stat,x,z] = dsimplex(A,b,c)
[m,n]=size(A);
A=[A'; eye(m)]';
c=[-c zeros(1,m)];
I=eye(m);
B=I;
cb=zeros(1,m);
status=[zeros(1,m+n)];
for i=1:m, p(i)=n+i; status(n+i)=i; end
iter = 0;
sol_stat=-2;
while ( sol_stat < 0),
    Binv=inv(B);
    cbar=c-cb*Binv*A;
    bbar=Binv*b';
    if (bbar >=0)
        fprintf('Optimal solution found in %d iterations.\n',iter);
        z=-cb*bbar;
        x=zeros(1,n);
        for i=1:m, x(p(i))=bbar(i); end
        sol_stat=1;
        return;
    end
    r=1;
    while (bbar(r) >=0), r=r+1; end
    l=p(r);
    ur=I(:,r);
    yr=ur'*Binv*A;
    if (yr >= 0),
        fprintf('Primal infeasibility.');
        sol_stat=-1;
        return
    end
    for i=1:n,
        if (yr(i) < 0 & status(i)==0)
            rat(i)=-cbar(i)/yr(i);
        end
    end
    [gamma,j]=min(rat);
    B=B+(A(:,j)-A(:,l))*ur';
    cb=cb+(c(j)-c(l))*ur';
    p(r)=j;
    status(j)=r;
    status(l)=0;
    iter=iter+1;
end

```

The input data for the problem of Exercise 6.8 (see below) – after converting the problem from canonical to standard form – are put as follows into a file called `dsdat.m`:

```
c=[-2 -3 -4 -2];
b=[ -10 -12];
A=[ -1 -1 -1 -1 ; -3 -1 -4 -2 ];
```

The following shows the function call from MATLAB and the output for the above data (assuming that they are in the file `dsdat.m`):

```
>> dsdat
>> [stat,x,z]=dsimplex(A,b,c)
Optimal solution found in 1 iterations.

stat =
    1
x =
    10      0      0      0      0      18
z =
   -20
```

---

### Exercise 6.8

Solve the following linear program by the dual simplex algorithm using your own precise choice rules for (6.7) and (6.8).

$$\begin{aligned} \max \quad & -2x_1 - 3x_2 - 4x_3 - 2x_4 \\ \text{subject to} \quad & -x_1 - x_2 - x_3 - x_4 \leq -10 \\ & -3x_1 - x_2 - 4x_3 - 2x_4 \leq -12 \\ & x_i \geq 0 \quad \text{for } i = 1, \dots, 4. \end{aligned}$$


---

First we bring the problem in the form required by the dual simplex algorithm, i.e. in the form  $\min\{\mathbf{c}\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  where

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 \\ 3 & 1 & 4 & 2 & 0 & -1 \end{pmatrix}, \quad \mathbf{c} = (2, 3, 4, 2, 0, 0), \quad \mathbf{b} = \begin{pmatrix} 10 \\ 12 \end{pmatrix}.$$

We use rules (c2\*) and (r2\*) for (6.7) and (6.8).

Starting with the basis  $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$  with  $I = \{5, 6\}$  which is dual feasible since  $\bar{\mathbf{c}} = \mathbf{c} \geq \mathbf{0}$  since  $\mathbf{c}_B = \mathbf{0}$ . Thus we have

$$\mathbf{B}^{-1} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \bar{\mathbf{c}} = (2, 3, 4, 2, 0, 0), \quad \bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 10 \\ 12 \end{pmatrix} = \begin{pmatrix} -10 \\ -12 \end{pmatrix}, \quad \ell = 5.$$

$$\mathbf{y}^r = \mathbf{u}_r^T \mathbf{B}^{-1} \mathbf{R} = (1, 0) \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & 4 & 2 \end{pmatrix} = (-1, -1, -1, -1), \gamma = \min\left\{\frac{2}{1}, \frac{3}{1}, \frac{4}{1}, \frac{2}{1}\right\} = 2, j = 1.$$

$$\mathbf{B} = \mathbf{B} + (\mathbf{a}_1 - \mathbf{a}_5) \mathbf{u}_r^T = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} + \begin{pmatrix} 2 \\ 3 \end{pmatrix} (1, 0) = \begin{pmatrix} 1 & 0 \\ 3 & -1 \end{pmatrix}, I = \{1, 6\}.$$

$$\mathbf{B}^{-1} = \begin{pmatrix} 1 & 0 \\ 3 & -1 \end{pmatrix}, \mathbf{c}_B = (0, 0) + (2 - 0)(1, 0) = (2, 0).$$

$$\bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} = (2, 3, 4, 2, 0, 0) - (2, 0) \begin{pmatrix} 1 & 0 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 \\ 3 & 1 & 4 & 2 & 0 & -1 \end{pmatrix} = (0, 1, 2, 0, 2, 0).$$

$$\bar{\mathbf{b}} = \mathbf{B}^{-1} \mathbf{b} = \begin{pmatrix} 1 & 0 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 10 \\ 12 \end{pmatrix} = \begin{pmatrix} 10 \\ 18 \end{pmatrix}.$$

Since  $\bar{\mathbf{b}} \geq 0$  the current solution is optimal. So the optimal solution is  $x_1 = 10, x_2 = x_3 = x_4 = 0$  and the optimal value is  $-2(10) = -20$ .

---

### Exercise 6.9

(i) Show that the linear program

$$(pLP) \quad \min\{\mathbf{c}\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

has an unbounded optimum if and only if there exists  $d \in \mathbb{R}^n$  such that  $\mathbf{Ad} = \mathbf{0}, d \geq \mathbf{0}$  and  $cd < 0$ .

(ii) Show that  $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  is bounded if and only if there exists a  $\mathbf{u} \in \mathbb{R}^m$  such that  $\mathbf{u}^T \mathbf{A} > \mathbf{0}$ .

---

**(i)** Suppose that the problem (pLP) is unbounded. Then the dual problem is infeasible, i.e.  $\{\mathbf{u} \in \mathbb{R}^m : \mathbf{A}^T \mathbf{u}^T \leq \mathbf{c}^T\} = \emptyset$ . By Exercise 6.5 (i) this happens if and only if there exists a vector  $d \in \mathbb{R}^n$  such that  $d \geq \mathbf{0}, d^T \mathbf{A}^T = \mathbf{0}$  and  $d^T \mathbf{c}^T < 0$ .

On the other hand, suppose that there exists  $d \in \mathbb{R}^n$  such that  $\mathbf{Ad} = \mathbf{0}, d \geq \mathbf{0}$  and  $cd < 0$  and that (pLP) has an optimal solution  $\mathbf{x}^*$  with optimal value  $z^*$ . Then we have  $\mathbf{A}(\mathbf{x}^* + d) = \mathbf{Ax}^* + \mathbf{Ad} = \mathbf{b}$  and  $\mathbf{x}^* + d \geq \mathbf{0}$ , i.e.  $\mathbf{x}^* + d$  is a feasible solution to (pLP). But  $c(\mathbf{x}^* + d) = c\mathbf{x}^* + cd < z^*$  since  $cd < 0$ , which contradicts the optimality of  $\mathbf{x}^*$ .

**(ii)** Suppose that  $\mathcal{X}$  is bounded. Then the linear program  $\max\{\mathbf{e}\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  has an optimal solution and so does its dual. In particular, the dual problem is feasible, i.e.  $\{\mathbf{u} \in \mathbb{R}^m : \mathbf{u}\mathbf{A} \geq \mathbf{e}\} \neq \emptyset$ . Thus there exists  $\mathbf{u} \in \mathbb{R}^m$  such that  $\mathbf{u}\mathbf{A} \geq \mathbf{e} > \mathbf{0}$ .

On the other hand if there exists  $\mathbf{u} \in \mathbb{R}^m$  such that  $\mathbf{u}\mathbf{A} > \mathbf{0}$  then by Exercise 6.5 (iv) we have  $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \neq \mathbf{0}\} = \emptyset$ . It follows that there exists no vector  $d \geq \mathbf{0}$  such that  $\mathbf{Ad} = \mathbf{0}$  and  $cd < 0$  and thus by part (i) (pLP) has a bounded optimal solution for all vectors  $\mathbf{c}$ . Selecting

$\mathbf{c} = \mathbf{u}^i$ , the  $i$ -th unit vector, we conclude that all components of  $\mathbf{x}$  are bounded and thus  $\mathcal{X}$  is bounded.

---

### Exercise 6.10

Consider the problem with a parametric objective function

$$(LP_\mu) \quad z(\mu) = \min\{(\mathbf{c} + \mu\mathbf{d})\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

where  $\mathbf{d} \in \mathbb{R}^n$  is a row vector of changes to the vector  $\mathbf{c}$  and  $\mu$  is a parameter. What can you prove about the function  $z(\mu)$ ? Give a procedure that finds  $z(\mu)$  for all  $0 \leq \mu < +\infty$  when  $z(0)$  is finite.

---

If  $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} = \emptyset$ , then  $z(\mu) = +\infty$  for all  $\mu \in \mathbb{R}$ . Suppose that  $\mathcal{X} \neq \emptyset$  and that there exist  $\mu_1 < \mu_2$  such that  $z(\mu_1) > -\infty$  and  $z(\mu_2) > -\infty$ .

It follows that  $z(\mu_i) = \max\{\mathbf{u}\mathbf{A} : \mathbf{u}\mathbf{A} \leq \mathbf{c} + \mu_i\mathbf{d}\}$  for  $i = 1, 2$ . Let  $\mathbf{u}^1, \mathbf{u}^2$  be the corresponding optimal dual solutions. Then  $\mathbf{u}(\alpha) = \alpha\mathbf{u}^1 + (1 - \alpha)\mathbf{u}^2$  is a feasible solution to  $\mathbf{u}\mathbf{A} \leq \mathbf{c} + \mu(\alpha)\mathbf{d}$  where  $\mu(\alpha) = \alpha\mu_1 + (1 - \alpha)\mu_2$  with objective function value  $\alpha z(\mu_1) + (1 - \alpha)z(\mu_2)$ . Hence  $(LP_{\mu(\alpha)})$  has a finite optimum since  $\mathcal{X} \neq \emptyset$  and  $\alpha z(\mu_1) + (1 - \alpha)z(\mu_2) \leq z(\mu(\alpha)) = z(\alpha\mu_1 + (1 - \alpha)\mu_2)$  for all  $0 \leq \alpha \leq 1$  since we are maximizing in the dual problem. This shows that  $z(\mu)$  is a concave function whenever  $z(\mu)$  is finite. Moreover, it follows that the domain of finiteness of  $z(\mu)$  is some interval of the real line.

Suppose now  $z(0) > -\infty$  and  $\mathcal{X} \neq \emptyset$ . Let  $B$  be an optimal basis for  $(LP_0)$ . Then we calculate

$$\overline{\mathbf{c} + \mu\mathbf{d}} = \mathbf{c} + \mu\mathbf{d} - (\mathbf{c}_B + \mu\mathbf{d}_B)\mathbf{B}^{-1}\mathbf{A} = \mathbf{c} - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A} + \mu(\mathbf{d} - \mathbf{d}_B\mathbf{B}^{-1}\mathbf{A}) = \bar{\mathbf{c}} + \mu\bar{\mathbf{d}}.$$

Since  $B$  is optimal for  $\mu = 0$ , we have  $\bar{\mathbf{c}} \geq \mathbf{0}$ . Consequently, if  $\bar{\mathbf{d}} \geq \mathbf{0}$  then  $z(\mu) = z(0)$  for all  $\mu \geq 0$ , because  $B$  remains optimal for all  $\mu \geq 0$ . So suppose that  $\bar{\mathbf{d}} \not\geq \mathbf{0}$  and let

$$\mu_0 = \min\left\{\frac{\bar{c}_j}{\bar{d}_j} : \bar{d}_j < 0\right\}.$$

Then  $B$  remains optimal for  $0 \leq \mu \leq \mu_0$  and thus  $z(\mu) = z(0)$  for  $0 \leq \mu \leq \mu_0$ . Let  $j \in \{1, \dots, n\}$  be such that  $\mu_0 = \bar{c}_j/\bar{d}_j$ . Since the variable  $x_j$  is necessarily nonbasic and its reduced cost in the linear program  $(LP_{\mu_0})$  equals zero, we can pivot  $x_j$  into the basic set without changing the objective function. If the transformed column  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j \leq \mathbf{0}$ , then for all  $\mu > \mu_0$  we have  $z(\mu) = -\infty$  and the procedure stops. Otherwise, pivoting variable  $x_j$  into the basis will (typically) change the solution vector, i.e., produce an alternative optimum solution to  $(LP_{\mu_0})$ .

To systematically find a new basis displaying the optimality of an alternative optimum we increment  $\mu_0$  by a “small enough”  $\epsilon_0 > 0$ . A theoretical estimation of  $\epsilon_0$  is possible using the material of Chapter 7, but in practice one simply uses  $\epsilon_0 = 10^{-2}$  or  $10^{-3}$  or smaller depending upon the numerical accuracy of the LP solver that is utilized. Now the “current” solution is no longer optimal and we can use the primal simplex algorithm to reoptimize the problem. If unboundedness is detected the procedure stops; otherwise we find a new basis  $B_\epsilon$  different from  $B$  that displays optimality of a new solution  $\mathbf{x}^*$  for all  $0 \leq \epsilon \leq \epsilon_0$ . From the reduced cost of  $(LP_{\mu_0})$  with respect to  $B_\epsilon$  we compute like above  $\mu_1 > \mu_0$  and iterate. (Note that in the computation of  $\mu_1$  the reduced cost  $\bar{c}_j$  is thus replaced by the reduced cost  $\bar{c}_j + \mu_0\bar{d}_j$  in the new basis.) Moreover,

we calculate  $z(\mu_0 + \epsilon) = z(\mu_0) + \epsilon d\mathbf{x}^*$  for  $0 \leq \epsilon \leq \mu_1 - \mu_0$ . Hence  $z(\mu)$  is a piecewise linear concave function of  $\mu$  wherever  $z(\mu)$  is finite. Since there are only finitely many bases and no basis is repeated the procedure is finite.

---

### Exercise 6.11

*Assume that the first constraint of a linear program ( $LP_H$ ) is the constraint (6.9) where  $M$  is a suitably chosen “large” number, an input parameter if you wish. Write a program in a computer language of your choice for the dynamic simplex algorithm utilizing the programs of Exercises 5.2 and 6.7 as subroutines.*

---

The following program is an implementation of the algorithm as a MATLAB function. The program is stored in the file `dynsmplx.m`.

```
%%%%%%%%
% Dynamic simplex algorithm for LPs in canonical form
% min {cx: A~x <= b, x >= 0}
% INPUT VARIABLES
% A,b,c          -> LP data
% RETURN VARIABLES
% sol_stat = 0 -> unbounded, -1 -> infeasible, 1 -> finite
% x              -> primal solution vector
% z              -> optimal objective function value
%%%%%%%
function [sol_stat,x,z] = dynsmplx(A,b,c)
[m,n]=size(A);
zlow=-inf;
l=[1 1 1 0 0 0];
p=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0];
sol_stat=-2;
qcnt=1; kcnt=1;
while ( qcnt ~= 0 | kcnt ~= 0 ),
    while (qcnt ~= 0),
        [ALP,AL,AP,bL,cP]=getmatrix(A,b,c,p,l);
        [status,x,z,u,s]=psimplex(ALP,bL,cP);
        fprintf('Reduced problem solved.')
        qcnt=0;
        for j=1:n,
            if (p(j)==0 & c(j)-u*AL(:,j) > 0)
                p(j)=1; qcnt=qcnt+1;
            end
        end
        fprintf(' Adding %2g columns.\n',qcnt)
    end
end
```

```

if (z > zlow)
    zlow=z
k=0;
for i=1:m,
    if (l(i) == 1)
        k=k+1;
    if ( s(k) > 0), l(i)=0; end
end
end
kcnt=0;
for i=1:m,
    if (l(i)==0 & AP(i,:)*x' - b(i) > 0)
        l(i)=1; kcnt=kcnt+1;
    end
end
if (kcnt == 0)
    fprintf('Optimal solution found.\n');
    return;
end
fprintf('Adding %2g rows.\n',kcnt)
[ALP,AL,AP,bL,cP]=getmatrix(A,b,c,p,l);
[status,x,z,u,s]=d simplex(ALP,bL,cP);
end

```

Note that in order to find a dual feasible starting basis for the dual simplex, we have to implement the technique with the big  $M$  described in the text, and that's why in the statement of the exercise the assumption regarding the first constraint is made. Here is an implementation of the "trick".

```

function [A, b, c, B, Binv, cb, q, status] = dbinit(A,b,c)
[m,n]=size(A);
A=[A~zeros(m,1) eye(m)];
A=[ones(1,n+m+1); A];
[m,n]=size(A);
c=[-c zeros(1,m)];
b=[10^3 b]; B=1; Binv=1;
[aux,j]=min(c); p=zeros(1,n);
q(1)=j; p(j)=1;
status=zeros(1,n);
r(1)=1; qcmt=1; rcmt=1; status(j)=1;
k=1;
for i=1:n,
    if p(i) == 0,
        cbar(i)=c(i)-c(j)*A(1,i);
    end
end

```

```

while k < m,
    a=A(k+1,:);
    for i=1:n,
        if p(i) == 0,
            abar(i)=a(i)-a(q)*Binv*A(r,i);
        end
    end
    if (abar == 0),
        if (b(k+1) == a(q)*Binv*b(r)),
            k=k+1;
            fprintf('Skipping row %2g.\n',k-1)
        else
            error('There is no feasible dual basis!\n')
            return
        end
    else
        minr=10^30;
        for i=1:n,
            if (p(i) == 0 & abar(i) ~= 0 & cbar(i)/abs(abar(i)) < minr),
                l=i; minr=cbar(i)/abs(abar(i));
            end
        end
        qcmt=qcmt+1; q(qcmt)=l; p(l)=1; status(l)=qcmt;
        rcmt=rcmt+1; r(rcmt)=k+1;
        B=A(r,q); Binv=inv(B); k=k+1;
        for i=1:n,
            if p(i) == 0,
                cbar(i)=c(i)-c(q)*Binv*A(r,i);
            end
        end
    end
end
cb=c(q);

```

We test the routine in a linear program with the following input data

```

c=[3 47 43 73 86 36 96 47 36 61 46 98 63 71 62 33 26 16 80 45 60];
A=[97 74 24 67 62 42 81 14 57 20 42 53 32 37 32 27 7 36 7 51 24;
16 76 62 27 66 56 50 26 71 7 32 90 79 78 53 13 55 38 58 59 88;
12 56 85 99 26 96 96 68 27 31 5 3 72 93 15 57 12 10 14 21 88;
55 59 56 35 64 38 54 82 46 22 31 62 43 9 90 6 18 44 32 53 23;
16 22 77 94 39 49 54 43 54 82 17 37 93 23 78 87 35 20 96 43 84;
84 42 17 53 31 57 24 55 6 88 77 4 74 47 67 21 76 33 15 25 83];
b=[400 400 350 320 420 400];

```

The output of the dynamic simplex routine follows:

```
>>p21rwdat
>>[stat,x,z]=dynsmplx(A,b,c);
Optimum found in 7 iterations.
Reduced problem solved. Adding 15 columns.
Optimum found in 16 iterations.
Reduced problem solved. Adding 0 columns.
Adding 3 rows.
Optimal solution found in 12 iterations.
Optimal solution found.

>> x
x =
    Columns 1 through 7
        0         0         0         0         0         0    1.7715
    Columns 8 through 14
        0         0    2.5410    1.0426    2.0553         0    0.9654
    Columns 15 through 21
        0         0         0         0         0         0         0

>>z
z =
  642.9870
>>
```

---

### \*Exercise 6.12

Like in the primal simplex algorithm normed pivot row selection rules are frequently used in Step (6.7) of the dual simplex algorithm (see also Chapter 5). Consider the two different row norms called the “full” and “partial” row norms, respectively, given by

$$d_i^2 = 1 + \|\mathbf{u}_i^T \mathbf{B}^{-1}\|^2 + \|\mathbf{u}_i^T \mathbf{B}^{-1} \mathbf{R}\|^2, \quad \delta_i^2 = \|\mathbf{u}_i^T \mathbf{B}^{-1}\|^2,$$

where  $\mathbf{u}_i \in \mathbb{R}^m$  is the  $i$ -th unit vector. Let  $\mathbf{B}_{new}$  be a new basis obtained from  $\mathbf{B}$  by pivoting in column  $j$  and row  $r$ . Show that

$$(d_i^{new})^2 = d_i^2 + \Theta_i^2 d_r^2 - 2\Theta_i (\mathbf{b}^i(\mathbf{b}^r)^T + \mathbf{y}^i(\mathbf{y}^r)^T) \text{ for } i \neq r, \quad (d_r^{new})^2 = (y_j^r)^{-2} d_r^2,$$

$$(\delta_i^{new})^2 = \delta_i^2 + \Theta_i^2 \delta_r^2 - 2\Theta_i \mathbf{b}^i(\mathbf{b}^r)^T \text{ for } i \neq r, \quad (\delta_r^{new})^2 = (y_j^r)^{-2} \delta_r^2,$$

where  $\mathbf{b}^i = \mathbf{u}_i^T \mathbf{B}^{-1}$ ,  $\mathbf{y}^i = \mathbf{b}^i \mathbf{R}$ ,  $y_j^i = \mathbf{b}^i a_j$  and  $\Theta_i = y_j^i/y_j^r$ . Discuss various ways of using these relations to carry out an update of the “new” norms from the “old” norms. (See Exercise 7.17 for a geometric interpretation of the normed pivot row selection criteria.)

---

From formula (4.5) of Chapter 4 we have

$$\mathbf{B}_{new}^{-1} = \mathbf{B}^{-1} - \frac{1}{y_j^r} (\mathbf{y}_j - \mathbf{u}_r) \mathbf{u}_r^T \mathbf{B}^{-1}$$

where  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$  and  $y_j^r$  is the pivot element (see also Chapter 5). Denoting  $(\mathbf{B}^{-1})^T$  simply by  $\mathbf{B}^{-T}$  we calculate for  $i \neq r$

$$(\delta_i^{new})^2 = \|\mathbf{u}_i^T \mathbf{B}_{new}^{-1}\|^2 = \|(\mathbf{u}_i^T - \Theta_i \mathbf{u}_r^T) \mathbf{B}^{-1}\|^2 = \delta_i^2 - 2\Theta_i \mathbf{u}_i^T \mathbf{B}^{-1} \mathbf{B}^{-T} \mathbf{u}_r + \Theta_i^2 \|\mathbf{u}_r^T \mathbf{B}^{-1}\|^2$$

and thus the formula follows. For  $i = r$  we get

$$(\delta_r^{new})^2 = \|\mathbf{u}_r^T \mathbf{B}_{new}^{-1}\|^2 = \|\frac{1}{y_j^r} \mathbf{u}_r^T \mathbf{B}^{-1}\|^2 = (y_j^r)^{-2} \delta_r^2$$

and thus the assertion is correct. To calculate  $(d_i^{new})^2$  we note that  $d_i^2 = 1 + \delta_i^2 + \|\mathbf{y}^i\|^2$  and that  $\mathbf{R}_{new} = \mathbf{R} + (\mathbf{a}_\ell - \mathbf{a}_j)\mathbf{u}_p^T$ , where  $\mathbf{u}_p \in \mathbb{R}^{n-m}$  is a unit vector with an entry +1 in the position  $p$  of column  $\mathbf{a}_j$  in the matrix  $\mathbf{R}$ . Moreover, we calculate for  $i \neq r$

$$\|(\mathbf{u}_i^T - \Theta_i \mathbf{u}_r^T) \mathbf{B}^{-1} \mathbf{R}_{new}\|^2 = \|(\mathbf{u}_i^T - \Theta_i \mathbf{u}_r^T) \mathbf{B}^{-1} \mathbf{R}\|^2 + \Theta_i^2$$

because  $(\mathbf{u}_i^T - \Theta_i \mathbf{u}_r^T) \mathbf{B}^{-1} \mathbf{a}_j = 0$ . We calculate for  $i \neq r$  using the formula for  $\delta_i^{new}$

$$\begin{aligned} (d_i^{new})^2 &= 1 + \|\mathbf{u}_i^T \mathbf{B}_{new}^{-1}\|^2 + \|\mathbf{u}_i^T \mathbf{B}_{new}^{-1} \mathbf{R}_{new}\|^2 \\ &= 1 + \delta_i^2 - 2\Theta_i \mathbf{b}^i (\mathbf{b}^r)^T + \Theta_i^2 \delta_r^2 + \|(\mathbf{u}_i^T - \Theta_i \mathbf{u}_r^T) \mathbf{B}^{-1} \mathbf{R}\|^2 + \Theta_i^2 \\ &= 1 + \delta_i^2 + \|\mathbf{u}_i^T \mathbf{B}^{-1} \mathbf{R}\|^2 + \Theta_i^2 (1 + \delta_r^2 + \|\mathbf{u}_r^T \mathbf{B}^{-1} \mathbf{R}\|^2) - 2\Theta_i (\mathbf{b}^i (\mathbf{b}^r)^T + \mathbf{y}^i (\mathbf{y}^r)^T) \end{aligned}$$

as claimed and the calculation of  $(d_r^{new})^2$  goes likewise. The calculation of  $\delta_r^{new}$  and  $d_r^{new}$  from  $\delta_r$  and  $d_r$ , respectively, is trivial. So let us assume that  $i \neq r$ . The calculation of  $\delta_i^{new}$  from the updating formula requires the calculation of  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$  and the inner products  $\mathbf{b}^i (\mathbf{b}^r)^T$ . To avoid taking the inner products  $\mathbf{b}^i (\mathbf{b}^r)^T$  we can proceed as follows. Since  $\mathbf{b}^i (\mathbf{b}^r)^T = \mathbf{u}_i^T \mathbf{B}^{-1} (\mathbf{b}^r)^T$  we can first determine the solution vector to  $\mathbf{Bw} = (\mathbf{b}^r)^T$  as we need to compute  $\mathbf{b}^r = \mathbf{u}_r^T \mathbf{B}^{-1}$  anyway for an efficient calculation of  $\mathbf{y}^r$ . Then we get  $(\delta_i^{new})^2 = \delta_i^2 + \Theta_i^2 \delta_r^2 - 2\Theta_i w_i$  as the update for the partial row norm  $\delta_i$  where  $w_i$  is the  $i$ -th component of  $\mathbf{w}$ . The update of the full row norm  $d_i$  is computationally more expensive as we need here the inner products  $\mathbf{y}^i (\mathbf{y}^r)^T$  as well. Since  $\mathbf{b}^i (\mathbf{b}^r)^T + \mathbf{y}^i (\mathbf{y}^r)^T = \mathbf{u}_i^T \mathbf{B}^{-1} ((\mathbf{b}^r)^T + \mathbf{R}(\mathbf{y}^r)^T)$  we can first calculate the vector  $\mathbf{z} = (\mathbf{b}^r)^T + \sum_{j \in J} y_j^r \mathbf{a}_j$ , which requires a pass through all nonbasic columns with  $y_j^r \neq 0$ . Then we determine the vector  $\mathbf{v} \in \mathbb{R}^m$  by solving the system of equations  $\mathbf{Bv} = \mathbf{z}$ . The updating formula for  $(d_i^{new})^2$  becomes  $(d_i^{new})^2 = d_i^2 + \Theta_i^2 d_r^2 - 2\Theta_i v_i$  where  $v_i$  is the  $i$ -th component of  $\mathbf{v}$  and  $i \neq r$ .

### \*Exercise 6.13

Consider the linear program in  $ns \geq 1$  variables  $\xi$

$$\max\{\hat{c}\xi : A_1\xi \leq b_1, A_2\xi \geq b_2, A_3\xi = b_3, \ell \leq \xi \leq u\},$$

where  $-\infty \leq \ell_j \leq u_j \leq +\infty$  for  $1 \leq j \leq ns$  like in Exercise 5.12, and bring the problem into the standard form (BVLP) for a linear program in bounded variables like done there. Given a basis  $B$  of  $A$  with index set  $I$  and a partitioning of  $J = N - I$  into  $G, L, U$  we call  $(I, G, L, U)$  a **dual basis** if

$$c_G - c_B B^{-1} R_G = 0_G, c_L - c_B B^{-1} R_L \leq 0_L, c_U - c_B B^{-1} R_U \geq 0_U.$$

- (i) State a dual simplex algorithm for (BVLP) and prove its correctness.  
(ii) Give a procedure that lets you find a dual basis for (BVLP) or conclude that none exists.
- 

**(i)** The dual simplex algorithm for (BVLP) goes as follows.

**Dual BVSimplex Algorithm** ( $m, n, \ell, u, A, b, c$ )

**Step 0:** Find a dual feasible basis  $B$  cum partitioning  $(G, L, U)$  of the nonbasic variables. Let  $I$  be the index set of  $B$ ,  $c_B = (c_j)_{j \in I}$  and initialize  $p_k$  for all  $k \in I$ .  
**if** none exists **then stop** “(BVLP) is either infeasible or unbounded”.

**Step 1:** Compute  $B^{-1}$ ,  $\bar{c} := c - c_B B^{-1} A$  and  $\bar{b} = B^{-1}(b - R_L \ell_L - R_U u_U)$ .

**if**  $\ell_B \leq x_B \leq u_B$  **then**

set  $x_B := \bar{b}$ ;  $x_G := 0$ ,  $x_L = \ell_L$ ,  $x_U = u_U$ , and

**stop** “ $x$  is an optimal feasible solution to (BVLP)”.

**else**

(1) choose  $\ell \in I$  such that  $\bar{b}_{p_\ell} < \ell_\ell$  or  $\bar{b}_{p_\ell} > u_\ell$  and set  $r := p_\ell$ .

**endif.**

**Step 2:** Compute  $y^r := u_r^T B^{-1} A$ .

**if**  $\bar{b}_r < \ell_\ell$  **then** compute the least ratio

$$\lambda = \min\{\min\{\bar{c}_k/y_k^r : y_k^r < 0, k \in G \cup L\}, \min\{\bar{c}_k/y_k^r : y_k^r > 0, k \in G \cup U\}\}.$$

**else** compute the least ratio

$$\lambda = \max\{\max\{\bar{c}_k/y_k^r : y_k^r > 0, k \in G \cup L\}, \max\{\bar{c}_k/y_k^r : y_k^r < 0, k \in G \cup U\}\}.$$

**endif.**

**if**  $\lambda$  is undefined **then stop** “BVLP has no feasible solution”.

(2) Let  $j \in G \cup L \cup U$  be any index for which the least ratio  $\lambda$  is attained.

Set  $\theta = (\bar{b}_r - \ell_\ell)/y_j^r$  if  $\bar{b}_r < \ell_\ell$ ,  $\theta = (\bar{b}_r - u_\ell)/y_j^r$  otherwise.

**Step 3:** Set  $B := B + (a_j - a_\ell)u_r^T$ ,  $c_B := c_B + (c_j - c_\ell)u_r^T$ ,  $I := I - \ell \cup \{j\}$ ,  $p_j := r$ ,  $p_\ell := 0$

and update  $(G, L, U)$  according to Table 5.1 of Exercise 5.12(iii) using  $\ell \in I \cap (D \cup C) \downarrow$   
if  $\bar{b}_r < \ell_\ell$  and  $\ell \in I \cap (H \cup C) \uparrow$  if  $\bar{b}_r > u_\ell$ ; **go to Step 1**.

To prove the correctness of the Dual BVSimplex Algorithm let  $B$  cum the partitioning  $(G, L, U)$  be a dual basis. If  $\ell_B \leq x_B \leq u_B$  then by Exercise 5.12(ii) the algorithm terminates correctly. Otherwise, if the least ratio  $\lambda$  in Step 2 is defined, the updating formulas of Exercise 5.12(iv) apply and by construction,  $B_{new}$  cum the new partition  $(G^{new}, L^{new}, U^{new})$  is a dual basis (see also below). Moreover, we find

$$z_{B_{new}}(G^{new}, L^{new}, U^{new}) = z_B(G, L, U) + \bar{c}_j \theta \text{ with } \bar{c}_j \theta \leq 0$$

since  $\bar{c}_j \theta = \lambda(\bar{b}_r - \ell_\ell) \leq 0$  if  $\bar{b}_r < \ell_\ell$  and  $\bar{c}_j \theta = \lambda(\bar{b}_r - u_\ell) \leq 0$  if  $\bar{b}_r > u_\ell$ . Consequently, as long as the algorithm repeats no basis, we have finite convergence.

To ensure that no basis is repeated e.g. a least-index rule in the pivot row selection (1) and the pivot column selection (2) can be used to break ties. Note that if  $j \in G$  is selected in (2) then necessarily  $\bar{c}_j = 0$ , but since free variables are never pivoted out of the basis, a basis repetition cannot occur. Thus the least-index rule needs to be applied only to the lower-bounded, upper-bounded, and bounded variables of (BVLP).

Suppose now that the least ratio  $\lambda$  is not defined in Step 2. Then necessarily  $y_k^r = 0$  for all  $k \in G$  or  $G = \emptyset$ , i.e.,  $y_G^r = \mathbf{0}_G$ . Using the conventions  $-\infty \cdot 0 = +\infty \cdot 0 = 0$  let

$$\min\{\mathbf{v}\mathbf{b} + \mathbf{w}\mathbf{u} - \boldsymbol{\eta}\ell : \mathbf{v}\mathbf{A} + \mathbf{w} - \boldsymbol{\eta} = \mathbf{c}, \mathbf{w} \geq \mathbf{0}, \boldsymbol{\eta} \geq \mathbf{0}\}$$

be the dual of (BVLP) and assume WROG that the basis  $B$  occurs in the  $m$  **first** columns of  $A$ , i.e., that  $I = \{1, \dots, m\}$  and thus  $\ell = r$ . Let  $\mathbf{y}_U^r = (y_j^r)_{j \in U}$ , etc. If  $\bar{b}_r < \ell_r$  we define for  $\lambda \geq 0$ ,

$$\begin{aligned} \mathbf{v}(\lambda) &= \mathbf{c}_B B^{-1} + \lambda \mathbf{u}_r^T B^{-1}, \mathbf{w}_I(\lambda) = \mathbf{0}_I, \mathbf{w}_G(\lambda) = \mathbf{0}_G, \mathbf{w}_L(\lambda) = \mathbf{0}_L, \mathbf{w}_U(\lambda) = \mathbf{c}_U - \mathbf{c}_B B^{-1} \mathbf{R}_U - \lambda \mathbf{y}_U^r, \\ \boldsymbol{\eta}_G(\lambda) &= \mathbf{0}_G, \boldsymbol{\eta}_L(\lambda) = -\mathbf{c}_L + \mathbf{c}_B B^{-1} \mathbf{R}_L + \lambda \mathbf{y}_L^r, \boldsymbol{\eta}_U(\lambda) = \mathbf{0}_U, \eta_r(\lambda) = \lambda, \eta_i(\lambda) = 0 \text{ for } i \in I - r. \end{aligned}$$

Since the minimum ratio test for  $\lambda$  fails,  $(\mathbf{v}(\lambda), \mathbf{w}(\lambda), \boldsymbol{\eta}(\lambda))$  is feasible to the dual for all  $\lambda \geq 0$  and

$$\mathbf{v}(\lambda)\mathbf{b} + \mathbf{w}(\lambda)\mathbf{u} - \boldsymbol{\eta}(\lambda)\ell = z_B(G, L, U) + \lambda(\bar{b}_r - \ell_r) \rightarrow -\infty \text{ for } \lambda \rightarrow +\infty.$$

Consequently, the dual is unbounded and thus a feasible solution to (BVLP) does not exist. If  $\bar{b}_r > u_\ell$  we define  $(\mathbf{v}(\lambda), \mathbf{w}(\lambda), \boldsymbol{\eta}(\lambda))$  for  $\lambda \leq 0$  as before except that

$$\boldsymbol{\eta}_I(\lambda) = \mathbf{0}_I, w_r(\lambda) = -\lambda, w_i(\lambda) = 0 \text{ for } i \in I - r.$$

Since the minimum ratio test for  $\lambda$  fails,  $(\mathbf{v}(\lambda), \mathbf{w}(\lambda), \boldsymbol{\eta}(\lambda))$  is feasible to the dual for all  $\lambda \leq 0$  and the objective function satisfies

$$\mathbf{v}(\lambda)\mathbf{b} + \mathbf{w}(\lambda)\mathbf{u} - \boldsymbol{\eta}(\lambda)\ell = z_B(G, L, U) + \lambda(\bar{b}_r - u_\ell) \rightarrow -\infty \text{ for } \lambda \rightarrow -\infty.$$

Thus the algorithm is correct. The efficient organization of the calculations of the Dual BVSimplex Algorithm is left to the reader.

**(ii)** If all structural variables of (BVLP) are in  $C$ , i.e., if they all have finite upper and lower bounds, then it is straightforward how to start the Dual BVSimplex Algorithm: we choose  $I = \{n+1, \dots, n+m\}$ ,  $B$  is the (signed) unit matrix and  $\mathbf{c}_B = \mathbf{0}_I$ . Consequently, the reduced profit  $\bar{c}_j = c_j$  for all  $j \in NS$ , we initialize the partitioning of the nonbasic variables by  $L = \{j \in NS : c_j \leq 0\}$ ,  $U = \{j \in NS : c_j > 0\}$  and we run the algorithm. In the general case – if a dual basis is not at hand – we use a two-phase procedure. Let  $Z = \{n+i : \text{row } i \text{ is an equation}\}$ ,  $D_V = D \cap (N - Z)$ ,  $I = \{n+1, \dots, n+m\}$ , and consider the Phase I problem

$$\begin{aligned} \max\{\mathbf{c}_F \mathbf{x}_F + \mathbf{c}_{D_V} \mathbf{x}_{D_V} + \mathbf{c}_H \mathbf{x}_H : \mathbf{A}_F \mathbf{x}_F + \mathbf{A}_{D_V} \mathbf{x}_{D_V} + \mathbf{A}_H \mathbf{x}_H + \mathbf{A}_Z \mathbf{x}_Z = \mathbf{0}, \\ \mathbf{0}_{D_V} \leq \mathbf{x}_{D_V} \leq \mathbf{e}_{D_V}, -\mathbf{e}_H \leq \mathbf{x}_H \leq \mathbf{0}_H, \mathbf{x}_Z \geq \mathbf{0}_Z, \mathbf{x}_Z \leq \mathbf{0}_Z\}. \end{aligned} \quad (\text{Phase I})$$

The structural variables  $C \cap NS$  are missing from the Phase I problem, the artificial variables are required to be zero and all but the free structural variables are bounded variables. Thus if  $F = \emptyset$  we are in the case where all variables of (Phase I) are bounded and we know how to start the Phase I calculation for the Dual BVSimplex Algorithm (just choose  $I = \{n+1, \dots, n+m\}$ , let  $B$  be the corresponding basis and  $\mathbf{c}_B = \mathbf{0}_I$ , etc.). If  $F \neq \emptyset$  we also let initially  $I = \{n+1, \dots, n+m\}$ ,  $B$  be the corresponding basis and  $\mathbf{c}_B = \mathbf{0}_I$ . We set  $P = \emptyset$  and carry out the following pivoting procedure where  $k(i) \in I$  is the index of the variable corresponding to row  $i$  of  $B^{-1}$  (or column  $i$  of  $B$ ).

**Procedure** FREEVARIABLES **for** (BVLP)**while** ( $F \neq P$ ) **do**choose  $j \in F - P$  and compute  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$ .**if**  $y_j^i = 0$  for all  $k(i) \in I - F$  **then**compute  $\bar{c}_j = c_j - \mathbf{c}_B \mathbf{y}_j$ .**if**  $\bar{c}_j = 0$  **then** $P := P \cup \{j\}$ **else****stop** "BVLP is either infeasible or unbounded".**endif.****else**Let  $y_j^r \neq 0$  for some  $k(r) \in I - F$  and  $\ell = k(r)$ . Replace  $P := P \cup \{j\}$ ,  $I = I - \ell \cup \{j\}$ , $B := B + (\mathbf{a}_j - \mathbf{a}_\ell)\mathbf{u}_r^T$ ,  $\mathbf{c}_B := \mathbf{c}_B + (c_j - c_\ell)\mathbf{u}_r^T$ .**endif.**

The pivoting procedure iterates at most  $|F|$  times and thus it is finite. We claim that if the procedure terminates with  $P = F$  then we have a basis  $B$  such that all nonbasic free variables satisfy  $\bar{c}_j = 0$ , where  $\bar{c}_j = c_j - \mathbf{c}_B \mathbf{B}^{-1}\mathbf{a}_j$ . Suppose to the contrary that  $\bar{c}_j \neq 0$  for some  $j \in F$ . Then variable  $x_j$  is nonbasic, let  $B$  denote the basis when  $j$  enters the set  $P$  and  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$ . By the check in the third line of the procedure,  $y_j^i = 0$  for all  $k(i) \in I - F$  where  $I$  is the index set of  $B$ , i.e.,  $\mathbf{a}_j$  is a linear combination of  $\mathbf{a}_k$  with  $k \in I \cap F$ , and the corresponding  $\bar{c}_j = 0$ . It follows that  $\mathbf{a}_j = \sum_{k \in I \cap F} \lambda_k \mathbf{a}_k$  and  $c_j = \sum_{k \in I \cap F} \lambda_k c_k$  for some  $\lambda_k \in \mathbb{R}$ . Since none of the variables in  $I \cap F$  leaves the basis,  $\bar{c}_j \neq 0$  is impossible for the final basis produced by FREEVARIABLES and the claim follows.

Suppose that the pivoting procedure stops with the message that (BVLP) is either infeasible or unbounded. Like in the previous case, denote by  $B$  the basis when this happens, by  $I$  the index set of  $B$  and let  $\mathbf{y}_j = \mathbf{B}^{-1}\mathbf{a}_j$ . Again  $\mathbf{a}_j$  is a linear combination of  $\mathbf{a}_k$  with  $k \in I \cap F$ , but  $\bar{c}_j \neq 0$ . Define  $\mathbf{y} \in \mathbb{R}^n$  by

$$y_k = -y_{p_k}^i \text{ for } k \in I, y_j = 1, y_k = 0 \text{ otherwise.}$$

Then  $A\mathbf{y} = \mathbf{0}$  and  $\mathbf{c}\mathbf{y} = c_j - \mathbf{c}_B y_j = \bar{c}_j$ . Suppose there exists a feasible  $\mathbf{x} \in \mathbb{R}^n$  for (BVLP). Then  $\mathbf{x} + \lambda \mathbf{y}$  is a feasible solution to (BVLP) for all  $\lambda \in \mathbb{R}$ . Hence

$$\mathbf{c}(\mathbf{x} + \lambda \mathbf{y}) = \mathbf{c}\mathbf{x} + \lambda \bar{c}_j \rightarrow +\infty \text{ for } \lambda \rightarrow +\infty \text{ if } \bar{c}_j > 0 \text{ and for } \lambda \rightarrow -\infty \text{ if } \bar{c}_j < 0$$

and the procedure is correct. Consequently, we can find a basis  $B$  for (Phase I) such that

$$\bar{c}_j = c_j - \mathbf{c}_B \mathbf{B}^{-1}\mathbf{a}_j = 0 \text{ for all } j \in F$$

if (BVLP) has a finite optimum solution, i.e., we can start the Dual BVSimplex Algorithm to solve the linear program (Phase I) as we did in the case where  $F = \emptyset$ .

Setting all variables in (Phase I) equal to zero we have a feasible solution and thus the optimal objective value of (Phase I) is greater than or equal to zero. Suppose (Phase I) produces an optimal objective function equal to zero. Then the optimal basis  $B$  cum partitioning  $(G^*, L^*, U^*)$ , say, provides a dual basis  $\bar{B}$  cum partitioning  $(G, L, U)$  for (BVLP) as follows. We compute  $\bar{c}_j = c_j - \mathbf{c}_B \mathbf{B}^{-1}\mathbf{a}_j$  for  $j \in C \cap NS$  and set

$$L = L^* \cup \{j \in C \cap NS : \bar{c}_j \leq 0\}, U = U^* \cup \{j \in C \cap NS : \bar{c}_j > 0\}.$$

Now suppose that (Phase I) has an optimal objective function value greater than zero. We claim that (BVLP) either has no feasible solution at all or that it has an unbounded optimum solution. So suppose there exists  $x \in \mathbb{R}^n$  that is feasible for (BVLP). We construct  $y \in \mathbb{R}^n$  satisfying  $Ay = 0$  by setting  $y_{C\cap NS} = 0$  and  $y$  equal to the optimal solution of (Phase I) otherwise. It follows that  $x + \lambda y$  is feasible for (BVLP) for all  $\lambda \geq 0$  and

$$c(x + \lambda y) = cx + \lambda cy \rightarrow +\infty \text{ for } \lambda \rightarrow +\infty$$

since  $cy > 0$ . Consequently, we know how to start the Dual BVSimplex Algorithm, if (BVLP) has a finite optimum solution. The efficient implementation of finding an initial dual basis for (BVLP) along the lines outlined here is left to the reader.

---

### \*Exercise 6.14

The traffic equations for a (single class general open queueing network model of a) job shop with  $n \geq 2$  work centers are given by

$$(P) \quad \sum_{i=1}^n \gamma_i = 1, \quad \sum_{i \neq j} p_{ji} \leq 1 \text{ for } j = 1, \dots, n, \quad \gamma_i + \sum_{j \neq i} v_j p_{ji} = v_i \text{ for } i = 1, \dots, n,$$

$$\gamma_i \geq 0 \text{ for } i = 1, \dots, n, \quad p_{ji} \geq 0 \text{ for } 1 \leq j \neq i \leq n,$$

where  $v_i$  denotes the expected number of visits a typical job makes to work center  $i$ .  $\gamma_i$  denotes the fraction of jobs arriving to the network that first visit work center  $i$ , while the “switching probabilities”  $p_{ji}$  denote the probability that a job upon completion of service at work center  $j$  visits work center  $i$ . The  $\gamma_i$  and  $p_{ji}$  are the “design parameters” of the job shop. The choice of their numerical values affects the performance of the job shop as measured e.g. by the number of jobs waiting to be served in the shop. The objective is to find design parameters that minimize the expected number of jobs in the entire shop (cf. Buzacott and Shanthikumar, *Stochastic Models of Manufacturing Systems*, Prentice Hall, 1993, pp.330). Also of interest is the conditional problem  $(P_\gamma)$  where for given values  $\gamma_1, \dots, \gamma_n$  satisfying  $0 \leq \gamma_i \leq v_i$  for  $i = 1, \dots, n$  and  $\sum_{i=1}^n \gamma_i = 1$  we wish to find switching probabilities  $p_{ji}$  satisfying

$$(P_\gamma) \quad \sum_{i \neq j} p_{ji} \leq 1 \text{ for } j = 1, \dots, n, \quad \sum_{j \neq i} v_j p_{ji} = v_i - \gamma_i \text{ for } i = 1, \dots, n,$$

$$p_{ji} \geq 0 \text{ for } 1 \leq j \neq i \leq n.$$

In this exercise we are solely interested in the solvability of  $(P)$  and  $(P_\gamma)$ . We will assume  $v_i > 0$  for  $i = 1, \dots, n$  and  $\sum_{i=1}^n v_i \geq 1$ , because otherwise a work center can be removed or neither system is solvable. Show:

- (i)  $(P_\gamma)$  is solvable if and only if  $v_i - \gamma_i \leq \sum_{j \neq i} v_j$  for  $i = 1, \dots, n$ .

(ii)  $(\mathcal{P})$  is solvable if and only if there exist  $\epsilon_i \geq 0$  with  $\epsilon_i \leq v_i$  for  $i = 1, \dots, n$  and  $\sum_{i=1}^n \epsilon_i = 1$  such that  $v_i - \epsilon_i \leq \sum_{j \neq i} v_j$  for  $i = 1, \dots, n$ .

(iii)  $(\mathcal{P})$  is solvable if and only if  $v_{j_{max}} - \sum_{j \neq j_{max}} v_j \leq 1$ , where  $j_{max}$  is such that  $v_{j_{max}} \geq v_j$  for all  $1 \leq j \leq n$ .

---

(i) By Farkas' Lemma  $(\mathcal{P}_\gamma)$  is solvable if and only if every solution  $\mathbf{u} = (u_1, \dots, u_n)$  and  $\mathbf{w} = (w_1, \dots, w_n)$  to

$$u_j + v_j w_i \geq 0 \text{ for } 1 \leq j \neq i \leq n, \quad u_j \geq 0 \text{ for } j = 1, \dots, n \quad (6.14)$$

satisfies  $\sum_{j=1}^n u_j + \sum_{i=1}^n (v_i - \gamma_i) w_i \geq 0$ , i.e., if and only if  $(\mathbf{u}, \mathbf{w}) \in \mathbb{R}^{2n}$  satisfying (6.14) and

$$\sum_{j=1}^n u_j + \sum_{i=1}^n (v_i - \gamma_i) w_i < 0 \quad (6.15)$$

do not exist.

Suppose that a condition is violated. Without loss of generality we can assume that

$$v_1 > \gamma_1 + \sum_{j=2}^n v_j.$$

Then  $w_1 = -1$ ,  $w_j = 0$  for  $j = 2, \dots, n$ ,  $u_1 = 0$  and  $u_j = v_j$  for  $j = 2, \dots, n$  satisfy (6.14) and (6.15), which is a contradiction. Hence the conditions are necessary.

Suppose the conditions are satisfied. Let  $(\mathbf{u}, \mathbf{w}) \in \mathbb{R}^{2n}$  be any solution to (6.14). Without loss of generality we can assume that

$$w_1 = \min_{1 \leq j \leq n} w_j \quad \text{and} \quad w_2 = \min_{2 \leq j \leq n} w_j.$$

Let  $w_1^* = \min\{0, w_1\}$  and  $w_2^* = \min\{0, w_2\}$ . From (6.14) it follows that

$$u_1 \geq -v_1 w_2^* \quad \text{and} \quad u_j \geq -v_j w_1^* \text{ for } j = 2, \dots, n. \quad (6.16)$$

From  $v_i - \gamma_i \geq 0$  for  $i = 1, \dots, n$ ,  $w_1 \geq w_1^*$ ,  $w_j \geq w_2 \geq w_2^*$  for  $j = 2, \dots, n$  and  $\sum_{i=1}^n \gamma_i = 1$  we calculate using (6.16)

$$\begin{aligned} & \sum_{j=1}^n u_j + \sum_{i=1}^n (v_i - \gamma_i) w_i \\ & \geq -v_1 w_2^* - \sum_{j=2}^n v_j w_1^* + (v_1 - \gamma_1) w_1^* + \sum_{j=2}^n (v_j - \gamma_j) w_2^* \\ & = (v_1 - \gamma_1 - \sum_{j=2}^n v_j) w_1^* - (v_1 - \sum_{j=2}^n (v_j - \gamma_j)) w_2^* \\ & = -w_1^* + (v_1 - \sum_{j=2}^n (v_j - \gamma_j))(w_1^* - w_2^*) \geq -w_2^* \geq 0, \end{aligned}$$

because  $w_1^* \leq w_2^* \leq 0$  and, by  $v_1 - \gamma_1 \leq \sum_{j=2}^n v_j$  and  $\sum_i \gamma_i = 1$ ,  $v_1 - \sum_{j=2}^n (v_j - \gamma_j) \leq 1$ . Since  $(\mathbf{u}, \mathbf{w}) \in \mathbb{R}^{2n}$  is any solution to (6.14), by Farkas' Lemma,  $(\mathcal{P}_\gamma)$  is solvable.

**(ii)** If  $(\mathcal{P})$  is solvable, let  $(\gamma, p)$  be any solution to  $(\mathcal{P})$ , where  $\gamma$  is the  $n$  vector of the  $\gamma_i$ 's and  $p$  is the  $n(n-1)$  vector corresponding to the  $p_{ji}$ 's. Then  $p$  is a solution to the conditional system  $(\mathcal{P}_\gamma)$  and thus by (i) the conditions of (ii) are met with  $\epsilon_i = \gamma_i$  for  $i = 1, \dots, n$ . Suppose the conditions are satisfied. By (i)  $(\mathcal{P}_\gamma)$  with  $\gamma_i = \epsilon_i$  for  $i = 1, \dots, n$  is solvable and hence so is  $(\mathcal{P})$ .

**(iii)** It suffices to show that  $\epsilon_i$  satisfying (ii) exist if and only if  $v_{j_{max}} - \sum_{j \neq j_{max}} v_j \leq 1$ . If  $v_{j_{max}} - \sum_{j \neq j_{max}} v_j > 1$ , then from  $v_i - \epsilon_i \leq \sum_{j \neq i} v_j$  for  $i = j_{max}$  it follows that  $\epsilon_{j_{max}} > 1$ , which is a contradiction. Thus the condition of (iii) is necessary. To show that it is sufficient, note that from the choice of  $j_{max}$

$$-v_{j_{max}} + \sum_{j \neq j_{max}} v_j \leq -v_i + \sum_{j \neq i} v_j \text{ for } 1 \leq i \leq m \text{ and } \sum_{j \neq i} v_j - v_i \geq 0 \text{ for all } i \neq j_{max}.$$

Thus the constraints  $v_i - \epsilon_i \leq \sum_{j \neq i} v_j$  of (ii) are redundant for all  $i \neq j_{max}$ . For  $i = j_{max}$  we get from  $\epsilon_i \leq v_i$  that  $v_{j_{max}} - \sum_{j \neq j_{max}} v_j \leq \epsilon_{j_{max}} \leq v_{j_{max}}$ . Since  $v_{j_{max}} - \sum_{j \neq j_{max}} v_j \leq 1$  the existence of  $\epsilon_i$  follows from  $\sum_i v_i \geq 1$ . Hence (iii) is correct.

## 7. Analytical Geometry

Μηδείς ἀγεωμέτρητος εἰσίτω μου τὴν στέγην.<sup>1</sup>  
Plato of Athens (c. 427–347 B.C.)

Here we summarize the essentials of Chapter 7 of the text. No proofs are given, but aside from the proofs the summary is intended to be self-contained and to serve as a study guide for an in-depth study of the subject of the polyhedral underpinnings of linear and combinatorial optimization using the text.

### 7.1 Points, Lines, Subspaces

A vector  $\mathbf{x} \in \mathbb{R}^n$  is also called a *point* of  $\mathbb{R}^n$ . The unit vectors  $\mathbf{u}_1, \dots, \mathbf{u}_n$  of  $\mathbb{R}^n$  form a *basis* of  $\mathbb{R}^n$ . The points  $\mathbf{x}^1, \dots, \mathbf{x}^t$  of  $\mathbb{R}^n$  are *affinely independent* if the unique solution to  $\lambda_1\mathbf{x}^1 + \dots + \lambda_t\mathbf{x}^t = \mathbf{0}$ ,  $\lambda_1 + \dots + \lambda_t = 0$  is  $\lambda_1 = \dots = \lambda_t = 0$ ; otherwise they are *affinely dependent*. For any  $\mathbf{x}^1, \dots, \mathbf{x}^t$  and scalars  $\mu_i$  satisfying  $\sum_{i=1}^t \mu_i = 1$  the point  $\mathbf{x} = \sum_{i=1}^t \mu_i \mathbf{x}^i$  is an *affine combination* of  $\mathbf{x}^1, \dots, \mathbf{x}^t$ .

The point  $\mathbf{u}^0 = \mathbf{0}$  is the *origin* of  $\mathbb{R}^n$ . The  $n+1$  points  $\mathbf{u}^0, \mathbf{u}_1, \dots, \mathbf{u}_n$  of  $\mathbb{R}^n$  are affinely independent and moreover, they are a *maximal* set of such points in  $\mathbb{R}^n$ .  $\mathbb{R}^n$  is the  $n$ -dimensional affine vector space (over the field of reals) and  $\mathbf{u}^0, \mathbf{u}_1, \dots, \mathbf{u}_n$  an (affine) *coordinate system* for  $\mathbb{R}^n$ .  $\mathbf{x}^1, \dots, \mathbf{x}^t \in \mathbb{R}^n$  are affinely independent if and only if  $\mathbf{x}^2 - \mathbf{x}^1, \dots, \mathbf{x}^t - \mathbf{x}^1$  are linearly independent.

**Definition DI** Let  $L \subseteq \mathbb{R}^n$  be any set. The *affine rank* of  $L$  is  $ar(L) = \max\{t : \mathbf{x}^1, \dots, \mathbf{x}^t \in L, \mathbf{x}^1, \dots, \mathbf{x}^t \text{ are affinely independent}, 0 \leq t < \infty\}$ . The *dimension* of  $L$ ,  $\dim L$ , equals  $ar(L) - 1$ . If  $\dim L = n$  then  $L$  is of full dimension.  $L$  is a *subspace* of  $\mathbb{R}^n$  if and only if for all  $\mathbf{x}, \mathbf{y} \in L$  and  $\lambda, \mu \in \mathbb{R}$  we have  $\lambda\mathbf{x} + \mu\mathbf{y} \in L$ .  $lin(L) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \sum_{i=1}^t \mu_i \mathbf{x}^i \text{ where } \mathbf{x}^i \in L, \mu_i \in \mathbb{R}, 1 \leq i \leq t, 0 \leq t < \infty\}$  is the *linear hull* of  $L$ .  $aff(L) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \sum_{i=1}^t \mu_i \mathbf{x}^i \text{ where } \mathbf{x}^i \in L, \mu_i \in \mathbb{R}, \sum_{i=1}^t \mu_i = 1, 1 \leq i \leq t, 0 \leq t < \infty\}$  is the *affine hull* of  $L$ . By convention,  $lin(\emptyset) = \{\mathbf{0}\}$  and  $aff(\emptyset) = \emptyset$ .

For any  $L \subseteq \mathbb{R}^n$ ,  $lin(L)$  is a subspace of  $\mathbb{R}^n$ . If  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} \neq \mathbf{0}$ , and  $L = \{\mathbf{x}\}$ ,  $lin(L)$  is a *line* of  $\mathbb{R}^n$  that passes through 0. For any  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} \neq \mathbf{0}$ , let  $(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} = \lambda\mathbf{x} \text{ for any } \lambda \geq 0\}$ ;  $(\mathbf{x})$  is the *halfline* containing the origin and evidently,  $lin(\{\mathbf{x}\}) = (\mathbf{x}) + (-\mathbf{x})$ . There we have used the convention that for any two sets  $S, T \subseteq \mathbb{R}^n$  the *sum* of  $S$  and  $T$  is defined by

$$S + T = \{z \in \mathbb{R}^n : \exists \mathbf{x} \in S, \mathbf{y} \in T \text{ such that } z = \mathbf{x} + \mathbf{y}\}.$$

If  $L = \{\mathbf{x}^1, \mathbf{x}^2\}$  and  $\mathbf{x}^1 \neq \mathbf{x}^2$ , then  $aff(L)$  is the *line* of  $\mathbb{R}^n$  that passes through  $\mathbf{x}^1$  and  $\mathbf{x}^2$ . Since  $\mathbf{x} = \mu_1\mathbf{x}^1 + \mu_2\mathbf{x}^2 = \mathbf{x}^1 + \mu_2(\mathbf{x}^2 - \mathbf{x}^1)$  for all  $\mu_1, \mu_2 \in \mathbb{R}$  with  $\mu_1 + \mu_2 = 1$  we get  $aff(\{\mathbf{x}^1, \mathbf{x}^2\}) = \mathbf{x}^1 + lin(\{\mathbf{x}^2 - \mathbf{x}^1\})$  and the vector  $\mathbf{y} = \mathbf{x}^2 - \mathbf{x}^1 \neq \mathbf{0}$  is the *direction vector* of the line  $aff(\{\mathbf{x}^1, \mathbf{x}^2\})$ .

If  $L = \{\mathbf{x}^1, \dots, \mathbf{x}^t\}$  then  $aff(L) = \mathbf{x}^1 + lin(L - \mathbf{x}^1)$  so that  $aff(L)$  is a *displaced subspace*.  $\mathbf{x}^1$  is the *displacement* (or *translation*) vector and  $L$  an *affine subspace* of  $\mathbb{R}^n$ , for short. Every point of  $\mathbb{R}^n$  is an affine subspace of  $\mathbb{R}^n$  and so is every line of  $\mathbb{R}^n$ .  $aff(L) = lin(L)$  if and only if  $0 \in aff(L)$ .

A function  $N : \mathbb{R}^n \rightarrow \mathbb{R}$  is a *norm* on  $\mathbb{R}^n$  if it satisfies (i)  $N(\mathbf{x}) \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^n$ ; (ii)  $N(\mathbf{x}) = 0$  if and only if  $\mathbf{x} = \mathbf{0}$ ; (iii)  $N(\alpha\mathbf{x}) = \alpha N(\mathbf{x})$  for all  $\mathbf{x} \in \mathbb{R}^n$  and  $\alpha \geq 0$  and (iv)  $N(\mathbf{x} + \mathbf{y}) \leq N(\mathbf{x}) + N(\mathbf{y})$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . Point (iv) is the **triangle inequality**. Every norm  $N$  on  $\mathbb{R}^n$  induces a measure of *distance* between any two points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  via the *distance function*  $d_N(\mathbf{x}, \mathbf{y}) = N(\mathbf{x} - \mathbf{y})$ . The

<sup>1</sup>No non-geometrician shall enter under my roof.

distance of  $\mathbf{x}$  from  $\mathbf{0}$  is the norm of  $\mathbf{x}$  or the *length* of  $\mathbf{x}$  in the norm  $N$ . For  $L \subseteq \mathbb{R}^n$  the distance of  $\mathbf{x} \in \mathbb{R}^n$  from  $L$  in the norm  $N$  is  $d_N(L, \mathbf{x}) = \min\{d_N(\mathbf{y}, \mathbf{x}) : \mathbf{y} \in L\}$ . The *Euclidean norm* or  $\ell_2$ -norm  $\|\mathbf{x}\|_2$ , the  $\ell_1$ -norm  $\|\mathbf{x}\|_1$  and the  $\ell_\infty$ -norm  $\|\mathbf{x}\|_\infty$  of  $\mathbf{x} \in \mathbb{R}^n$ ,

$$\|\mathbf{x}\|_2 = \left( \sum_{i=1}^n x_i^2 \right)^{1/2}, \quad \|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|, \quad \|\mathbf{x}\|_\infty = \max\{|x_i| : i = 1, \dots, n\},$$

are the most frequently encountered norms on  $\mathbb{R}^n$ .  $\|\mathbf{x}\|_2$  is generally abbreviated by  $\|\mathbf{x}\|$  without the subscript 2 and the associated distance function  $\|\mathbf{x} - \mathbf{y}\|$  is denoted by  $d(\mathbf{x}, \mathbf{y})$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . All three functions  $\|\cdot\|_p$  are norms on  $\mathbb{R}^n$  where  $p \in \{1, 2, \infty\}$ . The function  $\|\mathbf{x}\|$  defines a norm on  $\mathbb{R}^n$  because of the following C-S inequality which is also true if  $\mathbf{x} = \mathbf{0}$  or  $\mathbf{y} = \mathbf{0}$ .

**7.1(a) (Cauchy-Schwarz inequality)** For any  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ,  $\mathbf{x} \neq \mathbf{0} \neq \mathbf{y}$ , we have  $|\mathbf{x}^T \mathbf{y}| \leq \|\mathbf{x}\| \|\mathbf{y}\|$  with equality if and only if  $\mathbf{x} = \alpha \mathbf{y}$  for some  $\alpha \in \mathbb{R}$ .

The angle  $\phi = \phi_{\mathbf{x}\mathbf{y}}$  between two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  is given by  $\cos \phi = \frac{(\mathbf{x})^T \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$ .

Two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  are *orthogonal* if and only if  $\mathbf{y}^T \mathbf{x} = 0$ . By convention, the vector  $\mathbf{0}$  is orthogonal to all vectors  $\mathbf{x} \in \mathbb{R}^n$ . Two nonzero vectors are parallel if and only if they are linearly dependent.

For any  $L \subseteq \mathbb{R}^n$  the set  $L^\perp = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^T \mathbf{x} = 0 \text{ for all } \mathbf{x} \in L\}$  is the *orthogonal complement* of  $L$  which is a subspace of  $\mathbb{R}^n$ .  $L^\perp = \mathbb{R}^n$  if  $L = \emptyset$  or  $L = \{\mathbf{0}\}$ ,  $L^\perp = \{\mathbf{0}\}$  if  $L = \mathbb{R}^n$ . It follows that  $\text{lin}(L) + L^\perp = \text{aff}(L) + L^\perp = \mathbb{R}^n$  and  $\dim L + \dim L^\perp = n$ .

Let  $L \neq \emptyset$  be any subspace of  $\mathbb{R}^n$  and  $\mathbf{x}^1, \dots, \mathbf{x}^t \in L$  be a *maximal* set of linearly independent points in  $L$ , i.e., a *basis* of  $L$ . Let  $\{\mathbf{y}^1, \dots, \mathbf{y}^s\}$  be a basis of  $L^\perp$ , where  $s = n - t$ . Let  $\mathbf{X} = (\mathbf{x}^1 \cdots \mathbf{x}^t)$  and  $\mathbf{Y} = (\mathbf{y}^1 \cdots \mathbf{y}^{n-t})$ . Then  $L = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{X}\mu \text{ for } \mu \in \mathbb{R}^t\} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Y}^T \mathbf{x} = 0\}$ . The points  $\mathbf{x}^1, \dots, \mathbf{x}^t$  generate all of the points of  $L$ , i.e.,  $\{\mathbf{x}^1, \dots, \mathbf{x}^t\}$  is a **finite generator** of  $L$ . The points  $\mathbf{y}^1, \dots, \mathbf{y}^{n-t}$  define homogeneous linear equations  $(\mathbf{y}^i)^T \mathbf{x} = 0$  that linearly describe  $L$ , i.e.,  $(\mathbf{Y}^T, \mathbf{0})$  is a **finite linear description** of  $L$ . Every subspace of  $\mathbb{R}^n$  possesses both a finite generator and a finite linear description. If  $L$  is a (nonempty) affine subspace, then, likewise,  $L$  has a finite generator and a finite linear description by way of *inhomogeneous* linear equations. Every subspace and affine subspace of  $\mathbb{R}^n$  thus possesses a **double description**.

Let  $\mathbf{a} \in \mathbb{R}^n$  be any row vector and  $a_0 \in \mathbb{R}$  be a scalar. The sets  $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}\mathbf{x} < a_0\}$ ,  $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}\mathbf{x} \leq a_0\}$ ,  $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}\mathbf{x} = a_0\}$  are the *open halfspace*, the *(closed) halfspace* and the *hyperplane* defined by  $(\mathbf{a}, a_0)$ , respectively.

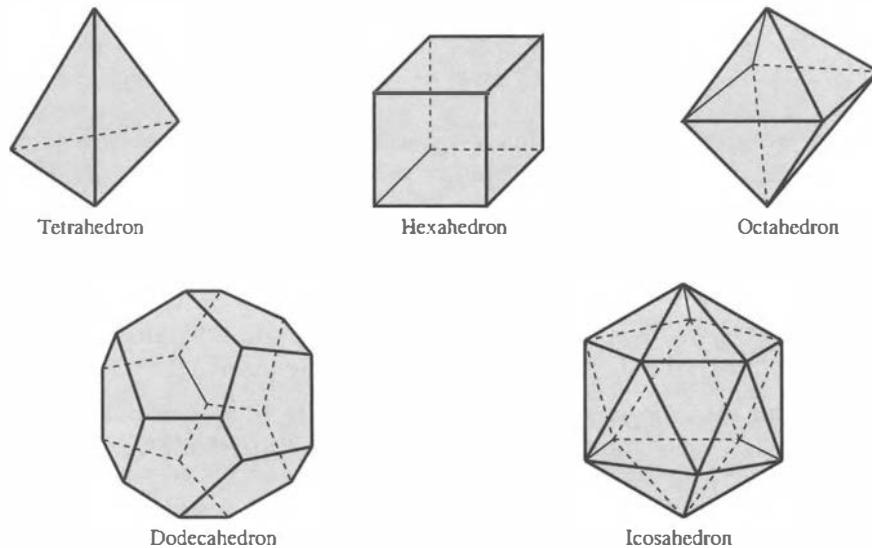
Let  $\mathbf{x}^1, \dots, \mathbf{x}^t$  be a basis of some subspace  $L \subseteq \mathbb{R}^n$  and like above,  $\mathbf{X} = (\mathbf{x}^1 \cdots \mathbf{x}^t)$ . The linear transformations from  $\mathbb{R}^n$  to  $\mathbb{R}^n$  given by  $\mathbf{y} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{x}$  and  $\mathbf{z} = (\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \mathbf{x}$  are the *orthogonal projections* of  $\mathbb{R}^n$  onto  $L$  and  $L^\perp$ , respectively. Every  $\mathbf{x} \in \mathbb{R}^n$  can be written *uniquely* as  $\mathbf{x} = \mathbf{y} + \mathbf{z}$  where  $\mathbf{y} \in L$  and  $\mathbf{z} \in L^\perp$ , i.e.  $\mathbf{y}^T \mathbf{z} = 0$ .

If a basis  $\{\mathbf{y}^1, \dots, \mathbf{y}^{n-t}\}$  of  $L^\perp$  is given then the orthogonal projections of  $\mathbb{R}^n$  onto  $L$  and  $L^\perp$  are given by  $\mathbf{y} = (\mathbf{I}_n - \mathbf{Y}(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T) \mathbf{x}$  and  $\mathbf{z} = \mathbf{Y}(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{x}$ , respectively.

Let  $\mathbf{A}$  be any  $n \times n$  matrix with columns  $\mathbf{a}_i$  for  $1 \leq i \leq n$ . We construct a new matrix  $\mathbf{B}$  with columns  $\mathbf{b}_k$  *recursively* from the matrix  $\mathbf{A}$  as follows

$$\mathbf{b}_1 = \mathbf{a}_1, \quad \mathbf{b}_{k+1} = \mathbf{a}_{k+1} - \sum_{i=1}^k \frac{\mathbf{a}_{k+1}^T \mathbf{b}_i}{\|\mathbf{b}_i\|^2} \mathbf{b}_i,$$

where  $k = 1, \dots, n-1$  provided the calculations are well defined.

**Fig. 7.1.** The Platonic solids

**7.1(b) (Gram-Schmidt orthogonalization)** With the above notation, if  $A$  is nonsingular then  $B$  is nonsingular, orthogonal ( $b_k^T b_j = 0$  for  $k \neq j$ ) and  $|\det B| = |\det A| = \prod_{i=1}^n \|b_i\|$ .

**7.1(c) (Hadamard inequality)**  $|\det A| \leq \prod_{i=1}^n \|a_i\|$ . for every  $n \times n$  matrix  $A = (a_1 \dots a_n)$  with equality if and only if  $a_i^T a_j = 0$  for all  $1 \leq i < j \leq n$ .

## 7.2 Polyhedra, Ideal Descriptions, Cones

**Definition P1** A set  $P \subseteq \mathbb{R}^n$  is a polyhedron if and only if there exists an  $m \times n$  matrix  $H$  and a vector  $\mathbf{h}$  of  $m$  real numbers such that  $P = \{\mathbf{x} \in \mathbb{R}^n : H\mathbf{x} \leq \mathbf{h}\}$  where  $0 \leq m < \infty$ . The system of inequalities  $H\mathbf{x} \leq \mathbf{h}$  is a linear description of  $P$ .

In other words, a polyhedron is the intersection of finitely many (closed) halfspaces in some finite dimensional vector space (over the field of reals). We write  $P = P(H, \mathbf{h})$  to denote the polyhedron defined by  $H, \mathbf{h}$  and make the blanket assumption that  $(H, \mathbf{h})$  has no row consisting entirely of zeroes only. Let  $\mathbf{x}(\lambda) = \mathbf{x}^0 + \lambda \mathbf{y}$  for some  $\mathbf{x}^0, \mathbf{y} \in \mathbb{R}^n$ ,  $\mathbf{y} \neq 0$  be some line in  $\mathbb{R}^n$  where  $\lambda \in \mathbb{R}$  is a parameter and  $\mathbf{y}$  is the direction vector of the line  $\mathbf{x}(\lambda)$ . Then  $\mathbf{x}(\lambda) \in P$  for all  $\lambda \in \mathbb{R}$  if and only if  $\mathbf{x}^0 \in P$  and  $H\mathbf{y} = 0$ . A polyhedron  $P$  in  $\mathbb{R}^n$  may contain lines. We let  $L_P = \{\mathbf{x} \in \mathbb{R}^n : H\mathbf{x} = 0\}$  be the linearity space of  $P$ .

**7.2(a)** A polyhedron  $P(H, \mathbf{h})$  is line free if and only if the rank of  $H$  equals  $n$ .

Let  $L_P^\perp = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T \mathbf{y} = 0 \text{ for all } \mathbf{y} \in L_P\}$  be the orthogonal complement of  $L_P$  in  $\mathbb{R}^n$ . Then  $\dim L_P^\perp = n - r(H)$ ,  $\dim L_P^\perp = r(H)$  and  $L_P = \{0\}$  if and only if  $L_P^\perp = \mathbb{R}^n$ . Let the rows of the matrix  $G$  correspond to the vectors of a basis of the subspace  $L_P$ . Then  $G$  has  $n - r(H)$  rows and

$n$  columns,  $r(\mathbf{G}) = n - r(\mathbf{H})$  and  $L_P^\perp = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{G}\mathbf{x} = 0\}$ . Thus

$$P^0 = P \cap L_P^\perp = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}\mathbf{x} \leq \mathbf{h}, \mathbf{G}\mathbf{x} \leq 0, -\mathbf{G}\mathbf{x} \leq 0\}$$

is a linefree polyhedron and every  $\mathbf{x} \in P$  has a unique representation  $\mathbf{x} = \mathbf{y} + \ell$  where  $\mathbf{y} \in P^0$  and  $\ell \in L_P$ , i.e., we have the orthogonal decomposition  $P = P^0 + L_P$ , and  $\dim P(\mathbf{H}, \mathbf{h}) = d_0 + n - r(\mathbf{H})$  for any polyhedron  $P = P(\mathbf{H}, \mathbf{h})$ , where  $d_0 = \dim P^0$ .

**Definition EP** Let  $P \subseteq \mathbb{R}^n$  be any set.  $\mathbf{x} \in P$  is an extreme point of  $P$  if and only if for any  $\mathbf{x}^1, \mathbf{x}^2 \in P$  and  $0 < \mu < 1$  such that  $\mathbf{x} = \mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2$  it follows that  $\mathbf{x} = \mathbf{x}^1 = \mathbf{x}^2$ .

If  $P$  contains a line then we say that  $P$  is **blunt**. Blunt polyhedra have no extreme points. We say that a polyhedron  $P$  is **pointed** if  $P$  has at least one extreme point.

**7.2(b)**  $\mathbf{x}^0 \in \mathbb{R}^n$  is an extreme point of a polyhedron  $P(\mathbf{H}, \mathbf{h})$  if and only if  $\mathbf{H}\mathbf{x}^0 \leq \mathbf{h}$  and  $\mathbf{H}_1\mathbf{x}^0 = \mathbf{h}_1$  for some  $n \times (n+1)$  submatrix  $(\mathbf{H}_1, \mathbf{h}_1)$  of  $(\mathbf{H}, \mathbf{h})$  with  $r(\mathbf{H}_1) = n$ .

**7.2(c)** A line free polyhedron  $P(\mathbf{H}, \mathbf{h})$  is pointed if and only if it is nonempty.

**7.2(d)**  $\mathbf{x}^0 \in \mathbb{R}^n$  is an extreme point of a polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  if and only if for some row vector  $\mathbf{c} \in \mathbb{R}^n$  we have  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\} = \mathbf{c}\mathbf{x}^0 > \mathbf{c}\mathbf{x}$  for all  $\mathbf{x} \in P, \mathbf{x} \neq \mathbf{x}^0$ .

### 7.2.1 Faces, Valid Equations, Affine Hulls

**Definition FA** Let  $P \subseteq \mathbb{R}^n$  be a polyhedron and  $F \subseteq \mathbb{R}^n$  be any set.

(i)  $F$  is a face of  $P$  if and only if there exists a row vector  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  with  $\mathbf{f} \neq 0$  such that  $F = \{\mathbf{x} \in P : \mathbf{f}\mathbf{x} = f_0\}$  and  $\mathbf{f}\mathbf{x} < f_0$  for all  $\mathbf{x} \in P, \mathbf{x} \notin F$ . If  $F$  is a face of  $P$  and  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  such that  $F = \{\mathbf{x} \in P : \mathbf{f}\mathbf{x} = f_0\}$  then  $\mathbf{f}\mathbf{x} \leq f_0$  defines (or induces or determines) the face  $F$ .

(ii)  $F$  is a facet of  $P$  if and only if  $F$  is a face of  $P$  and  $\dim F = \dim P - 1$ .

(iii) A face of dimension 0 is an extreme point of  $P$ . A face of dimension 1 is an edge of  $P$ . Two extreme points of  $P$  are adjacent if they are contained in an edge of  $P$ . A nonempty face  $F$  is a proper face of  $P$  if  $\dim F < \dim P$ ; it is an improper face if  $\dim F = \dim P$ . A polyhedron is full dimensional if  $\dim P = n$ .

In the case that  $P$  is of full dimension, we drop the requirement that  $\mathbf{f} \neq 0$  in the definition of an improper face, so that  $P$  is always an improper face of itself.

**7.2(e)** A nonempty set  $F \subseteq \mathbb{R}^n$  is a proper face of dimension  $k$  of a polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  if and only if there exists a partitioning of  $(\mathbf{H}, \mathbf{h})$  into two submatrices  $(\mathbf{H}_1, \mathbf{h}_1)$  and  $(\mathbf{H}_2, \mathbf{h}_2)$  such that  $r(\mathbf{H}_1) = n - k$ ,  $F = \{\mathbf{x} \in P : \mathbf{H}_1\mathbf{x} = \mathbf{h}_1\}$  and  $\mathbf{H}_2\mathbf{x} < \mathbf{h}_2$  for some  $\mathbf{x} \in F$ , where  $0 \leq k \leq \dim P - 1$ .

The faces of smallest dimension of a polyhedron are the **minimal faces** of the polyhedron, i.e. they are precisely those faces of  $P$  that have no proper (sub-)faces.

**7.2(f)** The minimal faces of a nonempty polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  have the dimension  $n - r(\mathbf{H})$  and the minimal faces of the polyhedron  $P^0 = P \cap L_P^\perp$  are precisely the extreme points of  $P^0$ .

If for any row  $(\mathbf{h}^i, h_i)$  of  $(\mathbf{H}, \mathbf{h})$   $\mathbf{h}^i \mathbf{x} = h_i$  for all  $\mathbf{x} \in P$ , then we call  $\mathbf{h}^i \mathbf{x} = h_i$  a **valid equation** for  $P$ . We denote by  $(\mathbf{H}^=, \mathbf{h}^=)$  the (possibly empty) submatrix of  $(\mathbf{H}, \mathbf{h})$  of all valid equations for  $P$  and by  $(\mathbf{H}^<, \mathbf{h}^<)$  the remaining rows of  $(\mathbf{H}, \mathbf{h})$ . Every polyhedron can thus be written as

$$P(\mathbf{H}, \mathbf{h}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}^= \mathbf{x} = \mathbf{h}^=, \mathbf{H}^< \mathbf{x} \leq \mathbf{h}^<\}. \quad (7.1)$$

If  $P \neq \emptyset$  we can assume WROG that  $r(\mathbf{H}^=) = m^=$ , where  $m^=$  is the number of rows of  $(\mathbf{H}^=, \mathbf{h}^=)$ . The **relative interior** of the polyhedron  $P$  is

$$\text{relint } P(\mathbf{H}, \mathbf{h}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}^= \mathbf{x} = \mathbf{h}^=, \mathbf{H}^< \mathbf{x} < \mathbf{h}^<\}. \quad (7.2)$$

The relative interior of  $P$  is the set of points of  $P$  that are not contained in any proper face of  $P$ .

**7.2(g)** For a nonempty polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  let  $(\mathbf{H}^=, \mathbf{h}^=)$  and  $(\mathbf{H}^<, \mathbf{h}^<)$  be the partition of  $(\mathbf{H}, \mathbf{h})$  defined in (7.1). Then  $\dim P = n - r(\mathbf{H}^=)$ ,  $\text{aff}(P) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}^= \mathbf{x} = \mathbf{h}^=\}$  and  $\dim P^0 = r(\mathbf{H}) - r(\mathbf{H}^=)$  where  $P^0 = P \cap L_P^\perp$ .

Polyhedra  $P \subseteq \mathbb{R}^n$  are called **solids** if  $\dim P = n$  and **flats** otherwise.

### 7.2.2 Facets, Minimal Complete Descriptions, Quasi-Uniqueness

Denote by  $d = \dim P$  the dimension of a polyhedron  $P$  and define  $d = -1$  if  $P = \emptyset$ . If  $F$  is a face of dimension  $k$  of  $P(\mathbf{H}, \mathbf{h}) \neq \emptyset$  then  $k$  satisfies  $0 \leq n - r(\mathbf{H}) \leq k \leq n - r(\mathbf{H}^=) = d$ . We index the rows of  $(\mathbf{H}, \mathbf{h})$  corresponding to  $(\mathbf{H}^<, \mathbf{h}^<)$  by  $1, \dots, m^<$ , those of  $(\mathbf{H}^=, \mathbf{h}^=)$  by  $m^< + 1, \dots, m^< + m^=$  and let  $M = \{1, \dots, m^< + m^=\}$  and  $M^< = \{1, \dots, m^<\}$ .

**7.2(h)** Let  $P = P(\mathbf{H}, \mathbf{h}) \neq \emptyset$  and  $(\mathbf{H}, \mathbf{h})$  be partitioned like in (7.1). An inequality  $\mathbf{h}^j \mathbf{x} \leq h_j$  with  $j \in M^<$  is redundant relative to  $\mathbf{H}^{M-j} \mathbf{x} \leq \mathbf{h}_{M-j}$  if and only if  $\mathcal{U}_j \neq \emptyset$  where  $\mathcal{U}_j = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{m^< + m^=} : \mathbf{u} \mathbf{H}^< + \mathbf{v} \mathbf{H}^= = \mathbf{0}, \mathbf{u} \mathbf{h}^< + \mathbf{v} \mathbf{h}^= \leq \mathbf{0}, u_j = -1, u_k \geq 0 \text{ for all } k \in M^< - j\}$ .

**7.2(i)** With the same notation as under 7.2(h), if for some  $j \in M^<$  the inequality  $\mathbf{h}^j \mathbf{x} \leq h_j$  is not redundant, then the set  $F_j = \{\mathbf{x} \in P : \mathbf{h}^j \mathbf{x} = h_j\}$  is a facet of  $P$ .

**7.2(j)** With the same notation as under 7.2(h) and 7.2(i), if for  $j \in M^<$  the inequality  $\mathbf{h}^j \mathbf{x} \leq h_j$  is redundant and  $F_j$  a facet of  $P$ , then there exist  $(\mathbf{u}, \mathbf{v}) \in \mathcal{U}_j$  such that  $\mathbf{u} \mathbf{h}^< + \mathbf{v} \mathbf{h}^= = \mathbf{0}$  and  $k \in M^< - j$  such that  $F_k = F_j$  where  $F_k = \{\mathbf{x} \in P : \mathbf{h}^k \mathbf{x} = h_k\}$ .

Starting from any list of inequalities  $\mathbf{H} \mathbf{x} \leq \mathbf{h}$  defining a nonempty polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  partition  $(\mathbf{H}, \mathbf{h})$  into  $(\mathbf{H}^=, \mathbf{h}^=)$  and  $(\mathbf{H}^<, \mathbf{h}^<)$  as done in (7.1). From among the valid equations  $\mathbf{H}^= \mathbf{x} = \mathbf{h}^=$  retain only any subset  $(\mathbf{H}_1^=, \mathbf{h}_1^=)$  of  $r(\mathbf{H}^=)$  linearly independent rows of  $(\mathbf{H}^=, \mathbf{h}^=)$ . From among the inequalities  $\mathbf{H}^< \mathbf{x} \leq \mathbf{h}^<$  one needs to retain only any subset  $(\mathbf{H}_1^<, \mathbf{h}_1^<)$  such that every corresponding inequality defines a distinct facet of  $P$ . We call any such reduced system of linear equations and/or inequalities that describe a nonempty polyhedron  $P$  a **minimal complete description** of  $P$  or an **ideal description** of  $P$ , for short. Of course, different ideal descriptions of a nonempty polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  may exist as is shown by point 7.2(j). Nevertheless, every ideal description of a nonempty polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  is **quasi-unique**.

**7.2(k)** Let  $(\mathbf{H}, \mathbf{h})$  be an ideal description of a polyhedron  $P = P(\mathbf{H}, \mathbf{h})$ ,  $L_P$  the lineality space of  $P$  and  $P^0 = P \cap L_P^\perp$ . An ideal description of the polyhedron  $P^0$  is given by  $\mathbf{H}^< \mathbf{x} \leq \mathbf{h}^<$ ,  $\mathbf{H}^= \mathbf{x} = \mathbf{h}^=$ ,  $\mathbf{G} \mathbf{x} = \mathbf{0}$ , where  $(\mathbf{H}, \mathbf{h})$  is partitioned like in (7.1) and the rows of  $\mathbf{G}$  are a basis of  $L_P$ .

### 7.2.3 Asymptotic Cones and Extreme Rays

Let  $x(\lambda) = x + \lambda y$  for  $\lambda \geq 0$  be any halfline in  $\mathbb{R}^n$  where  $y \neq 0$  is a direction vector. Then  $x(\lambda) \in P = P(H, h)$  for all  $\lambda \geq 0$  if and only if  $x \in P$  and  $Hy \leq 0$ . We define

$$C_\infty(H) = \{y \in \mathbb{R}^n : Hy \leq 0\} \quad (7.3)$$

or  $C_\infty = C_\infty(H)$ , for short.  $C_\infty$  is the **asymptotic cone** (or *recession cone* or *characteristic cone*) of  $P$ .

#### Definition CO

- (i) A subset  $C \subseteq \mathbb{R}^n$  is a cone if and only if  $y^1, y^2 \in C$  implies  $\lambda_1 y^1 + \lambda_2 y^2 \in C$  for all  $\lambda_1 \geq 0$  and  $\lambda_2 \geq 0$ .
- (ii) The lineality space  $L_C$  of a cone  $C$  is the set  $L_C = \{y \in \mathbb{R}^n : y \in C \text{ and } -y \in C\}$ .
- (iii) A cone  $C$  is pointed if and only if  $y \in C$  and  $-y \in C$  imply  $y = 0$ .
- (iv) A halfline  $(y)$  is an extreme ray of a cone  $C$  if and only if  $(y) \in C$ ,  $(-y) \notin C$  and for any  $y^1, y^2 \in C$  and  $0 < \mu < 1$  such that  $y = \mu y^1 + (1 - \mu)y^2$  it follows that  $(y) = (y^1) = (y^2)$ .

Since  $C_\infty$  is also a polyhedron it is called a *polyhedral cone*. Every polyhedron  $P = P(H, h)$  with  $h = 0$  is a polyhedral cone and vice versa. While Definition CO applies to arbitrary (convex) cones none of the cones that are of interest to us are shaped like “ice-cream” cones.  $C_\infty$  is line free if and only if  $P$  is line free. Like in the case of  $P$ , let the rows of  $G$  correspond to a basis of  $L_P$ . Then the set  $C_\infty^0 = C_\infty \cap L_P^\perp$ , i.e.

$$C_\infty^0 = \{y \in \mathbb{R}^n : Hy \leq 0, Gy = 0\}, \quad (7.4)$$

is a pointed cone and we have the orthogonal decomposition  $C_\infty = L_P + C_\infty^0$ . By point 7.2(f), the minimal faces of  $C_\infty$  have the dimension  $n - r(H)$  and the cone  $C_\infty^0$  has exactly one extreme point  $y = 0$ .

**7.2(l)** A halfline  $(y)$  is an extreme ray of  $C_\infty^0$  if and only if  $y \in C_\infty^0$  and there are exactly  $r(H) - 1$  linearly independent rows  $h^i$  of  $H$  such that  $h^i y = 0$ .

The extreme rays of  $C_\infty^0$  are precisely the faces of dimension one of  $C_\infty^0$ , i.e. they are *edges* of the polyhedron  $C_\infty^0$ . If the cone  $C_\infty(H)$  contains lines then it does not have extreme rays in the sense of Definition CO.

### 7.2.4 Adjacency I, Extreme Rays of Polyhedra, Homogenization

**7.2(m)** Let  $P = P(H, h)$  be a polyhedron,  $L_P$  its lineality space,  $P^0 = P \cap L_P^\perp$ ,  $C_\infty$  its asymptotic cone and  $C_\infty^0 = C_\infty \cap L_P^\perp$ . If  $F \subseteq \mathbb{R}^n$  is a 1-dimensional face of  $P^0$  then every  $x \in F$  is either an element of some halfline  $x^0 + (y)$  where  $x^0$  is an extreme point of  $P^0$  and  $y$  an extreme ray of  $C_\infty^0$  or  $x = \mu x^0 + (1 - \mu)x^1$  for some  $0 \leq \mu \leq 1$  where  $x^0 \neq x^1$  are two adjacent extreme points of  $P^0$ .

**7.2(n)** With the same notation as under 7.2(m), let  $x^0 \neq x^1$  be two extreme points of  $P^0$  and  $(y)$  be an extreme ray of  $C_\infty^0$ . The halfline  $x^0 + (y)$  is a 1-dimensional face of  $P^0$  if and only if for all  $w \in \mathbb{R}^n$ ,  $w \neq 0$  such that  $x^0 + y + w \in P^0$  and  $x^0 + y - w \in P^0$  it follows that

$(w) = (\mathbf{y})$ . Two extreme points  $\mathbf{x}^0$  and  $\mathbf{x}^1$  are adjacent if and only if for all  $w \in \mathbb{R}^n$ ,  $w \neq 0$  such that  $\mathbf{x} + w \in P^0$  and  $\mathbf{x} - w \in P^0$  it follows that  $w = \lambda(\mathbf{x}^1 - \mathbf{x}^0)$  for some  $\lambda \in \mathbb{R}$ , where  $\mathbf{x} = (1/2)\mathbf{x}^0 + (1/2)\mathbf{x}^1$ .

If  $P^0$  has a 1-dimensional face of the form  $\mathbf{x}^0 + (\mathbf{y})$  where  $\mathbf{x}^0$  is an extreme point of  $P^0$  and  $(\mathbf{y})$  an extreme ray of  $C_\infty^0$  then  $P^0$  and hence  $P$  is *unbounded*, i.e. there exist no finite  $K > 0$  such that  $P \subseteq \{\mathbf{x} \in \mathbb{R}^n : -K \leq x_j \leq K \text{ for } j = 1, \dots, n\}$ .

**7.2(o)** With the notation as under 7.2(m), if  $P^0$  is nonempty and  $(\mathbf{y})$  an extreme ray of  $C_\infty^0$  then there exists an extreme point  $\mathbf{x}^0 \in P^0$  such that the halfline  $\mathbf{x}^0 + (\mathbf{y})$  is a 1-dimensional face of  $P^0$ .

It follows that the extreme rays of  $C_\infty^0$  are in one-to-one correspondence with the direction vectors  $\mathbf{y} \in \mathbb{R}^n$  that define the *unbounded* 1-dimensional faces of  $P^0$ . We call the extreme rays  $(\mathbf{y})$  of  $C_\infty^0$  the **extremal directions** and the corresponding halflines  $\mathbf{x}^0 + (\mathbf{y})$  given by point 7.2(o) the **extreme rays** of  $P^0$ . For a given list of inequalities  $H\mathbf{x} \leq \mathbf{h}$  denote

$$HP(H, \mathbf{h}) = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : H\mathbf{x} - \mathbf{h}x_{n+1} \leq 0, -x_{n+1} \leq 0\}, \quad (7.5)$$

where we have written  $(\mathbf{x}, x_{n+1})$  rather than  $(\mathbf{x}^T, x_{n+1})^T$  to denote  $(x_1, \dots, x_n, x_{n+1})^T \in \mathbb{R}^{n+1}$ .  $HP = HP(H, \mathbf{h})$  is a polyhedral cone in  $\mathbb{R}^{n+1}$ . Intersecting  $HP(H, \mathbf{h})$  with the hyperplane  $x_{n+1} = 1$  one gets the polyhedron  $P(H, \mathbf{h})$  embedded into  $\mathbb{R}^{n+1}$  in the natural way. This technique of embedding a polyhedron  $P$  in  $\mathbb{R}^n$  into a polyhedral cone in  $\mathbb{R}^{n+1}$  is known as the **homogenization technique**.

### 7.3 Point Sets, Affine Transformations, Minimal Generators

**Definition HU** Let  $S \subseteq \mathbb{R}^n$  be any set.  $\text{conv}(S) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \sum_{i=1}^t \mu_i \mathbf{x}^i \text{ where } \mu \in \Lambda_t, \mathbf{x}^i \in S, 1 \leq i \leq t, 0 \leq t < \infty\}$  is the *convex hull* of  $S$ , if  $\Lambda_t = \{\mu \in \mathbb{R}^t : \mu \geq 0, \sum_{i=1}^t \mu_i = 1\}$  and the *conical hull* of  $S$ , or  $\text{cone}(S)$  for short, if  $\Lambda_t = \{\mu \in \mathbb{R}^t : \mu \geq 0\}$ . By convention,  $\text{conv}(\emptyset) = \emptyset$  and  $\text{cone}(\emptyset) = \{0\}$ .

**7.3(a)** For any set  $S \subseteq \mathbb{R}^n$  we have  $\dim \text{conv}(S) = \dim S$ ,  $\text{conv}(S) \subseteq \text{cone}(S)$ ,  $\text{cone}(\text{conv}(S)) = \text{cone}(S)$  and  $\dim \text{cone}(S) = \dim S$ . Moreover, both  $\text{conv}(S)$  and  $\text{cone}(S)$  are convex subsets of  $\mathbb{R}^n$ .

By 7.2(b) every pointed polyhedron  $P$  has a finite number of extreme points. Let

$$S = \{\mathbf{x}^i \in \mathbb{R}^n : \mathbf{x}^i \text{ is an extreme point of } P(H, \mathbf{h}) \text{ for } i = 1, \dots, q\}.$$

By the convexity of  $P$ ,  $\text{conv}(S) \subseteq P$ . If  $P \subseteq \{\mathbf{x} \in \mathbb{R}^n : -K \leq x_j \leq K \text{ for } j = 1, \dots, n\}$  for some  $0 < K < +\infty$  then  $P$  is called a **polytope** to distinguish it from an unbounded polyhedron.

**7.3(b)** Let  $S$  be the set of extreme points of a polyhedron  $P$ . Then  $\text{conv}(S) = P$  if and only if  $P$  is a polytope. A polytope  $P$  of dimension  $d$  has exactly  $d+1$  affinely independent extreme points.

**7.3(c) (Carathéodory's Theorem)** A polyhedron  $P$  of dimension  $d$  is a polytope if and only if every  $\mathbf{x} \in P$  is the convex combination of at most  $d+1$  affinely independent extreme points of  $P$ .

**7.3(d) (Minkowski's Theorem)** Let  $L_P$  be the lineality space of a polyhedron  $P$ ,  $P^0 = P \cap L_P^\perp$ ,  $S = \{x^1, \dots, x^q\}$  the extreme points and  $T = \{y^1, \dots, y^r\}$  the extremal directions of  $P^0$ . Then  $P^0 = \text{conv}(S) + \text{cone}(T)$  and  $P = L_P + \text{conv}(S) + \text{cone}(T)$ .

**7.3(e)** Let  $d_0 = \dim P^0$ . Every  $x \in P^0$  can be written as  $x = \sum_{i \in I} \mu_i x^i + \sum_{j \in J} \lambda_j y^j$  where  $\mu_i \geq 0$  for all  $i \in I \subseteq \{1, \dots, q\}$ ,  $\sum_{i \in I} \mu_i = 1$ ,  $\lambda_j \geq 0$  for all  $j \in J \subseteq \{1, \dots, r\}$ , the extreme points  $x^i$  for  $i \in I$  are affinely independent,  $|I| \leq d_0 + 1$ , the extremal directions  $y^j$  for  $j \in J$  are linearly independent and  $|J| \leq d_0$ .

### 7.3.1 Displaced Cones, Adjacency II, Images of Polyhedra

For a given extreme point  $x^0 \in P^0$  denote by  $(H^0, h^0)$  the submatrix of  $(H^<, h^<)$  such that  $H^0 x^0 = h^0$  where we have partitioned  $Hx \leq h$  as done in (7.1). The set

$$C(x^0, H) = \{x \in \mathbb{R}^n : H^= x = h^=, H^0 x \leq h^0, Gx = 0\} \quad (7.6)$$

is a polyhedron having precisely one extreme point, i.e.  $x^0$ , and satisfies  $\dim C(x^0, H) = d_0 = \dim P^0$ .

**7.3(f)** Every extreme point of  $P^0$  is contained in at least  $d_0$  distinct edges of  $P^0$ . Every extreme point of a polytope of dimension  $d$  has at least  $d$  affinely independent adjacent extreme points.

By the translation  $y = x - x^0$  the polyhedron  $C(x^0, H)$  goes over into a polyhedral cone  $CC(x^0, H) = \{y \in \mathbb{R}^n : H^= y = 0, H^0 y \leq 0, Gy = 0\} \supseteq C_\infty^0$ .  $C(x^0, H)$  is the *displaced asymptotic cone* of  $P^0$  at  $x^0$  and  $x^0$  is its **apex**.

Let  $z = f + Lx$  be an *affine transformation* that maps  $\mathbb{R}^n$  into  $\mathbb{R}^p$  with  $1 \leq p \leq n$ . If  $f = 0$  then  $z = Lx$  is a *linear transformation*, e.g. an orthogonal projection from  $\mathbb{R}^n$  onto some subspace of it. We assume  $r(L) = p$ , i.e., that the transformation is of full rank. Let

$$P = \{z \in \mathbb{R}^p : \exists x \in Q \text{ such that } z = f + Lx\} \quad (7.7)$$

be the image of some polyhedron  $Q = Q(H, h) \subseteq \mathbb{R}^n$ .

**7.3(g)** The image of a polyhedron under an affine transformation (of full rank) is a polyhedron.

A linear description of  $P$  is obtained as follows: Since  $r(L) = p$  partition  $L$  into two parts  $L_1$  and  $L_2$  such that  $L_1$  is of size  $p \times p$  and nonsingular. WROG we will assume that  $L = (L_1 \ L_2)$ . Partition  $x \in \mathbb{R}^n$  accordingly into  $x_1$  and  $x_2$  and thus  $z = f + L_1 x_1 + L_2 x_2$ . Let the matrix  $(H, h)$  defining  $Q$  be partitioned into  $(H^=, h^=)$  and  $(H^<, h^<)$  like in (7.1). According to the partition of  $L$  we partition  $H^=$  into  $H_1^=$  and  $H_2^=$ ,  $H^<$  likewise into  $H_1^<$  and  $H_2^<$ . We index the rows of  $(H^<, h^<)$  by  $1, \dots, m^<$  and those of  $(H^=, h^=)$  by  $m^< + 1, \dots, m^< + m^=$ . In what follows  $u \in \mathbb{R}^{m^<}$  and  $v \in \mathbb{R}^{m^=}$ . Let

$$C = \{(u, v) \in \mathbb{R}^{m^<+m^=} : u(H_2^< - H_1^< L_1^{-1} L_2) + v(H_2^= - H_1^= L_1^{-1} L_2) = 0, u \geq 0\}, \quad (7.8)$$

and  $(0, v^i)$  for  $i = 1, \dots, s$  denote the basis of the lineality space  $L_C$  of  $C$ . Let  $(u^i, v^i)$  for  $i = s+1, \dots, t$  denote the direction vectors of the extreme rays of the pointed cone  $C^0 = C \cap L_C^\perp$ . Then

$$P = \{z \in \mathbb{R}^p : z \text{ satisfies } (**)\}, \quad (7.9)$$

where

$$\begin{aligned} \mathbf{v}^i \mathbf{H}_1^= \mathbf{L}_1^{-1} \mathbf{z} &= \mathbf{v}^i \mathbf{h}^= + \mathbf{v}^i \mathbf{H}_1^= \mathbf{L}_1^{-1} \mathbf{f} \quad \text{for } i = 1, \dots, s, \\ (\mathbf{u}^i \mathbf{H}_1^< + \mathbf{v}^i \mathbf{H}_1^=) \mathbf{L}_1^{-1} \mathbf{z} &\leq \mathbf{u}^i \mathbf{h}^< + \mathbf{v}^i \mathbf{H}^< + (\mathbf{u}^i \mathbf{H}_1^< + \mathbf{v}^i \mathbf{H}_1^=) \mathbf{L}_1^{-1} \mathbf{f} \quad \text{for } i = s+1, \dots, t. \end{aligned} \quad (**)$$

The assumption that  $r(\mathbf{L}) = p$  is made for notational convenience only. See the text for the general case.

### 7.3.2 Carathéodory, Minkowski, Weyl

**Definition P2** A set  $P \subseteq \mathbb{R}^n$  is a polyhedron if and only if there exists a finite set  $S = \{\mathbf{x}^1, \dots, \mathbf{x}^q\}$  and a finite set  $T = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$  such that  $P = \text{conv}(S) + \text{cone}(T)$  where  $\mathbf{x}^i \in \mathbb{R}^n$  for  $i = 1, \dots, q$  and  $\mathbf{y}^j \in \mathbb{R}^n$  for  $j = 1, \dots, r$ . The pair of sets  $(S, T)$  is a finite generator (or a pointwise description) of  $P$ .

According to Definition P2 a polyhedron is the set of points of  $\mathbb{R}^n$  that can be written as the convex combinations of some finite set  $S$  of points plus the nonnegative combinations of some other finite set  $T$  of direction vectors. We write  $P = P(S, T)$  to denote the polyhedron with generator  $(S, T)$ .

Let  $P = P(\mathbf{H}, \mathbf{h})$ ,  $\mathbf{g}^1, \dots, \mathbf{g}^s \in \mathbb{R}^n$  be a basis of its lineality space  $L_P$ ,  $\mathbf{x}^1, \dots, \mathbf{x}^q$  be the extreme points and  $\mathbf{y}^1, \dots, \mathbf{y}^r$  the extremal directions of  $P^0 = P \cap L_P^\perp$ . By point 7.3(d)  $P = \text{conv}(S^0) + \text{cone}(T^0)$ , where

$$S^0 = \{\mathbf{x}^1, \dots, \mathbf{x}^q\}, \quad T^0 = \{\mathbf{g}^1, \dots, \mathbf{g}^s, -\mathbf{g}^1, \dots, -\mathbf{g}^s, \mathbf{y}^1, \dots, \mathbf{y}^r\}, \quad (7.10)$$

and thus  $P$  is a polyhedron according to Definition P2. On the other hand, let  $P = \text{conv}(S) + \text{cone}(T)$  be a polyhedron according to P2 where  $S = \{\mathbf{x}^1, \dots, \mathbf{x}^q\}$  and  $T = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$  are some finite sets of points of  $\mathbb{R}^n$ . Denote by  $X = (\mathbf{x}^1 \dots \mathbf{x}^q)$ ,  $Y = (\mathbf{y}^1 \dots \mathbf{y}^r)$  the matrices of size  $n \times q$ , of size  $n \times r$ , respectively, corresponding to  $S$  and  $T$ , respectively. Every  $\mathbf{x} \in P$  can be written as  $\mathbf{x} = X\boldsymbol{\mu} + Y\boldsymbol{\lambda}$  with  $\boldsymbol{\mu} \in \mathbb{R}^q$ ,  $\boldsymbol{\mu} \geq 0$ ,  $e\boldsymbol{\mu} = 1$  and  $\boldsymbol{\lambda} \in \mathbb{R}^r$ ,  $\boldsymbol{\lambda} \geq 0$  where  $e$  is the row vector of  $q$  ones. Thus  $P$  is the projection of

$$Q = \{(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \in \mathbb{R}^{n+q+r} : \mathbf{x} = X\boldsymbol{\mu} + Y\boldsymbol{\lambda}, e\boldsymbol{\mu} = 1, \boldsymbol{\mu} \geq 0, \boldsymbol{\lambda} \geq 0\}. \quad (7.11)$$

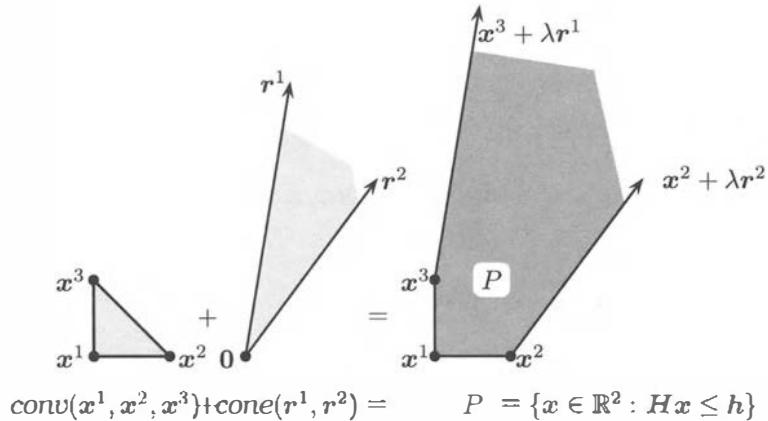
from  $\mathbb{R}^{n+q+r}$  onto  $\mathbb{R}^n$  and hence by point 7.3(g) a polyhedron according to Definition P1.

**7.3(h) (Weyl's Theorem)** Definitions P1 and P2 are equivalent definitions of a polyhedron in  $\mathbb{R}^n$ .

There are thus two distinctively different ways of describing a polyhedron in a finite-dimensional vector space over the reals. If  $P \subseteq \mathbb{R}^n$  is given by a linear description, i.e.  $P = P(\mathbf{H}, \mathbf{h})$ , then there exist finite sets  $S, T \subseteq \mathbb{R}^n$  such that  $P = P(S, T)$  and vice versa, if  $P$  is given by a pointwise description, i.e.  $P = P(S, T)$ , then there exists an  $m \times n$  matrix  $\mathbf{H}$  and a vector  $\mathbf{h} \in \mathbb{R}^m$  such that  $P = P(\mathbf{H}, \mathbf{h})$  and  $0 \leq m < \infty$ .

### 7.3.3 Minimal Generators, Canonical Generators, Quasi-Uniqueness

A finite generator  $(S, T)$  of a polyhedron  $P = P(S, T)$  is **minimal** if deleting anyone of the points in  $S$  or  $T$  alters the polyhedron. If  $P$  is line free then the set  $S$  of the extreme points and the set  $T$  of extremal directions of  $P$  form a minimal generator of  $P$  and vice versa, if  $(S, T)$  is a minimal generator of a line free polyhedron  $P$  then  $S$  is the set of extreme points and – in some scaling –



**Fig. 7.2.** Weyl's Theorem in  $\mathbb{R}^2$

$T$  is the set of extremal directions of  $P$ . Up to scaling the points in  $T$ , the minimal generator of a line free polyhedron is **unique**.

**7.3(i)** Let  $S = \{\mathbf{x}^1, \dots, \mathbf{x}^r\}$ ,  $T = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$ ,  $P = \text{conv}(S) + \text{cone}(T)$ ,  $\mathbf{x} \in P$  and  $\mathbf{y} \in \mathbb{R}^n$ ,  $\mathbf{y} \neq \mathbf{0}$ .  $P$  contains the halfline  $\mathbf{x} + (\mathbf{y})$  if and only if  $\mathbf{y} \in \text{cone}(T)$ .

If  $Hx \leq h$  is a linear description of the polyhedron  $P = P(S, T)$  then  $P = P(H, h)$ ,  $\text{cone}(T) = C_\infty(H)$  and the lineality space  $L_P$  of  $P$  satisfies  $L_P = \text{cone}(T) \cap \text{cone}(-T)$ , where  $-T = \{\mathbf{y} \in \mathbb{R}^n : -\mathbf{y} \in T\}$ . We can thus define the notions of the lineality space and of the asymptotic cone of any polyhedron  $P = P(S, T)$  in terms of the set  $T$  only, i.e. independently of the linear description of  $P$ .

**7.3(j)** With the same notation as under 7.3(i) let  $\mathbf{Y} = (\mathbf{y}^1 \dots \mathbf{y}^r)$ . The polyhedron  $P = P(S, T)$  is line free if and only if  $T = \emptyset$  or  $\{\lambda \in \mathbb{R}^r : \mathbf{Y}\lambda = \mathbf{0}, \lambda \geq \mathbf{0}\} = \{\mathbf{0}\}$ . Equivalently,  $P$  is line free if and only if  $\text{cone}(T)$  is line free.

**7.3(k)** With the same notation as under 7.3(i), the set  $T$  can be partitioned into  $T_0$  and  $T_1$  such that every point of  $T_0$  defines a line of  $P$  and such that  $P^1 = \text{conv}(S) + \text{cone}(T_1)$  is line free. Moreover, every maximal subset of linearly independent points in  $T_0$  is a basis of the lineality space  $L_P$  of  $P$ .

**7.3(l)** Given any finite generator  $(S, T)$  of a polyhedron  $P = P(S, T)$  a minimal generator  $(S^*, T_0 \cup T_1)$  of  $P$  can be found in a finite number of steps. If  $T^* \subseteq T$  is any subset of points of  $T$  such that  $\text{cone}(T - T^*)$  is line free and every point in  $T^*$  defines a line of  $P$ , then  $T_0 \subseteq T^*$  is any maximal subset of linearly independent vectors in  $T^*$ .  $T_1 \subseteq T - T^*$  is the subset of points in  $T - T^*$  that define distinct extreme rays of  $\text{cone}(T - T^*)$ .  $S^* \subseteq S$  is the subset of the extreme points of  $\text{conv}(S)$  that are distinct extreme points of the polyhedron  $P^1 = \text{conv}(S) + \text{cone}(T_1)$ .

Like any list of inequalities  $Hx \leq h$  defining  $P(H, h)$  can be reduced to a minimal set of valid equations and a minimal set of facet-defining inequalities, any finite generator  $(S, T)$  of  $P(S, T)$  can be reduced to its essential elements: it consists of a basis of the lineality space  $L_P$  of  $P$ , the extreme points and the extremal directions of an associated polyhedron  $P^1$ .

Minimal generators of polyhedra are not unique: see Exercise 7.6 (ii) and (iii). Among all generators of a polyhedron  $P \subseteq \mathbb{R}^n$  we distinguish the **canonical generator**  $(S^0, T^0)$  of  $P$  given by point 7.3(d), see (7.10), which is unique *modulo* the choice of a basis of the lineality space  $L_P$  and the scaling of the direction vectors  $\mathbf{y}^i \in T^0$  for  $i = 1, \dots, r$ .

**7.3(m)** *The canonical generator  $(S^0, T^0)$  of a polyhedron  $P$  is minimal.*

All minimal generators of a polyhedron  $P \subseteq \mathbb{R}^n$  share some commonalities. e.g. the cardinalities of the respective point sets  $S$  and  $T$  must be the same. Of course, if  $S = \emptyset$  and  $T \neq \emptyset$  then replacing  $S$  by  $\{\mathbf{0}\}$  does not change the polyhedron  $P(S, T)$  which is a polyhedral cone in both cases. So **assume** that  $T \neq \emptyset$  implies  $S \neq \emptyset$  for all generators that we consider. Let  $\pi_P$  be the orthogonal projection of  $\mathbb{R}^n$  onto the orthogonal complement  $L_P^\perp$  of the lineality space  $L_P$  of some polyhedron  $P \subseteq \mathbb{R}^n$  and  $G$  be a  $s \times n$  matrix of a basis of  $L_P$ . Then the linear transformation  $\mathbf{z} = \pi_P \mathbf{x}$  from  $\mathbb{R}^n$  into  $\mathbb{R}^n$  is

$$\mathbf{z} = (\mathbf{I}_n - G^T(GG^T)^{-1}G)\mathbf{x}. \quad (7.12)$$

For  $S \subseteq \mathbb{R}^n$  let  $\pi_P S = \{\mathbf{z} \in \mathbb{R}^n : \exists \mathbf{x} \in S \text{ such that } \mathbf{z} = \pi_P \mathbf{x}\}$  be the image of  $S$  under  $\pi_P$ . Let  $H\mathbf{x} \leq \mathbf{h}$  be a linear description of  $P$ ,  $C_\infty = C_\infty(H)$  be the asymptotic cone of  $P$ ,  $P^0 = P \cap L_P^\perp$  and  $C_\infty^0 = C_\infty \cap L_P^\perp$ . Hence  $\pi_P P = P^0$ ,  $\pi_P C_\infty = C_\infty^0$  and  $\pi_P L_P = \{\mathbf{0}\}$ . If the polyhedron  $P$  is pointed then the matrix  $G$  is vacuous and the transformation  $\pi_P$  is the identity that maps  $\mathbf{x} \in \mathbb{R}^n$  onto itself.

**7.3(n)** *Let  $\mathbf{x} \in P$  and  $\mathbf{y} \in C_\infty$  be arbitrary.  $\pi_P \mathbf{x}$  is an extreme point of  $P^0$  if and only if there exist exactly  $r(H)$  linearly independent rows  $(\mathbf{h}^i, h_i)$  of  $(H, \mathbf{h})$  such that  $\mathbf{h}^i \mathbf{x} = h_i$ . The halfline  $(\pi_P \mathbf{y})$  is an extreme ray of  $C_\infty^0$  if and only if  $\mathbf{y} \notin L_P$  and there exist exactly  $r(H) - 1$  linearly independent rows  $\mathbf{h}^i$  of  $H$  such that  $\mathbf{h}^i \mathbf{y} = 0$ .*

**7.3(o)** *A generator  $(S, T)$  of the polyhedron  $P = P(S, T)$  is minimal if and only if  $(\pi_P S, T_0 \cup -T_0 \cup (\pi_P T - \{\mathbf{0}\}))$  is the canonical generator of  $P$ , where  $T_0$  is a basis of the lineality space  $L_P$  of  $P$  and  $\pi_P$  the transformation (7.12).*

It follows that – *modulo* a basis of the lineality space  $L_P$  of  $P$  – a minimal generator of  $P$  is **quasi-unique**.

## 7.4 Double Description Algorithms

Given a linear description of a polyhedron  $P = P(H, \mathbf{h})$ , homogenize the constraint set of  $P$  and form the polyhedral cone  $HP = HP(H, \mathbf{h})$ , see (7.5). By Exercise 7.3, every minimal generator for  $HP$  furnishes a minimal generator for  $P$  and vice versa.

Given a pointwise description  $(S, T)$  of a polyhedron  $P = P(S, T)$ , let  $X, Y$  be the  $n \times q$  matrix of the points  $\mathbf{x}^1, \dots, \mathbf{x}^q$  of  $S$  and the  $n \times r$  matrix of the points  $\mathbf{y}^1, \dots, \mathbf{y}^r$  of  $T$ , respectively. The cone  $PC = PC(X, Y)$

$$PC(X, Y) = \{(\mathbf{v}, v_0) \in \mathbb{R}^{n+1} : \mathbf{v}X - v_0 \mathbf{e} \leq \mathbf{0}, \mathbf{v}Y \leq \mathbf{0}\}, \quad (7.13)$$

where  $\mathbf{e}$  is a row vector with  $q$  entries equal to one and  $\mathbf{v} \in \mathbb{R}^n$  is a row vector, is (frequently) called the  $v_0$ -*polar* of the polyhedron  $P = P(S, T)$ . Like in point 7.3(g) it follows that

$$P(S, T) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{v}^i \mathbf{x} = v_i \text{ for } i = 1, \dots, s, \mathbf{v}^i \mathbf{x} \leq v_i \text{ for } i = s + 1, \dots, t\}, \quad (7.14)$$

where  $(v^i, v_i)$  for  $i = 1, \dots, s$  form a basis of the lineality space  $L_{PC}$  of  $PC$  and  $(v^i, v_i)$  for  $i = s+1, \dots, t$  are the extreme rays of the associated pointed cone  $PC^0 = PC \cap L_{PC}^\perp$  satisfying  $v^i \neq 0$ . Any minimal generator of  $PC(X, Y)$  can be used in the linear description (7.14) of  $P(S, T)$ . By Exercise 7.8 we get an ideal description of  $P = P(S, T)$ .

The two problems of finding the *other* description of  $P$  from a given description are thus reduced to the problem of finding a minimal generator of some cone. Let  $H$  be any  $m \times n$  matrix of reals and denote by

$$C(H) = \{\mathbf{y} \in \mathbb{R}^n : H\mathbf{y} \leq \mathbf{0}\} \quad (7.15)$$

the polyhedral cone defined by  $H$ . The double description algorithm takes  $m, n$  and the  $m$  rows  $\mathbf{h}^1, \dots, \mathbf{h}^m$  of  $H$  as input. It returns the number  $nb$  of vectors in a basis of the lineality space  $L_C$  of the cone  $C = C(H)$ , a set of vectors  $BL$  that form a basis of  $L_C$ , the number  $nx$  of extreme rays of the associated pointed cone  $C^0 = C \cap L_P^\perp$  and a set of vectors  $EX$  that together with the point set  $BL \cup -BL$  and  $S = \{\mathbf{0}\}$  form a minimal generator for  $C$ . We **assume** that  $\mathbf{h}^i \neq \mathbf{0}$  for  $i = 1, \dots, m$  since rows  $\mathbf{h}^i = \mathbf{0}$  can be deleted from  $H$ . For every value of the counter  $k \geq 1$  of the algorithm we denote the elements of the point set  $BL_{k-1}$  generically by  $\mathbf{b}^1, \dots, \mathbf{b}^{nb}$ , those of  $EX_{k-1}$  by  $\mathbf{y}^1, \dots, \mathbf{y}^{nx}$  and  $NX = \{1, \dots, nx\}$ .

#### Double Description Algorithm ( $m, n, H, nb, BL, nx, EX$ )

**Step 0:** Set  $k := 0$ ,  $nb := n$ ,  $BL_0 := \{\mathbf{b}^1, \dots, \mathbf{b}^{nb}\}$  where  $\mathbf{b}^i \in \mathbb{R}^n$  for  $i = 1, \dots, n$  is the  $i^{th}$  unit vector,  $nx := 0$  and  $EX_0 := \emptyset$ .

**Step 1:** **if**  $k \geq m$  **then**

**stop** “output  $nb$ ,  $BL := BL_m$ ,  $nx$  and  $EX := EX_m$ ”.

**else**

set  $k := k + 1$ ,  $\mathbf{h} := \mathbf{h}^k$  and **go to** Step 2.

**endif.**

**Step 2:** **if**  $nb = 0$  **or**  $\mathbf{h}\mathbf{b}^i = 0$  for  $i = 1, \dots, nb$  **then**

**go to** Step 4.

**else**

let  $j \in \{1, \dots, nb\}$  be the smallest index  $j$  such that  $\mathbf{h}\mathbf{b}^j \neq 0$ ,

set  $\mathbf{b} := \mathbf{b}^j$  if  $\mathbf{h}\mathbf{b}^j < 0$ ,  $\mathbf{b} := -\mathbf{b}^j$  if  $\mathbf{h}\mathbf{b}^j > 0$  and **go to** Step 3.

**endif.**

**Step 3:** Replace  $BL_{k-1}$  by  $BL_k = \{(\mathbf{h}\mathbf{b})\mathbf{b}^i - (\mathbf{h}\mathbf{b}^i)\mathbf{b} : i \in \{1, \dots, nb\} - j\}$ , set  $nb := nb - 1$ , replace  $EX_{k-1}$  by  $EX_k = \{\mathbf{b}\} \cup \{(\mathbf{h}\mathbf{y}^i)\mathbf{b} - (\mathbf{h}\mathbf{b})\mathbf{y}^i : i \in NX\}$ , set  $nx := nx + 1$  and **go to** Step 1.

**Step 4:** Set  $N_0 := \{i \in NX : \mathbf{h}\mathbf{y}^i = 0\}$ ,  $N_+ := \{i \in NX : \mathbf{h}\mathbf{y}^i > 0\}$ ,  $N_- := \{i \in NX : \mathbf{h}\mathbf{y}^i < 0\}$ .

For all  $\ell \in NX$  calculate  $M_\ell := \{i \in \{1, \dots, k-1\} : \mathbf{h}^i\mathbf{y}^\ell = 0\}$  and

$$N^* := \{(i, j) : i \in N_+, j \in N_- \text{ such that } M_i \cap M_j \not\subseteq M_\ell \text{ for all } \ell \in NX - \{i, j\}\}. \quad (7.16)$$

Replace  $EX_{k-1}$  by  $EX_k = \{\mathbf{y}^i : i \in N_0 \cup N_-\} \cup \{|\mathbf{h}\mathbf{y}^j|\mathbf{y}^i + |\mathbf{h}\mathbf{y}^i|\mathbf{y}^j : (i, j) \in N^*\}$ , set  $nx := |N_0| + |N_-| + |N^*|$ , set  $BL_k := BL_{k-1}$  and **go to** Step 1.

#### 7.4.1 Correctness and Finiteness of the Algorithm

Denote by  $C_k = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{h}^i\mathbf{y} \leq 0 \text{ for } i = 1, \dots, k\}$  the polyhedral cone formed by the  $k$  first rows of  $H$ . For  $k = 0$  we define  $\mathbf{h}^0 = \mathbf{0}$  and the cone  $C_0 = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{0}\mathbf{y} \leq 0\}$  is the entire  $\mathbb{R}^n$ . In terms of

the value of the counter  $k$  in Step 2 the algorithm proceeds from a *current* cone  $C_{k-1}$  to the *new* cone  $C_k = C_{k-1} \cap \{\mathbf{y} \in \mathbb{R}^n : \mathbf{h}^k \mathbf{y} \leq 0\}$ .

**7.4(a)** If the double description algorithm executes Step 3, then the pair  $(\{0\}, BL_k \cup -BL_k \cup EX_k)$  is a minimal generator of  $C_k$ .

**7.4(b)** If the double description algorithm executes Step 4, then the pair  $(\{0\}, BL_k \cup -BL_k \cup EX_k)$  is a minimal generator of  $C_k$ .

**7.4(c)** The double description algorithm finds a minimal generator for a polyhedral cone in a finite number of steps.

The test in line (7.16) of Step 4 requires a comparison of each pair  $(i, j)$  with  $i \in N_+$  and  $j \in N_-$  with all other elements of  $NX$ . To carry it out one constructs the “inner product table” consisting of the inner products  $\mathbf{h}^r \mathbf{y}^i$  of all processed rows with all elements of the set  $EX_{k-1}$ , one forms  $M_i \cap M_j$  and scans the table for containment of  $M_i \cap M_j$  in some  $M_\ell$ ,  $\ell \in NX - \{i, j\}$ . The test of line (7.16) can be sharpened; see Exercise 7.9.

#### 7.4.2 Geometry, Euclidean Reduction, Analysis

Geometrically it is clear how the double description algorithm works. It determines minimal generators for a sequence of cones  $C_0, \dots, C_m$  satisfying  $C_0 = \mathbb{R}^n \supseteq C_1 \supseteq \dots \supseteq C_m = C(H)$ , where  $C_k$  is obtained from  $C_{k-1}$  by intersecting  $C_k$  with a single halfspace  $\{\mathbf{y} \in \mathbb{R}^n : \mathbf{h}^k \mathbf{y} \leq 0\}$ .

A rational number  $c$  divides a rational number  $a$  if there exists an integer number  $d$  such that  $a = c \cdot d$ . The greatest common divisor  $g.c.d.(a_1, \dots, a_n)$  of  $n \geq 2$  rational numbers that are not all zero is the greatest rational number that divides each of  $a_1, \dots, a_n$ . The numbers  $a_1, \dots, a_n$  are called *relatively prime* if  $g.c.d.(a_1, \dots, a_n) = 1$ . To find the greatest common divisor of  $n$  rational numbers one uses repeatedly the Euclidean algorithm which determines the *g.c.d.*  $gcd$  of two positive numbers  $a$  and  $b$ . By  $[a]$  we denote the largest integer number less than or equal to  $a$ .

**Euclidean Algorithm** ( $a, b, gcd$ )

**Step 0:** Set  $k := 1$ ,  $a_0 := a$ ,  $b_0 := b$ .

**Step 1:** Set  $a_k := a_{k-1} - \left\lfloor \frac{a_{k-1}}{b_{k-1}} \right\rfloor b_{k-1}$  and **if**  $a_k = 0$  **stop** “output  $gcd := b_{k-1}$ .”

Set  $b_k := b_{k-1} - \left\lfloor \frac{b_{k-1}}{a_k} \right\rfloor a_k$  and **if**  $b_k = 0$  **stop** “output  $gcd := a_k$ .”

Set  $k := k + 1$  and **go to** Step 1.

To control the size of the numbers that define the components of the elements in  $BL_k$  and  $EX_k$  of the double description algorithm one runs the Euclidean algorithm on all elements of those sets and divides each vector by the greatest common divisor of its components. We refer to this process of simplifying vectors of length  $n$  via the Euclidean algorithm as **Euclidean reduction**; see Exercise 7.10.

To prove that the Euclidean algorithm *always* achieves the desired reduction in the size of the numbers for the double description algorithm, like e.g. in Exercise 7.10, we need several determinantal identities.

For any  $a, b, c, d \in \mathbb{R}$  we denote  $\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$ .

**7.4(d)** Let  $J = E = \{1, \dots, k\}$ ,  $\mathbf{H}_J^E = (h_s^r)_{s \in J}^{r \in E}$  and  $k \geq 2$ . Then

$$\begin{aligned} \det \mathbf{H}_J^E &= (-1)^{i+t} \sum_{r=1}^{k-1} \sum_{s=r+1}^k (-1)^{r+s} \begin{vmatrix} h_i^r & h_t^r \\ h_i^s & h_t^s \end{vmatrix} \det \mathbf{H}_{J-\{i,t\}}^{E-\{r,s\}}, \\ &= (-1)^{p+q} \sum_{j=1}^{k-1} \sum_{\ell=j+1}^k (-1)^{j+\ell} \begin{vmatrix} h_j^p & h_\ell^p \\ h_j^q & h_\ell^q \end{vmatrix} \det \mathbf{H}_{J-\{j,\ell\}}^{E-\{p,q\}}, \end{aligned}$$

respectively, where  $1 \leq i < t \leq k$ ,  $1 \leq p < q \leq k$  and by convention  $\det \mathbf{H}_\emptyset^\emptyset = 1$ .

**7.4(e)** With the above notation we have for all  $1 \leq p < q \leq k$  and  $1 \leq i < t \leq k$  that

$$\det \mathbf{H}_{J-i}^{E-p} \det \mathbf{H}_{J-t}^{E-q} - \det \mathbf{H}_{J-t}^{E-p} \det \mathbf{H}_{J-i}^{E-q} = \det \mathbf{H}_J^E \det \mathbf{H}_{J-\{i,t\}}^{E-\{p,q\}}.$$

Let  $\mathbf{h}_i \in \mathbb{R}^k$  be column  $i$  of the  $k \times k$  matrix  $\mathbf{H}_J^E$  where  $1 \leq i \leq k$ . Consider any column vector  $\mathbf{h}_\ell \in \mathbb{R}^k$ , say, where  $\ell > k$  is arbitrary. Then

$$\mathbf{H}_{J-i+\ell}^E = (\mathbf{h}_1, \dots, \mathbf{h}_{i-1}, \mathbf{h}_{i+1}, \dots, \mathbf{h}_k, \mathbf{h}_\ell)$$

is the matrix obtained from  $\mathbf{H}_J^E$  by dropping  $\mathbf{h}_i$  and adding  $\mathbf{h}_\ell$  as the last column where  $1 \leq i \leq k$ .

**7.4(f)** Like above let  $\ell > k$  and  $\mathbf{h}_\ell \in \mathbb{R}^k$  be arbitrary. Then for all  $1 \leq i < t \leq k$  and  $1 \leq p \leq k$

$$\det \mathbf{H}_{J-t+\ell}^E \det \mathbf{H}_{J-i}^{E-p} - \det \mathbf{H}_{J-i+\ell}^E \det \mathbf{H}_{J-t}^{E-p} = \det \mathbf{H}_J^E \det \mathbf{H}_{J-\{i,t\}+\ell}^{E-p}.$$

### 7.4.3 The Basis Algorithm and All-Integer Inversion

In the modified double description algorithm or MDDA, for short, the computation is organized somewhat differently from the original algorithm. The part of the algorithm that finds a basis of the lineality space  $L$  of the cone  $C(\mathbf{H})$  is separated out. We use the same notational conventions as in the original algorithm.

**Basis Algorithm**( $m, n, \mathbf{H}, nb, BL, nx, EX, E, J, Det$ )

**Step 0:** Set  $k := 0$ ,  $nb := n$ ,  $BL_0 = \{\mathbf{b}^1, \dots, \mathbf{b}^{nb}\}$  where  $\mathbf{b}^i \in \mathbb{R}^n$  is the  $i^{th}$  unit vector,  $nx := 0$ ,  $EX_0 := \emptyset$ ,  $d_0 := 1$ ,  $p := 0$ ,  $E_0 := \emptyset$ ,  $J_0 := \emptyset$ ,  $R_0 := \{1, \dots, n\}$ ,  $A_0 = \{1, \dots, m\}$ .

**Step 1:** Set  $k := k + 1$ ,  $\mathbf{h} := \mathbf{h}^r$  for some  $r \in A_{k-1}$  and  $A_k := A_{k-1} - r$ .

**if**  $\mathbf{h}\mathbf{b}^i = 0$  for  $i = 1, \dots, nb$  **then**

**if**  $k \geq m$  **then go to** Step 3 **else go to** Step 1.

**else**

let  $j \in \{1, \dots, nb\}$  be the smallest index such that  $\mathbf{h}\mathbf{b}^j \neq 0$ ,

set  $\mathbf{b} := \mathbf{b}^j$ ,  $\sigma := 1$  if  $\mathbf{h}\mathbf{b} < 0$ ,  $\sigma := -1$  if  $\mathbf{h}\mathbf{b} > 0$ ,

set  $p := p + 1$ ,  $d_p := \mathbf{h}\mathbf{b}$ ,  $E_p := E_{p-1} + r$ ,  $J_p := J_{p-1} + \ell$  where  $\ell \in R_{p-1}$  is the smallest index such that  $b_\ell^j \neq 0$ ,  $R_p := R_{p-1} - \ell$  and **go to** Step 2.

**endif.**

- Step 2:** Replace  $BL_{p-1}$  by  $BL_p = \{(d_p b^i - (\mathbf{h} b^i) b) / d_{p-1} : i \in \{1, \dots, nb\} - j\}$ , set  $nb := nb - 1$ , replace  $EX_{p-1}$  by  $EX_p = \{\sigma b\} \cup \{\sigma((\mathbf{h} y^i) b - d_p y^i) / |d_{p-1}| : i \in \{1, \dots, nx\}\}$ , set  $nx := nx + 1$ .  
**if**  $nb = 0$  **then go to** Step 3 **else go to** Step 1.
- Step 3:** **stop** “output  $nb$ ,  $BL := BL_p$ ,  $nx$ ,  $EX := EX_p$ ,  $E := E_p$ ,  $J := J_p$ ,  $Det := d_p$ ”.

**7.4(g)** The basis algorithm finds the rank  $r(\mathbf{H}) = n - nb = nx$  of  $\mathbf{H}$  and a row set  $E$  and column set  $J$  with  $|E| = |J| = r(\mathbf{H})$  and  $Det = \det \mathbf{H}_J^E \neq 0$ . The points  $b^r \in BL$  for  $1 \leq r \leq nb$  form a basis of the linearity space  $L$  of  $C(\mathbf{H})$  and satisfy  $b_{j_i}^r = (-1)^{p+i+1} \det \mathbf{H}_{J-j_i+\ell_r}^E$  for  $1 \leq i \leq nx$ ,  $b_{\ell_r}^r = \det \mathbf{H}_J^E$ ,  $b_j^r = 0$  otherwise. The points  $y^r \in EX$  for  $1 \leq r \leq nx$  together with the points in  $BL$  form a basis of  $\mathbb{R}^n$  and a minimal generator for the cone  $\mathbf{H}^E y \leq 0$ . Moreover, they satisfy  $y_{j_i}^r = \sigma(-1)^{r+i} \det \mathbf{H}_{J-j_i}^{E-i_r}$  for  $1 \leq i \leq nx$ ,  $y_j^r = 0$  otherwise, where  $\sigma = -\text{sign } \det \mathbf{H}_J^E$ .

The basis algorithm calculates the inverse of the matrix  $\mathbf{H}_J^E$  since  $\mathbf{Y}^T \mathbf{H}_J^E = \sigma \det \mathbf{H}_J^E \mathbf{I}_p$  where  $\mathbf{Y} = (y_{j_i}^r)_{i=1, \dots, p}^{r=1, \dots, p}$ . It is an “all-integer” **inversion routine** for any square integer matrix of full rank since all divisions that the algorithm performs have a remainder of zero and the determinant of a matrix of integers is itself an integer number. If the input matrix is not of full rank then the algorithm returns the rank of the matrix along with a submatrix of maximal rank and its inverse.

#### 7.4.4 An All-Integer Algorithm for Double Description

**Modified Double Description Algorithm**( $m, n, \mathbf{H}, nb, BL, nx, EX$ )

**Step 0:** Run the basis algorithm with the matrix  $\mathbf{H}$  as input, i.e.

**call** Basis Algorithm( $m, n, \mathbf{H}, nb, BL, nx, EX, E, J, Det$ ).

Set  $k := 0$ ,  $m_a := m - nx$ ,  $p := nx$ ,  $A_p := \{1, \dots, m\} - E$ ,  $E_p := E$ ,  $BL_p := BL$ ,  $EX_p := EX$ ,  $NX := \{1, \dots, nx\}$  and **go to** Step 1.

**Step 1:** **if**  $k \geq m_a$  **then**

**stop** “output  $nb$ ,  $BL := BL_p$ ,  $nx$  and  $EX := EX_p$ .”

**else**

set  $k := k + 1$ ,  $\mathbf{h} := h^r$  for some  $r \in A_p$  and **go to** Step 2.

**endif**

**Step 2:** Set  $N_0 := \{i \in NX : \mathbf{h} y^i = 0\}$ ,  $N_+ := \{i \in NX : \mathbf{h} y^i > 0\}$ ,  $N_- := \{i \in NX : \mathbf{h} y^i < 0\}$ .

For all  $\ell \in NX$  calculate  $M_\ell = \{i \in E_p : \mathbf{h}^i y^\ell = 0\}$  and then the set  $N^*$  defined by (7.16).

For every  $(i, j) \in N^*$  find  $s \in M_i - M_j$  and  $t \in M_j - M_i$ , calculate  $d_t^s = |\det \mathbf{H}_J^{S \cup \{s, t\}}|$  for some  $S \subseteq M_i \cap M_j$  by running the basis algorithm with the matrix  $\mathbf{H}_J^Q$  as input where  $Q = M_i \cap M_j \cup \{s, t\}$  and set  $y^{ij} := d_t^s (|\mathbf{h} y^i| y^j + |\mathbf{h} y^j| y^i) / (\mathbf{h}^s y^j) (\mathbf{h}^t y^i)$ .

Set  $p := p + 1$ ,  $E_p := E_{p-1} + r$ ,  $A_p := A_{p-1} - r$ , replace  $EX_{p-1}$  by

$EX_p = \{y^i : i \in N_0 \cup N_-\} \cup \{y^{ij} : (i, j) \in N^*\}$ , set  $nx := |N_0| + |N_-| + |N^*|$ , set  $BL_p := BL_{p-1}$  and **go to** Step 1.

In the statement of MDDA it is assumed that the basis algorithm “returns” to the calling program so that the respective scalars and arrays such as  $nb$ ,  $BL$ ,  $nx$ ,  $EX$ , etc. are all properly set in the subroutine.

**7.4(h)** MDDA finds a minimal generator for the polyhedral cone  $C(\mathbf{H})$  such that every nonzero component of the vectors in the respective point sets equals the determinant of some submatrix of  $\mathbf{H}$  in absolute value.

If all elements of  $H$  are integer numbers then by carrying out first the multiplication by  $d_t^s$  and subsequently the division by the product  $(h^s y^j)(h^t y^i)$  we get an integer number in the calculation of each component of  $y^{ij}$  in Step 2 so that only divisions with a remainder of zero are performed.

Applying Euclidean reduction generally brings about a *stronger reduction* in the size of numbers produced by the original double description algorithm than the remainderless divisions of the MDDA.

Returning to the problems posed at the beginning of Chapter 7.4, suppose first that a linear description  $Hx \leq h$  of  $P \subseteq \mathbb{R}^n$  is given. We form the polyhedral cone  $HP \subseteq \mathbb{R}^{n+1}$ , see (7.5), and execute the MDDA to find a minimal generator

$$(\{\mathbf{0}\}, BL_{HP} \cup -BL_{HP} \cup EX_{HP})$$

of  $HP$ , where  $BL_{HP}$  is a basis of the lineality space  $L_{HP}$  of  $HP$ . Let

$$S = \{x_{n+1}^{-1} \mathbf{x} : (\mathbf{x}, x_{n+1}) \in EX_{HP}, x_{n+1} > 0\}, \quad T = BL_{HP} \cup -BL_{HP} \cup (EX_{HP} - S).$$

It follows that  $(S, T)$  is a minimal generator of  $P = P(H, h)$ .

**7.4(i)** Given a linear description  $Hx \leq h$  of a polyhedron  $P = P(H, h) \subseteq \mathbb{R}^n$  the MDDA applied to  $HP$  determines a minimal generator  $(S, T)$  of  $P$  such that every nonzero component of a point in  $S$  is the ratio of the determinant of some submatrix of  $(H, h)$  and the determinant of some submatrix of  $H$  and every nonzero component of a point in  $T$  equals the determinant of some submatrix of  $H$ .

Suppose next that a finite generator  $(S, T)$  of some polyhedron  $P \subseteq \mathbb{R}^n$  is given. We form the polyhedral cone  $PC = PC(X, Y)$ , see (7.13), and execute the MDDA to find a minimal generator

$$(\{\mathbf{0}\}, BL_{PC} \cup -BL_{PC} \cup EX_{PC})$$

of  $PC$ , where  $BL_{PC}$  is a basis of the lineality space  $L_{PC}$  of  $PC$ . Let  $(v^i, v_i)$  for  $i = 1, \dots, s$  be the points in  $BL_{PC}$  and  $(v^i, v_i)$  for  $i = s+1, \dots, t$  be the points in  $EX_{PC}$  satisfying  $v^i \neq \mathbf{0}$ . Then like in (7.14) we obtain an ideal description of  $P = P(S, T)$ .

**7.4(j)** Given a finite generator  $(S, T)$  of a polyhedron  $P = P(S, T) \subseteq \mathbb{R}^n$  the MDDA applied to  $PC$  determines an ideal description of  $P$  such that every nonzero component of an equation and/or inequality of the ideal description of  $P$  equals the determinant of some submatrix of  $\begin{pmatrix} X & Y \\ -e & 0 \end{pmatrix}$ .

## 7.5 Digital Sizes of Rational Polyhedra and Linear Optimization

A polyhedron  $P \subseteq \mathbb{R}^n$  is a *rational polyhedron* if there exists a linear description  $Hx \leq h$  of  $P$  such that all elements of the matrix  $(H, h)$  are rational numbers. By point 7.4(i), every rational polyhedron has a minimal generator  $(S, T)$ , such that every point in  $S \cup T$  has rational components only.

There are two different ways to measure the “size” of a polyhedron  $P$  where by “size” we mean – roughly – the amount of data necessary to represent the polyhedron on a digital computer. The first measure is the number of bits that are necessary to store all data of an ideal description of

$P$ , the second one is the number of bits necessary to store all data of a minimal generator of  $P$ . So as a first approximation to the “size” of a polyhedron  $P$  we can take the *cardinality* of an ideal description which we denote by  $|P|_\ell$  and the cardinality of a minimal generator of  $P$  which we denote by  $|P|_p$ . Neither measure is appropriate as can be seen from the examples of Exercise 7.2 and Exercise 7.7; see also Exercise 7.13.

To optimize a linear function over  $P$  we need, however, *some* elements of a linear description of  $P$  or *some* elements of a finite generator of  $P$  and thus we are led to consider the digital size of the *individual* elements in either set. By estimating the “largest” digital size of any such element and multiplying by  $|P|_\ell$  and  $|P|_p$ , respectively, one gets an upper bound on the digital sizes required to represent the entire polyhedron on a digital computer as well – which is, however, what we hope to avoid anyway.

### 7.5.1 Facet Complexity, Vertex Complexity, Complexity of Inversion

$\langle a \rangle = 1 + \lceil \log_2(1 + |a|) \rceil$  bits are required to store an integer number  $a$  and its sign on a digital computer.  $\langle a \rangle$  is the *encoding length* or the *digital size* of  $a$ . Any rational number  $r$  can be written as the ratio of two integer numbers  $p/q$ , say, with  $q > 0$  and the denominator and numerator are stored separately so that  $\langle r \rangle = \langle p \rangle + \langle q \rangle$  is the digital size of  $r$ .

Let  $\mathbf{x} \in \mathbb{R}^n$  be any vector with rational components. The digital size of  $\mathbf{x}$  is the sum of the digital sizes of its components, i.e.  $\langle \mathbf{x} \rangle = \sum_{i=1}^n \langle x_i \rangle$ . The Euclidean norm  $\|\mathbf{x}\|$  of  $\mathbf{x}$  satisfies

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\| \leq \|\mathbf{x}\|_1 \leq 2^{\langle \mathbf{x} \rangle - n} - 1. \quad (7.17)$$

The digital size of an  $m \times n$  matrix  $\mathbf{H}$  of rational numbers is the sum of the digital sizes of its elements, i.e.  $\langle \mathbf{H} \rangle = \sum_{i=1}^m \sum_{j=1}^n \langle h_{ij} \rangle$ . The determinant and the inverse of  $\mathbf{H}$  satisfy

$$\langle \det \mathbf{H} \rangle \leq 2\langle \mathbf{H} \rangle - n^2, \quad \langle \mathbf{H}^{-1} \rangle \leq 4n^2\langle \mathbf{H} \rangle. \quad (7.18)$$

The digital size of an inequality  $\mathbf{f}\mathbf{x} \leq f_0$  with rational  $\mathbf{f}$  and  $f_0$  is defined as  $\langle \mathbf{f} \rangle + \langle f_0 \rangle$ .

**Definition FC** Let  $P$  be a rational polyhedron in  $\mathbb{R}^n$  and  $\phi, \nu$  be positive integers.

- (i)  $P$  has a *facet complexity* of  $\phi$  if  $\phi \geq n+1$  and there exists a linear description  $\mathbf{H}\mathbf{x} \leq \mathbf{h}$  of  $P$  such that  $\langle \mathbf{h}^i \rangle + \langle h_i \rangle \leq \phi$  for each row  $(\mathbf{h}^i, h_i)$  of  $(\mathbf{H}, \mathbf{h})$ .
- (ii)  $P$  has a *vertex complexity* of  $\nu$  if  $\nu \geq n$  and there exists a finite generator  $(S, T)$  of  $P$  such that  $\langle \mathbf{x} \rangle \leq \nu$  for each  $\mathbf{x} \in S \cup T$ .

So if  $|P|_\ell$  is the cardinality of a linear description of a polyhedron  $P$  of facet complexity  $\phi$  then at most  $\phi|P|_\ell$  bits are required to represent  $P$  on a digital computer and likewise, if  $|P|_p$  is the cardinality of a finite generator of vertex complexity  $\nu$  of  $P$  then we need at most  $\nu|P|_p$  bits.

When discussing the “polynomiality” of algorithms for problems with rational data, we can rid ourselves of the *rationality* of the data. Whenever it is convenient we can assume that the data of e.g. a linear description of a rational polyhedron are all *integers*. The vertex and facet complexity of rational polyhedra are interrelated.

**7.5(a)** If a polyhedron  $P \subseteq \mathbb{R}^n$  has a facet complexity of  $\phi$ , then its vertex complexity is  $4n^2\phi$ .  
If  $P$  has a vertex complexity of  $\nu$ , then its facet complexity is  $4n^2\nu$ .

Finding e.g. the rank of a rational matrix and calculating its inverse –if nonsingular– and its determinant are tasks that can be executed in polynomial time on a digital computer having a maximal wordsize of  $\max\{16n\phi, 2\phi + 4n^2\phi\}$ . Finding a basis of the lineality space of a rational polyhedral cone requires time that is polynomial in the digital size of the input.

### 7.5.2 Polyhedra and Related Polytopes for Linear Optimization

**7.5(b)** If  $P$  is a nonempty polyhedron of facet complexity  $\phi$  then  $P$  has a minimal generator  $(S, T)$  such that every  $x \in S \cup T$  is a rational number satisfying  $\langle x \rangle < 4n^2\phi$ ,  $-2^{4n\phi} < x_j < 2^{4n\phi}$  and if  $x_j \neq 0$  then  $|x_j| = p_j/q_j$  with integer  $p_j, q_j$ ,  $1 \leq p_j < 2^{4n\phi}$ ,  $1 \leq q_j < 2^{4n\phi}$  and  $|x_j| > 2^{-4n\phi}$  for  $1 \leq j \leq n$  where  $x^T = (x_1, \dots, x_n)$ .

These facts permit to derive bounds that are polynomial in  $n$  and  $\phi$  for the linear program or the “linear optimization problem”

$$\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\},$$

where  $P \subseteq \mathbb{R}^n$  is a rational polyhedron and WROG  $\mathbf{c} \in \mathbb{R}^n$  is integer. Since  $P$  has a finite linear description  $H\mathbf{x} \leq \mathbf{h}$ , say, this problem either has no solution, i.e.  $P = \emptyset$ , or it is unbounded or it possesses a finite optimum solution since the number of rows of  $H$  is finite. We define

$$z_P = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$$

and like in Chapter 6 use the conventions that  $z_P = -\infty$  if  $P = \emptyset$  and  $z_P = +\infty$  if the objective function value  $\mathbf{c}\mathbf{x}$  over  $P$  is not bounded from above.

**7.5(c)** With the above notation, let  $(S, T)$  be any finite generator of some polyhedron  $P \subseteq \mathbb{R}^n$ . Then  $z_P = +\infty$  if and only if there exists  $\mathbf{y} \in T$  such that  $\mathbf{c}\mathbf{y} \neq 0$  if  $(\mathbf{y}) \in P$  and  $(-\mathbf{y}) \in P$ ,  $\mathbf{c}\mathbf{y} > 0$  otherwise. Moreover, if  $P$  has facet complexity  $\phi$  and  $-\infty < z_P < +\infty$ , then  $|z_P| < 2^{\langle \mathbf{c} \rangle + 4n\phi}$  and there exists a rational point  $\mathbf{x}^0 \in S$  such that  $z_P = \mathbf{c}\mathbf{x}^0$  and  $\langle x_j^0 \rangle \leq 4n\phi$  for  $1 \leq j \leq n$ .

Point 7.5(c) can be used in two ways: one consists in reducing the linear program over any rational polyhedron to a linear program over a *rational polytope in the nonnegative orthant*, the other one gives rise to algorithms for linear programming based on *binary search*.

**7.5(d)** Let  $P \subseteq \mathbb{R}^n$  be a polyhedron of facet complexity  $\phi$  and define  $P_\Phi = P \cap \{\mathbf{x} \in \mathbb{R}^n : -2^\Phi \leq x_j \leq 2^\Phi \text{ for } 1 \leq j \leq n\}$  where  $\Phi = \langle \mathbf{c} \rangle + 8n\phi + 2n^2\phi + 2$ . Then  $\dim P = \dim P_\Phi$ ,  $P_\Phi$  has a facet complexity of  $\Phi + 3$  and  $z_P = +\infty$  if and only if  $z_{P_\Phi} \geq 2^{\langle \mathbf{c} \rangle + 4n\phi}$  where  $z_{P_\Phi} = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P_\Phi\}$  and  $\mathbf{c} \in \mathbb{R}^n$  has integer components.

Making the translation  $x'_j = x_j + 2^\Phi$  for  $1 \leq j \leq n$  one gets a polytope  $P'_\Phi$  in the nonnegative orthant of  $\mathbb{R}^n$  from the polytope  $P_\Phi$  of point 7.5(d). Thus all polynomiality issues regarding rational polyhedra can be dealt with within the *restricted class of rational polytopes in the nonnegative orthant*.

### 7.5.3 Feasibility, Binary Search, Linear Optimization

Suppose that we have a subroutine `FINDXZ` ( $P, n, \phi, \Phi, \mathbf{c}, z, \mathbf{x}, \text{FEAS}$ ) that **solves** the following **restricted feasibility problem** for the polyhedron  $P_z^c = P \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{c}\mathbf{x} \geq z\}$ .

**7.5(e)** Given a polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$ , a positive integer  $\Phi$ , a rational row vector  $c \in \mathbb{R}^n$  and a rational number  $z \in \mathbb{R}$  find a rational vector  $x = (x_1, \dots, x_n)^T \in P_z^c$  such that  $\langle x_j \rangle \leq \Phi$  for  $1 \leq j \leq n$  provided such  $x$  exists.

The subroutine (or *procedure* or *oracle*) FINDXZ returns a logical “flag” FEAS and a vector  $x$  of length  $n$ . If  $P_z^c = \emptyset$  then FEAS assumes the value “*false*.” and  $x$  is undefined, while if  $P_z^c \neq \emptyset$  then FEAS assumes the value “*true*.” and  $x$  is a rational vector with the required properties. From point 7.5(d) it follows that for every rational  $z$  satisfying

$$|z| \leq 2^{\langle c \rangle + 4n\phi} \text{ and } \Phi \geq \langle c \rangle + 8n\phi + 2n^2\phi + 2$$

the polyhedron  $P_z^c$  is nonempty if and only if a rational vector  $x \in \mathbb{R}^n$  with the properties required by the feasibility problem exists.

### Binary Search Algorithm 1 ( $P, n, \phi, c$ )

**Step 0:** Set  $k := 0$ ,  $z_L^0 := -2^{\langle c \rangle + 4n\phi + 1}$ ,  $z_U^0 = 2^{\langle c \rangle + 4n\phi + 1}$ ,  $z_U := z_U^0$ ,  $z_L := z_L^0$ ,  $\Phi := \langle c \rangle + 8n\phi + 2n^2\phi + 2$ .

**Step 1:** Set  $k := k + 1$ ,  $z^k := (z_U + z_L)/2$  and call FINDXZ( $P, n, \phi, \Phi, c, z^k, x, \text{FEAS}$ ).

**if** FEAS=*true*. **then**

set  $z_L := z^k$ ,  $z_U := z_U$ ,  $x^0 := x$  and **if**  $z_L \geq 2^{\langle c \rangle + 4n\phi}$  **go to** Step 2.

**else**

set  $z_L := z_L$ ,  $z_U := z^k$ ,  $x^0 := x$  and **if**  $z_U \leq -2^{\langle c \rangle + 4n\phi}$  **go to** Step 2.

**endif.**

**if**  $k \leq \langle c \rangle + n\Phi + 4n\phi + 4n^2\phi + 2$  **go to** Step 1.

**Step 2:** **if**  $z_L = z_L^0$  **stop** “ $P$  is empty and  $z_P := -\infty$ ”.

**if**  $z_U = z_U^0$  **stop** “Problem is unbounded and  $z_P := +\infty$ ”.

**stop** “ $x^0$  is an optimal solution with value  $z_P := cx^0$ ”.

By point 7.5(c) the interval  $[-2^{\langle c \rangle + 4n\phi + 1}, 2^{\langle c \rangle + 4n\phi + 1}]$  contains  $z_P = \max\{cx : x \in P\}$  if it is finite. The binary search algorithm locates  $z_P$  in this “interval of uncertainty” by successively *halving* the interval, testing whether or not one half of the interval can be discarded and thus “narrowing” the interval of uncertainty at a geometrical rate.

**7.5(f)** The binary search algorithm 1 finds the correct answer to the linear optimization problem  $\max\{cx : x \in P\}$  where  $P \subseteq \mathbb{R}^n$  is a polyhedron of facet complexity  $\phi$  and  $c \in \mathbb{R}^n$  is a row vector with integer components. If the running time of the subroutine FINDXZ is bounded by a polynomial in  $n, \phi, \Phi, \langle c \rangle$  and  $\langle z \rangle$ , then the total running time of the algorithm is bounded by a polynomial in  $n, \phi$  and  $\langle c \rangle$ .

The *restricted* feasibility problem 7.5(e) imposes a maximum digital size of  $\Phi$  for each component of the output vector  $x$ . Point 7.5(f) shows that this suffices to locate an optimal solution vector exactly and thus  $z_P$  in the case where  $-\infty < z_P < +\infty$ . Suppose now that subroutine FINDXZ( $P, n, \phi, c, z, x, \text{FEAS}$ ) solves the following (unrestricted) **feasibility problem** for the polyhedron  $P_z^c$ :

**7.5(g)** Given a polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$ , a row vector  $c \in \mathbb{R}^n$  with integer components and a rational number  $z \in \mathbb{R}$  find a rational vector  $x \in P_z^c$  provided such  $x$  exists.

**Binary Search Algorithm 2** ( $P, n, \phi, c$ )

**Step 0:** Set  $k := 0$ ,  $z_L^0 := -2\langle c \rangle + 4n\phi + 1$ ,  $z_U^0 = 2\langle c \rangle + 4n\phi + 1$ ,  $z_U := z_U^0$ ,  $z_L := z_L^0$ .

**Step 1:** Set  $k := k + 1$ ,  $z^k := (z_U + z_L)/2$  and **call** FINDZX( $P, n, \phi, c, z^k, x, \text{FEAS}$ ).

```

if FEAS=.true. then
    set  $z_L := z^k$ ,  $z_U := z_U$ ,  $x^0 := x$  and if  $z_L \geq 2\langle c \rangle + 4n\phi$  go to Step 2.
else
    set  $z_L := z_L$ ,  $z_U := z^k$ ,  $x^0 := x$  and if  $z_U \leq -2\langle c \rangle + 4n\phi$  go to Step 2.
endif.
if  $k \leq \langle c \rangle + 4n\phi + 8n^2\phi + 2$  go to Step 1.

```

**Step 2:** **if**  $z_L = z_L^0$  **stop** “ $P$  is empty and  $z_P := -\infty$ ”.

**if**  $z_U = z_U^0$  **stop** “Problem is unbounded and  $z_P := +\infty$ ”.

**stop** “ $x^0$  is an approximate optimal solution and  $z_P$  is the unique rational number satisfying  $z_L \leq z_P \leq z_U$  and  $z_P = r/s$  with  $r, s$  integer and  $1 \leq s \leq 2^{4n^2\phi}$ .”

The binary search algorithm 2 locates the optimal objective function value  $z_P = \max\{cx : x \in P\}$  uniquely in some finite interval and an approximately optimal solution vector  $x^0$  – if  $z_P$  is finite. Of course, the second binary search algorithm runs “faster” than the first one.

**7.5(h)** *The binary search algorithm 2 is correct. If the running time of the subroutine FINDZX is bounded by a polynomial in  $n, \phi, \langle c \rangle$  and  $\langle z \rangle$ , then the total running time of the algorithm is bounded by a polynomial in  $n, \phi$  and  $\langle c \rangle$ .*

Let  $Hx \leq h$  be a linear description of a polyhedron  $P \subseteq \mathbb{R}^n$ . Binary search reduces the linear optimization problem  $\max\{cx : Hx \leq h\}$  essentially to the problem of proving or disproving the feasibility of polynomially many linear inequality systems  $Hx \leq h, cx \geq z$  where the value of  $z$  varies. For the question of polynomial-time solvability of the linear optimization problem over rational polyhedra this means that we are left with proving the existence of subroutines FINDXZ or FINDZX that run in polynomial time. We come back to this question in Chapter 9.

**7.5.4 Perturbation, Uniqueness, Separation**

**7.5(i)** *Let  $P \subseteq \mathbb{R}^n$  be a nonempty polytope of facet complexity  $\phi$ ,  $c = (c_1, \dots, c_n) \in \mathbb{R}^n$  be a row vector with integer components,  $\Delta \geq 1 + 2^{4n\phi+8n^2\phi+1}$ ,  $d_j = \Delta^n c_j + \Delta^{n-j}$  for  $1 \leq j \leq n$  and  $d = (d_1, \dots, d_n)$ . Then  $\langle d \rangle \leq \langle c \rangle + 2n^2(4n\phi + 8n^2\phi + 3)$  and there exists a unique  $x^0 \in P$  such that  $dx^0 = \max\{dx : x \in P\}$ . Moreover,  $cx^0 = \max\{cx : x \in P\}$  and  $\langle x_j^0 \rangle \leq 4n\phi$  for  $1 \leq j \leq n$ .*

By point 7.5(i) we can assume WROG that the linear optimization problem  $\max\{cx : x \in P\}$  has a unique optimizer if  $P$  is a nonempty polytope since the digital size of the row vector  $d$  is bounded polynomially in  $n, \phi$  and  $\langle c \rangle$  and like  $c$  the vector  $d$  has integer components. The uniqueness of an optimizer helps the theoretical analysis of the linear optimization problem considerably. Dividing  $d$  by  $\Delta^n$  we get componentwise

$$c_j + \Delta^{-j} = c_j + \varepsilon^j \text{ for } 1 \leq j \leq n,$$

where  $\varepsilon = \Delta^{-1} > 0$  is a “small” number. So we have perturbed the objective function  $cx$  of the linear optimization problem in a certain way which permits us to conclude the uniqueness

of the optimizer. The particular **perturbation technique** that we have employed singles out a *lexicographically maximal* point from all candidates for the optimum solution.

When the polyhedron  $P$  contains lines then an optimizer of  $\max\{cx : x \in P\}$  is never unique if the maximum exists at all. Intuitively it is clear that by “some sort of perturbation” one can always achieve the uniqueness of the optimizer of the problem  $\max\{cx : x \in P\}$  when  $P$  is a *pointed polyhedron* rather than a polytope. However, the particular perturbation technique that we have employed in point 7.5(i) is not guaranteed to work; see the text. In the case of an unbounded polyhedron  $P$  we need additional information about the *asymptotic cone* of  $P$  in order to find a perturbation that permits us to conclude the uniqueness of an optimizing point if such a point exists.

Some very general remarks about the linear optimization problem over *pointed* polyhedra that are important in connection with the dynamic simplex algorithm of Chapter 6 and more generally, in connection with combinatorial optimization problems, follow.

Suppose that  $P \subseteq \mathbb{R}^n$  is a *pointed* polyhedron with a linear description  $Hx \leq h$  of  $P$ . Then  $r(H) = n$ . Let  $H_1$  be any  $n \times n$  nonsingular submatrix of  $H$  and denote by  $H_2$  all rows of  $H$  that are not in  $H_1$ . We partition the vector  $h$  accordingly into  $h_1$  and  $h_2$ . Given a row vector  $c \in \mathbb{R}^n$  it follows that  $x^0 = H_1^{-1}h_1$  is an optimal solution to the linear optimization problem

$$\max\{cx : x \in P\}$$

if  $H_2x^0 \leq h_2$  and  $cH_1^{-1} \geq 0$ . From an algorithmic point of view this has important consequences. Leaving aside the question of *how to find* an appropriate matrix  $H_1$ , given any nonsingular submatrix  $H_1$  of  $H$  such that  $cH_1^{-1} \geq 0$ , we need *only* to “check” the remaining inequalities for feasibility and we have solved the linear optimization problem over  $P$  – if the check comes out positive.

If the feasibility check can be done “implicitly”, i.e. without “listing and checking” every individual linear inequality of  $H_2$ , the better for us – we have avoided to represent the entire polyhedron on our digital computer. To answer the “feasibility” question we need some algorithm – another *routine* – to **solve** the following **separation problem** (or *constraint identification* problem).

**7.5(j)** Given a polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$  and a rational point  $x^0 \in \mathbb{R}^n$  **find** an inequality  $hx \leq h_0$  with  $\langle h \rangle + \langle h_0 \rangle \leq \phi$  such that  $P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}$  and  $hx^0 > h_0$  or **prove** that no such  $(h, h_0)$  exists.

The separation problem asks for a hyperplane  $hx \leq h_0$  of digital size  $\phi$  that *separates* the point  $x^0$  from the polyhedron  $P$ . If such a separating hyperplane does not exist then  $x^0 \in P$ , i.e.  $x^0$  is a *member* of  $P$ , since  $P$  has facet complexity  $\phi$ . One of the fundamental results regarding linear optimization over polyhedra is the *equivalence* of the polynomial-time solvability of the optimization and the separation problem, respectively – we return to this equivalence in Chapter 9.

Without going into any detail as to how to approach the separation problem, let us try to understand *geometrically* what is going on. For any submatrix  $H_1$  that “works” denote

$$OC(x^0, H) = \{x \in \mathbb{R}^n : H_1x \leq h_1\}$$

the *displaced outer cone with apex at  $x^0$* . Remember the displaced cone  $C(x^0, H)$  defined in (7.6), see Chapter 7.3.  $OC(x^0, H)$  and  $C(x^0, H)$  are different because  $C(x^0, H)$  is defined by *all* rows

$(h^i, h_i)$  of  $(H, h)$  for which  $h^i x^0 = h_i$  and thus  $OC(x^0, H) \supseteq C(x^0, H)$ . Because  $P \subseteq C(x^0, H)$  we have

$$\max\{cx : x \in P\} = \max\{cx : x \in C(x^0, H)\} = \max\{cx : x \in OC(x^0, H)\}$$

since  $x^0 \in P$  and by assumption  $x^0$  is an optimal solution to  $\max\{cx : x \in OC(x^0, H)\}$ . So rather than having to find the displaced cone  $C(x^0, H)$  – which could be rather difficult – it suffices to have the “outer inclusive” displaced cone  $OC(x^0, H)$  to conclude optimality of  $x^0$ . This simplifies the linear optimization task considerably and we refer to the principle that is at work here as the **outer inclusion principle**. We encourage *you* to draw appropriate pictures in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , e.g. for the linear optimization problems over all of the Platonic solids, to intuitively understand the differences between

- the linear optimization problem over  $P$ ,
- the “local” optimization problem over  $C(x^0, H)$  and
- the “very” local optimization problem over  $OC(x^0, H)$ , respectively,

where  $x^0$  is an optimizer of  $cx$  over  $P$ . If necessary for your understanding the differences, redo as well Exercise 7.13 using the outer inclusion principle; see also Chapter 9.5.

## 7.6 Geometry and Complexity of Simplex Algorithms

To give a geometric interpretation of simplex algorithms consider the linear program in standard form

$$(LP) \quad \text{minimize } cx \text{ subject to } x \in \mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\},$$

where  $\mathcal{X}$  is the feasible set and  $A$  is an  $m \times n$  matrix of data. By Definition P1 the set  $\mathcal{X}$  is a polyhedron in  $\mathbb{R}^n$  and  $Ax = b, x \geq 0$  is its linear description. The rank of the constraint set equals  $n$  because  $x = I_n x \geq 0$  is part of it. By points 7.2(a) and 7.2(c) the polyhedron  $\mathcal{X}$  is either empty or pointed. Let us assume WROG that  $\mathcal{X} \neq \emptyset$ ,  $r(A) = m$  and that  $A$  has no zero column. Now there are two possibilities: either  $x_j = 0$  for some  $j \in N = \{1, \dots, n\}$  and all  $x \in \mathcal{X}$  or there exist  $x \in \mathcal{X}$  such that  $x > 0$ . In the first case we may as well assume that variable  $x_j$  is dropped from the problem. Thus we have WROG

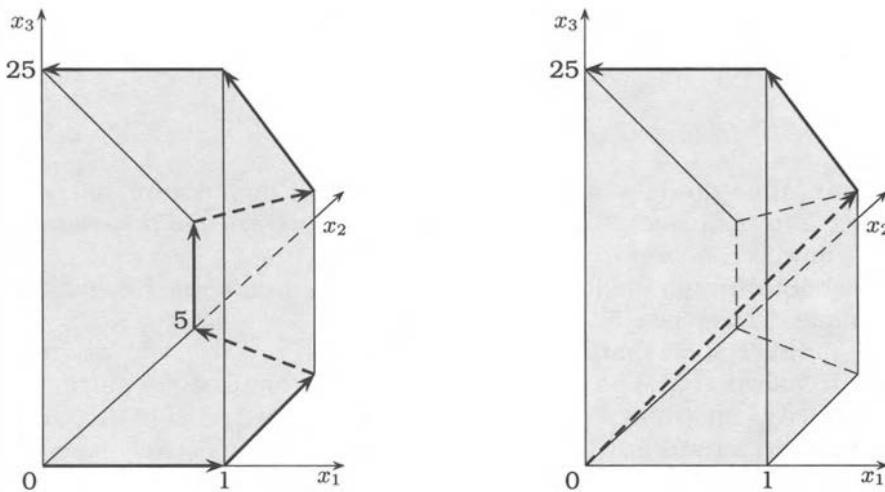
$$aff(\mathcal{X}) = \{x \in \mathbb{R}^n : Ax = b\}, \quad relint(\mathcal{X}) = \{x \in \mathcal{X} : x > 0\} \neq \emptyset,$$

and consequently  $\dim \mathcal{X} = n - m$ . By Minkowski’s theorem, see point 7.3(d), we can write

$$\mathcal{X} = conv(S) + cone(T),$$

where  $S$  is the set of extreme points of  $\mathcal{X}$  and  $T$  – in some scaling – the set of extreme rays of the asymptotic cone  $\mathcal{X}_\infty$  of  $\mathcal{X}$ . The pair  $(S, T)$  is a minimal generator of  $\mathcal{X}$  and since  $\mathcal{X}$  is line free, it is also the canonical generator of  $\mathcal{X}$ . The pair  $(\{0\}, T)$  is a minimal (and canonical) generator of  $\mathcal{X}_\infty$ .

**7.6(a)** Every basic feasible solution to the linear program (LP) is an extreme point of  $\mathcal{X}$  and vice versa, every extreme point of  $\mathcal{X}$  is a basic feasible solution to (LP).



**Fig. 7.3.** Simplex paths with and without block pivots

**7.6(b)** If the simplex algorithm changes from a feasible basis  $B$  to a new basis  $B'$  in a single pivot operation, then the respective basic feasible solutions  $x$  and  $x'$  for (LP) are adjacent extreme points of  $\mathcal{X}$  if  $x \neq x'$ .

**7.6(c)** If the simplex algorithm detects unboundedness of the objective function at some basis  $B$ , then the halfline  $x(\lambda)$  for  $\lambda \geq 0$  of Chapter 4 defines an extreme ray of  $\mathcal{X}$  and  $x(0)$  is an extreme point of  $\mathcal{X}$ .

### 7.6.1 Pivot Column Choice, Simplex Paths, Big M Revisited

In the geometric language of this chapter the primal simplex algorithm is an *edge-following* algorithm that moves on the “outside” of the polyhedron  $\mathcal{X}$  since the successive pivot operations describe a “path” on  $\mathcal{X}$  consisting of 1-dimensional faces or edges of  $\mathcal{X}$ , see Figure 7.3.

The various pivot column selection criteria of Chapter 5 select a particular one among all possible edges of  $\mathcal{X}$  that improve the objective function. E.g. criterion (c4) selects an edge where –in the absence of degeneracy– one makes –myopically– the largest possible gain. Criterion (c1) proceeds by selecting a “minimum gradient” edge in the space of all nonbasic variables.

Most commercial software packages for linear programming use some variation of the “steepest edge” criterion (c5). Like in Exercise 4.2 we write  $x(\lambda)$  for the edge of  $\mathcal{X}$  along which we wish to move to a new basic feasible solution, i.e.,

$$\mathbf{x}_B(\lambda) = \bar{\mathbf{b}} - \lambda \mathbf{y}_j, \quad x_j(\lambda) = \lambda \quad \text{and} \quad x_k(\lambda) = 0 \quad \text{for all } k \in N - I, k \neq j.$$

Let  $\mathbf{y} \in \mathbb{R}^n$  be the “direction” vector of the edge of  $\mathcal{X}$ . Partitioning  $\mathbf{y}$  into  $\mathbf{y}_B$  and  $\mathbf{y}_R$  we have  $\mathbf{y}_B = -\mathbf{y}_j$  and  $\mathbf{y}_R$  has exactly one entry equal to +1 corresponding to  $j \in N - I$ , all others equal to zero. The (normalized) vector  $\mathbf{c}/\|\mathbf{c}\|$  is perpendicular to the hyperplane  $\mathbf{c}\mathbf{x} = z_B$  and is the direction of “steepest” increase for the objective function since  $\mathbf{c}(\mathbf{x} + \lambda \mathbf{c}^T / \|\mathbf{c}\|) = z_B + \lambda \|\mathbf{c}\|$  increases for  $\lambda \geq 0$ . We want to move in the opposite direction since we are minimizing. The

cosine of the angle between  $c/\|c\|$  and  $y$  is

$$\cos \phi = \frac{cy}{\|c\|\|y\|} = \frac{c_j - c_B B^{-1} a_j}{\|c\| \sqrt{1 + \sum_{i=1}^m (y_j^i)^2}}.$$

The norm  $\|c\|$  is a constant. Thus the criteria (c5) of Chapter 5 select a “direction of steepest descent” in the space of all (basic and nonbasic) variables of (LP). In the presence of *degeneracy* these geometrical notions must, however, be interpreted with care.

*Block pivots* on the other hand “shoot through the (relative) interior” of  $\mathcal{X}$ , see Figure 7.3. for the geometry of the worst-case example of Exercises 5.9 and 5.10 with  $n = 3$ ,  $a = b = 2$ ,  $c = 5$ .

Let  $\phi \geq n + 1$  be any integer number such that  $\langle a^i \rangle + \langle b_i \rangle \leq \phi$  for each row of  $(A, b)$  where  $c$  and  $(A, b)$  are all rational data. It follows that  $\mathcal{X}$  is a polyhedron of facet complexity  $\phi$  since the encoding length of each nonnegativity constraint  $-x_j \leq 0$  is clearly less than  $\phi$ . Using the facet complexity  $\phi$  of  $\mathcal{X}$  the two Big M-devices used in Chapter 5 and Chapter 6, respectively, can be made *theoretically* precise; see the text.

### 7.6.2 Gaussian Elimination, Fill-In, Scaling

The worst-case example of Exercise 5.9 has shown that the running time of the primal simplex algorithm with choice rules (c1) and (r1) is not bounded by a polynomial in  $n$ ,  $\phi$  and  $\langle c \rangle$  since  $2^n - 1$  steps are required. This does *not* mean, however, that polynomial-time simplex algorithms *cannot exist*, but to date no variant of the simplex algorithm is known to have “good” worst-case behavior.

Just like in the analysis of the double description algorithm we are thus led to study a single iteration of the simplex algorithm for a linear program in standard form. The work of a single iteration calls for the solution of three systems of  $m$  equations in  $m$  unknowns – see Chapter 5. Gaussian elimination is used to solve systems of equations and so let us analyze this algorithm. Let  $Bx = a$ , i.e.

$$\begin{aligned} b_1^1 x_1 + b_2^1 x_2 + \cdots + b_m^1 x_m &= a_1 \\ b_1^2 x_1 + b_2^2 x_2 + \cdots + b_m^2 x_m &= a_2 \\ \vdots &\quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ b_1^m x_1 + b_2^m x_2 + \cdots + b_m^m x_m &= a_m, \end{aligned} \tag{7.19}$$

be the system of  $m$  linear equations in  $m$  unknowns. By re-indexing the columns and rows if necessary we can assume that  $b_1^1 \neq 0$  and we will call  $b_1^1$  the (first) pivot element. Assuming that  $\det B \neq 0$  we bring (7.19) iteratively into the upper triangular form (7.20).

$$\begin{aligned} c_1^1 x_1 + c_2^1 x_2 + \cdots + c_m^1 x_m &= d_1 \\ c_2^2 x_2 + \cdots + c_m^2 x_m &= d_2 \\ \vdots &\quad \vdots \quad \vdots \quad \vdots \\ c_m^m x_m &= d_m. \end{aligned} \tag{7.20}$$

Multiplying the first row of (7.19) by  $-b_1^i/b_1^1$  and adding the result to row  $i$  of (7.19) for  $2 \leq i \leq m$  we create a new system such that all coefficients in rows  $2, \dots, m$  of the first column are zero.

The right-hand sides  $a_i$  for  $2 \leq i \leq m$  are changed correspondingly. Variable  $x_1$  is thus uniquely determined by the values of  $x_2, \dots, x_m$  and the first row since  $b_1^1 = c_1^1 \neq 0$ . Hence we have reduced the problem in  $m$  variables to one in  $m - 1$  variables and we can repeat. If  $c_2^2 = b_2^2 - b_2^1 b_1^2 / b_1^1 = 0$  we permute the rows so as to get  $c_2^2 \neq 0$  which is the (second) pivot element. If no  $c_i^i \neq 0$  exists where  $2 \leq i \leq m$  then  $\det B = 0$  contrary to our assumption, otherwise we repeat.

Once the format (7.20) is obtained we solve the system (7.20) by “backward substitution”: since  $x_m = d_m/c_m^m$  we get  $x_{m-1} = (d_{m-1} - c_{m-1}^{m-1}x_m)/c_{m-1}^{m-1}$ , etc until we have found the value for  $x_1$ . Gaussian elimination requires  $\mathcal{O}(m^3)$  “flops”.

$$\begin{array}{lll} x_1 + x_2 & = 1 & x_1 + x_2 & = 1 \\ x_2 + x_3 & = 1 & x_2 + x_3 & = 1 \\ x_3 + x_4 & = 1 & \rightsquigarrow & x_3 + x_4 & = 1 \\ x_4 + x_5 & = 1 & & x_4 + x_5 & = 1 \\ x_1 & + x_5 & = 1 & & 2x_5 & = 1. \end{array} \quad (7.21)$$

To illustrate Gaussian elimination consider the  $n \times n$  matrix  $C_n$  that has exactly two ones per row, two ones per column and that does not “decompose” into two or more submatrices with the same characteristics. Suppose  $n \geq 3$  is odd and that we want to solve the system of equations  $C_n \mathbf{x} = \mathbf{e}$  where  $\mathbf{e}$  is vector of  $n$  ones. Carrying out Gaussian elimination you find that the total number of nonzero elements of the intermediate matrices equals  $2n$  during the iterations except for the last system which has  $2n - 1$  nonzeros and is triangular. The first and the last system of equations is shown in (7.21) for  $n = 5$  and the last one is readily solved by backward substitution. By contrast, the inverse  $C_n^{-1}$  is 100% *dense*, i.e. it has  $n^2$  nonzero entries. This shows that our use of the *basis inverse* in the statement of the simplex algorithms in Chapter 5 and Chapter 6 was *purely didactical* because solving systems of equations can be done with less storage space and greater accuracy directly, i.e. without the calculation of an inverse.

Setting aside a full  $m \times (m + 1)$  array of storage for the execution of Gaussian elimination is in most cases unnecessary. If the number of nonzeros in the course of calculation does not increase by “too much”, i.e. if the “fill-in” of the matrix is relatively modest, data structures exploiting the sparsity of  $B$  can be used. To keep the fill-in “down”, we need a *choice rule* for the “next” pivot element. On the other hand, because we are *dividing* a pivot element may become unacceptably small or unacceptably large relative to the wordsize of the computer. So either we must avoid divisions altogether or use a choice rule that takes into account the “size” of the pivot element as well as the fill-in. When divisions are used then the initial matrix ( $B \mathbf{a}$ ) is frequently *scaled* in order to improve the “quality” of the solution obtained; see the text.

### 7.6.3 Iterative Step I, Pivot Choice, Cholesky Factorization

We denote by  $B^{(k)}$  the transformed matrix at iteration  $k$  where we have subtracted the pivot row from itself as well so that it contains zeros only. Row  $i$  of  $B^{(k)}$  is denoted by  $_k b^i$ , column  $j$  by  $_k b_j$  and the element in row  $i$  and column  $j$  by  ${}_k b_j^i$ . Denote by  $E_k$  the index set of pivot rows, by  $J_k$  the index set of pivot columns that the algorithm has processed in iterations  $1, \dots, k$  where initially  $E_0 = J_0 = \emptyset$ . Denote by  $L^{(k)}$  and  $U^{(k)}$  matrices of size  $n \times k$  and  $k \times n$ , respectively, that are initially empty and built up during the course of calculations. Denote by  $a^k$  the transformed right-hand side at iteration  $k$  and  $M = \{1, \dots, m\}$ . Initially, we have  $k = 0$ ,  $B^{(0)} = B$  and  $a^0 = a$ . Assuming that  $\det B \neq 0$  the **iterative step of Gaussian elimination** goes as follows.

Find  $h \in M - E_k$ ,  $j \in M - J_k$  such that  ${}_k b_j^h$  satisfies the pivot choice rule. Set  $\ell_{k+1} := ({}_k b_j^h)^{-1} {}_k b_j$ ,  $\mathbf{u}^{k+1} := {}_k b^h$ ,  $\mathbf{B}^{(k+1)} := \mathbf{B}^{(k)} - \ell_{k+1} \mathbf{u}^{k+1}$ ,  $\mathbf{L}^{(k+1)} := (\mathbf{L}^{(k)} \ \ell_{k+1})$ ,  $\mathbf{U}^{(k+1)} := \begin{pmatrix} \mathbf{U}^{(k)} \\ \mathbf{u}^{k+1} \end{pmatrix}$ ,  $E_{k+1} := E_k + h$  and  $J_{k+1} := J_k + j$ . Set  $\mathbf{a}^{k+1} := \mathbf{a}^k - a_h^k \ell_{k+1}$  and  $a_h^{k+1} := a_h^k$ . Set  $k := k + 1$  and repeat while  $k < m$ .

We need to give some pivot choice rule. For any  $0 \leq k < m$  denote by  $r_h$  the number of nonzero entries of row  $h$  and by  $c_j$  that of column  $j$  of  $\mathbf{B}^{(k)}$  where  $1 \leq h, j \leq m$ . To control the “fill-in” one chooses  ${}_k b_j^h \neq 0$  such that  $(r_h - 1)(c_j - 1)$  is minimum among all candidates.

To control the size of the pivot element one chooses  $h$  and  $j$  such that  $|{}_k b_j^h|$  is maximum (*complete pivoting*) or one picks a column  $j \in M - J_k$ , e.g. the “first” one, and then selects  $h \in M - E_k$  such that  $|{}_k b_j^h|$  is maximum (*partial pivoting*) or one picks some column  $j \in M - J_k$  and then a row  $h \in M - E_k$  such that  $|{}_k b_j^h| / \max_{i,e} \{|{}_k b_e^i|\}$  is maximum.

The product  $\ell_{k+1} \mathbf{u}^{k+1}$  in the iterative step is the *dyadic product* that we have used already in Chapter 4. By induction it follows that  ${}_k b_j^i = 0$  for all  $(i, j)$  with  $i \in E_k$ ,  $j \in M$  and with  $i \in M$  and  $j \in J_k$ . Thus  $\mathbf{B}^{(m)} = \mathbf{O}$  and developing backwards we find

$$\mathbf{B} = \ell_1 \mathbf{u}^1 + \ell_2 \mathbf{u}^2 + \cdots + \ell_m \mathbf{u}^m = (\ell_1 \dots \ell_m) \begin{pmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^m \end{pmatrix} = \mathbf{LU},$$

where we have set  $\mathbf{L} = \mathbf{L}^{(m)}$  and  $\mathbf{U} = \mathbf{U}^{(m)}$  for notational simplicity. This is one form of the **Cholesky factorization** of the “concatenated” Gaussian algorithm. Let  $E_m = \{h_1, \dots, h_m\}$  and  $J_m = \{j_1, \dots, j_m\}$  be the ordered lists of indices constructed iteratively. Define the elements of two *permutation matrices*  $\mathbf{P}$  and  $\mathbf{Q}$  of size  $m \times m$  by

$$p_{h_i}^i = q_{j_i}^i = 1 \text{ for } 1 \leq i \leq m, \quad p_j^i = q_j^i = 0 \text{ otherwise.}$$

The matrix  $\mathbf{PL}$  is lower-triangular and  $\mathbf{UQ}$  is upper-triangular. The diagonal elements of  $\mathbf{PL}$  are all equal to 1, while the diagonal elements of  $\mathbf{UQ}$  are the pivot elements. Consequently,  $\mathbf{PBQ} = (\mathbf{PL})(\mathbf{UQ})$  is a decomposition of  $\mathbf{PBQ}$  into *triangular factors*  $\mathbf{PL}$  and  $\mathbf{UQ}$ . The matrices  $\mathbf{L}$  and  $\mathbf{U}$  are *generalized triangular* matrices, i.e. matrices that up to row and column permutations are triangular.

For the solution of  $\mathbf{Bx} = \mathbf{a}$  this means that  $\mathbf{LUx} = \mathbf{a}$ . You find  $\mathbf{x}$  in two steps: first determine a vector  $\mathbf{b}$  such that  $\mathbf{Lb} = \mathbf{a}$  and then determine  $\mathbf{x}$  from  $\mathbf{Ux} = \mathbf{b}$ . Both systems to be solved are generalized triangular systems.  $\mathbf{Lb} = \mathbf{a}$  is solved by “forward substitution” and  $\mathbf{Ux} = \mathbf{b}$  by “backward substitution”; see the text.

#### 7.6.4 Cross Multiplication, Iterative Step II, Integer Factorization

To get a “division free” form of Gaussian elimination let us go back to the original system of equations (7.19) and assume that the original data are all integers. Suppose that we multiply equation  $i$  of (7.19) by  $b_1^i$  and add to it the first row multiplied by  $-b_1^i$  where  $2 \leq i \leq m$ . This operation is called *cross multiplication* and it creates zeros in the first column as well. The resulting system is equivalent to the original one and all data remain integer.

Repeating this operation “blindly” creates numbers that grow rapidly in size. To see this, solve the system of equations  $Hx = e$  this way where  $H$  is the matrix of part (iv) of Exercise 7.10 and  $e$  is a vector of  $n$  ones. Using Euclidean reduction, however, we can control the growth of the numbers and in general the numbers simplify via appropriate divisions that are remainderless, i.e. the calculations are, in fact, *division free*.

We use the same notation, initialize all arrays like we did for the iterative step of the Gaussian algorithm with division and set  $d_0 := 1$ ,  $\mathbf{b} := \mathbf{0}$  where  $\mathbf{b} \in \mathbb{R}^m$ . For any nonzero scalars  $\alpha$ ,  $\beta$  and matrix  $B$  we write  $(\alpha B)/\beta$  to mean that we multiply every (nonzero) element of  $B$  by  $\alpha$  before dividing each product by the scalar  $\beta$ . We do likewise for vectors. Assuming that  $\det B \neq 0$  the **iterative step of division free Gaussian elimination** goes as follows.

Find  $h \in M - E_k$ ,  $j \in M - J_k$  such that  ${}_k b_j^h$  satisfies the pivot choice rule. Set  $d_{k+1} := {}_k b_j^h$ ,  $\ell_{k+1} := {}_k b_j$ ,  $\mathbf{u}^{k+1} := {}_k \mathbf{b}^h$ ,  $\mathbf{B}^{(k+1)} := (d_{k+1} \mathbf{B}^{(k)} - \ell_{k+1} \mathbf{u}^{k+1})/d_k$ ,  $\mathbf{L}^{(k+1)} := (\mathbf{L}^{(k)} \ell_{k+1})$ ,  $\mathbf{U}^{(k+1)} := \begin{pmatrix} \mathbf{U}^{(k)} \\ \mathbf{u}^{k+1} \end{pmatrix}$ ,  $E_{k+1} := E_k + h$ , and  $J_{k+1} := J_k + j$ . Set  $b_h := a_h^k$  and  $\mathbf{a}^{k+1} := (d_{k+1} \mathbf{a}^k - b_h \ell_{k+1})/d_k$ . Set  $k := k + 1$  and repeat while  $k < m$ .

We are *dividing*, but all divisions are remainderless. We can thus use just that part of the pivot choice rule which controls the fill-in since the size of the pivot element  ${}_k b_j^h \neq 0$  does not matter.

Like in Step I we conclude that  $\mathbf{B}^{(m)} = \mathbf{0}$ . Developing backwards and dividing by  $\prod_{i=1}^m d_i$  one gets

$$\mathbf{B} = \sum_{i=1}^m (d_{i-1} d_i)^{-1} \ell_i \mathbf{u}^i = \mathbf{L} \mathbf{D}^{-1} \mathbf{U},$$

where  $\mathbf{L} = \mathbf{L}^{(m)}$ ,  $\mathbf{U} = \mathbf{U}^{(m)}$  and  $\mathbf{D} = \text{diag}(d_0 d_1, \dots, d_{m-1} d_m)$ , which is an *integer* Cholesky factorization of  $\mathbf{B}$  if the data are integer.

Like above  $\mathbf{L}$  and  $\mathbf{U}$  are generalized triangular matrices. From the index sets  $E_m$  and  $J_m$  construct permutation matrices  $P$ ,  $Q$  and get  $P B Q = P L D^{-1} U Q$  where  $P L$  is lower triangular,  $U Q$  is upper triangular. By induction it follows that  $a_h^k = 0$  for all  $h \in E_k$  and thus  $\mathbf{a}^m = \mathbf{0}$ . Developing backwards and dividing by  $\prod_{i=1}^m d_i$  it follows that  $\mathbf{b} = (b_1, \dots, b_m)^T$  constructed in the iterative step satisfies  $\mathbf{a} = \mathbf{L} \mathbf{D}^{-1} \mathbf{b}$ .

Consequently, solving  $\mathbf{U} \mathbf{x} = \mathbf{b}$  by backward substitution like above the solution  $\mathbf{x}$  for the system  $\mathbf{B} \mathbf{x} = \mathbf{a}$  is found. Since all divisions are remainderless in the iterative step  $\mathbf{b}$  has integer components only and divisions (with a possible remainder) are performed only when the solution  $\mathbf{x}$  is determined.

### 7.6.5 Division Free Gaussian Elimination and Cramer's Rule

Let  $B$  be any  $m \times m$  matrix,  $M = \{1, \dots, m\}$  and  $\mathbf{a} \in \mathbb{R}^m$  be any column vector. For  $j \in M$  denote by  $B_j^a$  the matrix obtained from  $B$  by replacing column  $j$  of  $B$  by the vector  $\mathbf{a}$ .  $B_{M-j+a}$  denotes the matrix obtained from  $B$  by deleting column  $j$  from  $B$  and adding column  $\mathbf{a}$  as the last column to  $B_{M-j}$ . Likewise, for any  $i, \ell \in M$  denote by  $B^{M-\ell+i}$  the matrix obtained by dropping row  $\ell$  and adding row  $i$  as the last row of that matrix.

For any  $E \subseteq M$ ,  $J \subseteq M$  such that  $|E| = |J|$  and for any  $j \in J$  denote by  $B_{J-j+a}^E$  the matrix obtained from  $B_J^E$  by deleting column  $j$  and adding the column vector  $\mathbf{a}_E = (a_h)_{h \in E}$  as the last

column to the matrix  $B_{J-j}^E$ . We should write  $B_{J-j+a_E}^E$ , but for notational simplicity we drop the index  $E$  of  $a_E$ .

For further reference we also denote by  $B_{J+c}^{E+e}$  for  $c, e \in M$  the matrix that results when we add  $(b_c^h)_{h \in E}$  as the last column to  $B_J^E$  and  $(b_t^e)_{t \in J}$  plus the element  $b_c^e$  as the last row to  $B_{J+c}^E$  where  $b_c^e$  is put into the new column of the matrix of size  $(|E| + 1) \times (|J| + 1)$ .

**7.6(d)** With the above notation, we have for all  $E \subseteq M$ ,  $J \subseteq M$  with  $|E| = |J|$  and for all  $p \in J$  that

$$\sum_{j=1}^m \det B_{J-p+j}^E \det B_j^a = \det B \det B_{J-p+a}^E.$$

We state the division free Gaussian algorithm in “programmable” form using sets of vectors rather than matrices. This way it becomes clear how to use the sparse matrix storage techniques. We assume that the original matrix  $B$  and the intermediate matrices  $B^{(k)}$  are stored rowwise in a sparse structure called  $RB_k$  and columnwise in a structure  $CB_k$ . Moreover, we **assume** that  $B \neq 0$ .

**Division free Gaussian algorithm** ( $m, a, B, x, rk, Det, E, J, L, U, D$ )

**Step 0:** Set  $k := 0$ ,  $RB_0 := \{b^1, \dots, b^m\}$ ,  $CB_0 := \{b_1, \dots, b_m\}$ ,  $E_0 := \emptyset$ ,  $J_0 := \emptyset$ ,  $D_0 := \emptyset$ ,  $L_0 := \emptyset$ ,  $U_0 := \emptyset$ ,  $a^0 := a$ ,  $b := 0$ ,  $rk := 0$ ,  $Det := 0$ ,  $d_0 := 1$  and  $M = \{1, \dots, m\}$ .

**Step 1:** Find  $h \in M - E_k$ ,  $j \in M - J_k$  such that  ${}_k b_j^h$  satisfies the pivot choice rule.

**if** none found **go to** Step 3.

Set  $rk := rk + 1$ ,  $d_{k+1} := {}_k b_j^h$ ,  $D_{k+1} := D_k \cup \{d_{k+1}\}$ ,  $Det := d_{k+1}$ ,  $\ell := {}_k b_j$  and  $u = {}_k b^h$ .

**Step 2:** Set  $L_{k+1} := L_k \cup \{\ell\}$ ,  $U_{k+1} := U_k \cup \{u\}$ ,  $E_{k+1} := E_k + h$ ,  $J_{k+1} := J_k + j$ ,  $RB_{k+1} := \{(d_{k+1}({}_k b^p) - \ell_p u)/d_k : p \in M - E_{k+1}\}$ ,  $CB_{k+1} := \{(d_{k+1}({}_k b_q) - u_q \ell)/d_k : q \in M - J_{k+1}\}$ ,  $b_{k+1} := a_h^k$  and  $a^{k+1} := (d_{k+1} a^k - b_{k+1} \ell)/d_k$ . Set  $k := k + 1$  and **go to** Step 1.

**Step 3:** **if**  $a^{rk} \neq 0$  **stop** “output: the system  $Bx = a$  has no solution”.

Set  $E := E_{k-1}$ ,  $J := J_{k-1}$ ,  $D := D_{k-1}$ ,  $L := L_{k-1}$ ,  $U := U_{k-1}$  and  $x := 0$ .

Set  $x_{j_h} := (b_h Det - \sum_{i=h+1}^{|J|} u_{j_i}^h x_{j_i})/u_{j_h}^h$  for  $h = |J|, |J| - 1, \dots, 1$ .

**stop** “output  $x, rk, Det, E, J, L, U, D$ ”.

The division free Gaussian algorithm, or DFGA for short, implements the iterative step II as stated above except that we have dropped the assumption that  $\det B \neq 0$ .

The choice of the pivot element in Step 1 has been left open to *your* choice of *your favorite heuristic* for doing so. You must choose a pivot element  ${}_k b_j^h$  that is nonzero, so that “none found” signifies that  ${}_k b_j^h = 0$  for all  $h \in M - E_k$ ,  $j \in M - J_k$ . Since we are not dividing the size of  ${}_k b_j^h$  does not matter. The only concern is the fill-in and you may implement the above pivot choice rule for controlling the fill-in and to break ties you may choose the *smallest* nonzero  $|{}_k b_j^h|$  that qualifies.

In the next point remember that the set  $J$  is in reality an *ordered* list since the *position* of a column in a matrix affects the sign of its determinant.

**7.6(e)** DFGA finds the rank  $r(B) = rk$  of  $B$ , a row set  $E$  and column set  $J$  with  $|E| = |J| = rk$  and  $Det = \det B_J^E \neq 0$ . For  $1 \leq k \leq rk$  the divisors satisfy  $d_k = \det B_{J_k}^{E_k} \neq 0$ , the vectors  $\ell_k \in L$  are linearly independent and satisfy  $\ell_k^h = \det B_{J_k}^{E_{k-1}+h}$  for all  $h \in M - E_{k-1}$ ,  $\ell_k^h = 0$  otherwise.

The vectors  $u^k \in U$  for  $1 \leq k \leq rk$  are linearly independent and satisfy  $u_j^k = \det B_{J_{k-1}+j}^{E_k}$  for all  $j \in M - J_{k-1}$ ,  $u_j^k = 0$  otherwise. The vector  $b$  constructed by the algorithm satisfies

$b_k = \det B_{J_{k-1}+a}^{E_k}$  for  $1 \leq k \leq rk$ ,  $b_k = 0$  otherwise. If  $\mathbf{a}^{rk} \neq \mathbf{0}$  then  $rk < m$  and the system  $B\mathbf{x} = \mathbf{a}$  has no solution. Otherwise, DFGA returns a vector  $\mathbf{x}$  such that  $\mathbf{y} = \text{Det}^{-1}\mathbf{x}$  satisfies the system  $B\mathbf{y} = \mathbf{a}$  and  $x_j = (-1)^{|J|+h(j)} \det B_{J-j+a}^E$  for all  $j \in J$ ,  $x_j = 0$  otherwise, where  $h(j)$  is the position number of  $j$  in the set  $J$ .

If  $rk = m$  then DFGA produces the solution vector in exactly the form as given by **Cramer's rule**. If the system is solvable then this is true also, in an appropriately modified form, even when  $rk < m$ .

The number of elementary operations of DFGA are a polynomial function of  $m$ .  $\mathcal{O}(m^3)$  elementary operations are required for its execution. If properly implemented for *truly sparse* large-scale problems then the estimation  $\mathcal{O}(m^3)$  is a gross overestimation of the necessary work.

Like in the case of the double description algorithm, Euclidean reduction brings about a stronger reduction in the size of the numbers to be processed by DFGA than the respective divisions by  $d_k$ .

**7.6(f)** DFGA's running time for solving the system of equations (7.19) with rational data is bounded by a function that is polynomial in its digital input size.

Suppose that  $A\mathbf{x} = \mathbf{b}$  is any system of  $m$  equations of complexity  $\phi$  in  $n$  unknowns with rational data. If  $m < n$ , then add  $n - m$  trivial rows  $\sum_{j=1}^n 0x_j = 0$  and if  $n < m$ , then add  $m - n$  new variables and zero columns to  $A\mathbf{x} = \mathbf{b}$  to bring it into the form (7.19). Running DFGA on the square system of equations we either find a rational solution of size at most  $4p^2\phi$  where  $p = \min\{m, n\}$  or we get the message that it does not have a solution in time that is polynomial in  $m$ ,  $n$  and  $\phi$ . By "solving" a system of equations we mean the process of proving it unsolvable or producing a solution to it.

**7.6(g)** Every  $m \times n$  system of equations  $A\mathbf{x} = \mathbf{b}$  with rational data of facet complexity  $\phi$  can be "solved" in time that is polynomial in  $m$ ,  $n$  and  $\phi$ . Moreover, if  $A\mathbf{x} = \mathbf{b}$  is solvable, then a rational solution  $\mathbf{x} \in \mathbb{R}^n$  such that  $|x_j| \leq 2^{4p\phi}$ ,  $x_j = p_j/q_j$  with  $p_j, q_j$  integer and  $1 \leq q_j \leq 2^{4p\phi}$  for  $1 \leq j \leq n$  and thus  $\langle \mathbf{x} \rangle \leq 4p^2\phi$  can be found in time that is polynomial in  $m$ ,  $n$  and  $\phi$  where  $p = \min\{m, n\}$ .

## 7.7 Circles, Spheres, Ellipsoids

An  $n \times n$  matrix  $Q$  with real elements  $q_j^i$  is *symmetric* if  $q_j^i = q_i^j$  for all  $1 \leq i, j \leq n$ . A symmetric matrix  $Q$  is *positive definite* if  $\mathbf{x}^T Q \mathbf{x} > 0$  for all  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} \neq \mathbf{0}$ , it is *positive semi-definite* if  $\mathbf{x}^T Q \mathbf{x} \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^n$  and *indefinite* if  $\mathbf{x}^T Q \mathbf{x} > 0$ ,  $\mathbf{y}^T Q \mathbf{y} < 0$  for some  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . The concepts of a negative (semi-)definite matrix are defined likewise, with  $\mathbf{x}^T Q \mathbf{x} < 0$  and  $\mathbf{x}^T Q \mathbf{x} \leq 0$ , respectively.

Let  $q(\mathbf{x}) = \sum_{i=1}^n \sum_{j=i}^n r_j^i x_i x_j + \sum_{j=1}^n c_j x_j + c_0$  be any function of degree two or a *quadratic form* where  $r_j^i, c_j \in \mathbb{R}$  are arbitrary. Setting  $q_j^i = (r_j^i + r_i^j)/2$  for  $1 \leq i, j \leq n$  we can write  $q(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} + c_0$  where  $Q = (q_j^i)$  is a symmetric matrix and  $\mathbf{c} \in \mathbb{R}^n$  a row vector. If  $Q$  is nonsingular then we can set  $\mathbf{x}_C = -(1/2)Q^{-1}\mathbf{c}^T$ ; we get  $q(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_C)^T Q (\mathbf{x} - \mathbf{x}_C) + c_0^*$  where  $c_0^* = c_0 - (1/4)\mathbf{c}^T Q^{-1} \mathbf{c}^T$  and the point  $\mathbf{x}_C$  is the "center of the quadric"  $q(\mathbf{x})$ .

The matrix  $Q = I_n$  is positive definite. The set  $B(\mathbf{0}, r) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T I_n \mathbf{x} \leq r^2\}$  is the  $n$ -dimensional *ball* or *sphere* with center  $\mathbf{0}$  and radius  $r > 0$  in  $\mathbb{R}^n$ . The sphere  $B(\mathbf{x}_C, r)$  with center  $\mathbf{x}_C$  and radius  $r$  is

$$B(\mathbf{x}_C, r) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \mathbf{x}_C)^T I_n (\mathbf{x} - \mathbf{x}_C) \leq r^2\}.$$

Let  $E_k = \{1, \dots, k\}$  and  $J_k = \{1, \dots, k\}$  for  $1 \leq k \leq n$ . The  $k \times k$  submatrices  $Q_{J_k}^{E_k}$  of  $Q$  are the *principal minors* of  $Q$ . In the next points  $Q$  is assumed to be real and symmetric.

**7.7(a)**  $Q$  is positive definite if and only if  $\det Q_{J_k}^{E_k} > 0$  for  $1 \leq k \leq n$ .

**7.7(b)**  $Q$  is positive definite if and only if there exists an  $n \times n$  nonsingular matrix  $F$  such that  $Q = F^T F$ . If  $Q$  is positive definite, then so is  $Q^{-1}$ .

**7.7(c)**  $Q$  is orthogonal and positive definite if and only if  $Q = \text{diag}(q_1, \dots, q_n)$  where  $q_i > 0$  for  $1 \leq i \leq n$ .

From point 7.7(c) it follows that  $Q = I_n$  is the unique orthonormal positive definite matrix. This explains our particular interest in the sphere  $B(0, 1)$  which is called the *unit sphere*.

**7.7(d)** The function  $\|x\|_Q = \sqrt{x^T Q^{-1} x}$  defines a norm on  $\mathbb{R}^n$  if  $Q$  is positive definite.

The norm  $\|x\|_Q$  on  $\mathbb{R}^n$  is the *general Euclidean* or *ellipsoidal* norm. For any positive definite matrix  $Q$  of size  $n \times n$ ,  $x_C \in \mathbb{R}^n$  and  $r > 0$  we denote by

$$E_Q(x_C, r) = \{x \in \mathbb{R}^n : (x - x_C)^T Q^{-1} (x - x_C) \leq r^2\}$$

the *ellipsoid* defined by  $Q$  with center  $x_C$  and “radius”  $r$ . The radius  $r$  is a scaling factor or *blow-up factor*, i.e.  $E_Q(x_C, r) \subset E_Q(x_C, r')$  for all  $0 \leq r < r' < \infty$ .  $E_Q(x_C, r)$  is *affinely equivalent* to the sphere  $B(0, r)$ . Hence  $E_Q(x_C, r)$  is a *compact* convex subset of  $\mathbb{R}^n$ , i.e. it is closed, bounded and convex and we can optimize over  $E_Q(x_C, r)$  using standard techniques from calculus.

Let  $H$  be any  $n \times n$  matrix and consider the problem of finding a vector  $x \neq 0$  with  $n$  real or complex components such that  $Hx = \lambda x$  for some real or complex number  $\lambda$ . Any vector  $x \neq 0$  satisfying the equation is called an *eigenvector* and the corresponding value of  $\lambda$  an *eigenvalue* of the matrix  $H$ .

**7.7(e)** Let  $H$  be an  $n \times n$  symmetric matrix of reals. There exist  $n$  real eigenvalues  $\lambda_1 \leq \dots \leq \lambda_n$  and  $n$  linearly independent real eigenvectors  $x_1, \dots, x_n$  of  $H$  that are pairwise orthogonal.

Let  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  and  $X = (x_1 \dots x_n)$ , where  $\lambda_1 \leq \dots \leq \lambda_n$  are the eigenvalues and  $x_1, \dots, x_n$  the eigenvectors of  $H$ . It follows that  $H = X \Lambda X^{-1}$  where  $X$  is an orthogonal matrix, i.e.  $X^{-1} = X^T$  and  $\det X = \pm 1$ , and thus  $\det H = \det \Lambda = \prod_{i=1}^n \lambda_i$ . Consequently,  $H$  is singular if and only if  $\lambda_i = 0$  for some  $i$ . If  $r(H) = r$  then there are exactly  $n - r$  eigenvalues  $\lambda_i = \dots = \lambda_{i+n-r} = 0$  where  $1 \leq i \leq r$  and the corresponding eigenvectors  $x_i, \dots, x_{i+n-r}$  form an *orthonormal basis* of the subspace  $\{x \in \mathbb{R}^n : Hx = 0\}$  since  $x_i^T x_j = 0$  and  $\|x_i\| = 1$  for all  $1 \leq i \neq j \leq n$ .

The *trace* of  $H$  is the sum of its diagonal elements, i.e.  $\text{trace}(H) = \sum_{i=1}^n \lambda_i$  and thus  $\text{trace}(H) = \sum_{i=1}^n \lambda_i$  for any real, symmetric matrix  $H$ .

If  $H$  is a positive definite matrix, then  $H$  has  $n$  positive eigenvalues that need, however, not be distinct. For positive semi-definite matrices there is a similar statement, i.e. their eigenvalues are nonnegative.

Let  $0 < \lambda_1 \leq \dots \leq \lambda_n$  be the eigenvectors of the positive definite matrix  $Q$  defining the ellipsoid  $E_Q(x_C, r)$  and let  $x_1, \dots, x_n$  be the corresponding eigenvectors. From  $Q = X \Lambda X^T$  we have  $Q^{-1} = X \Lambda^{-1} X^T$  and thus the eigenvalues of  $Q^{-1}$  are given by  $1/\lambda_i$  for  $1 \leq i \leq n$  while the eigenvectors of  $Q$  and  $Q^{-1}$  are the same. The affine transformation  $y = -X^T x_C + X^T x$  for all  $x \in \mathbb{R}^n$  corresponds to a change of the coordinate system in  $\mathbb{R}^n$  which leaves the *length* of any vector unchanged as well

as the *angle* formed by any two vectors of  $\mathbb{R}^n$ . Under this transformation  $(\mathbf{x} - \mathbf{x}_C)^T Q^{-1}(\mathbf{x} - \mathbf{x}_C) = \sum_{i=1}^n y_i^2 / \lambda_i$ , and the resulting ellipsoid is  $E_\Lambda(0, r) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^T \Lambda^{-1} \mathbf{y} \leq r^2\}$ . Generalizing the concept of a principal axis from  $\mathbb{R}^2$  and/or  $\mathbb{R}^3$  to  $\mathbb{R}^n$  it follows that  $E_\Lambda(0, r)$  has exactly  $n$  linearly independent principal axes given by  $r\sqrt{\lambda_i}\mathbf{u}_i$  for  $1 \leq i \leq n$  where  $\mathbf{u}_i \in \mathbb{R}^n$  is the  $i^{th}$  unit vector. Moreover, in the ellipsoidal norm  $\|\mathbf{x}\|_\Lambda$  the principal axes of  $E_\Lambda(0, r)$  have a length of  $r$ . Thus  $E_Q(\mathbf{x}_C, r)$  has  $n$  linearly independent principal axes of Euclidean length  $r\sqrt{\lambda_i}$  for  $1 \leq i \leq n$  as well that together with  $\mathbf{x}_C$  form a rectangular coordinate system for  $\mathbb{R}^n$ .

We can thus apply the same geometric thinking to ellipsoids in  $\mathbb{R}^n$  that we are used to apply to ellipses in  $\mathbb{R}^2$  or ellipsoids in  $\mathbb{R}^3$ . The last affine transformation is referred to as the *principal axis transformation*.

Like in  $\mathbb{R}^2$  and  $\mathbb{R}^3$  where ellipsoids have a certain *volume* the general ellipsoids in  $\mathbb{R}^n$  have a volume. More precisely, the volume of the unit cube  $C_n = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$  in  $\mathbb{R}^n$  is given by

$$\text{vol}(C_n) = \int \cdots \int_{C_n} dx_1 \cdots dx_n = 1.$$

If  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^n$  are affinely independent and  $S_x = \text{conv}(\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\})$  is the simplex (in general position) defined by  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n$  then its volume is given by

$$\text{vol}(S_x) = \int \cdots \int_{S_x} dx_1 \cdots dx_n = \frac{1}{n!} \left| \det \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ 1 & 1 & \cdots & 1 \end{pmatrix} \right|.$$

The volume of the unit sphere  $B = B(\mathbf{0}, 1)$  equals

$$\text{vol}(B) = \int \cdots \int_{B(\mathbf{0}, 1)} dx_1 \cdots dx_n = \frac{\pi^{n/2}}{\Gamma(1 + n/2)}, \quad (7.22)$$

where  $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$  for  $x > 0$  is the **gamma function**, which satisfies

$$\Gamma(x+1) = x\Gamma(x) \text{ for all } x, \Gamma(\frac{1}{2}) = \sqrt{\pi} \text{ and } \Gamma(n) = (n-1)!$$

for all integer  $n \geq 1$ . The volume of the ellipsoid  $E = E_Q(\mathbf{x}_C, r)$  is calculated using the affine transformation implied by the factorization of point 7.7(b) to equal

$$\text{vol}(E) = \int \cdots \int_E dx_1 \cdots dx_n = r^n |\det Q|^{1/2} \text{vol}(B) = \frac{r^n |\det Q|^{1/2} \pi^{n/2}}{\Gamma(1 + n/2)} = \left( \prod_{j=1}^n (r\sqrt{\lambda_j}) \right) \pi^{n/2} / \Gamma(1 + n/2), \quad (7.23)$$

where  $\lambda_1, \dots, \lambda_n$  are the positive eigenvalues of  $Q$ . This shows that the volume of an ellipsoid  $E$  is monotonically increasing in the blow-up factor  $r$  and the Euclidean lengths of the principal axes of  $E_Q(\mathbf{x}_C, 1)$ .

For every  $\mathbf{x} \in \mathbb{R}^n$  let  $m(\mathbf{x})$  be the *arithmetic mean* of  $\mathbf{x}$  and for every  $\mathbf{x} \in \mathbb{R}^n$  with  $\mathbf{x} > \mathbf{0}$  let  $g(\mathbf{x})$  be the *geometric mean* of  $\mathbf{x}$ , i.e.

$$m(\mathbf{x}) = (1/n) \sum_{i=1}^n x_i \quad \text{and} \quad g(\mathbf{x}) = \left( \prod_{i=1}^n x_i \right)^{1/n}.$$

**7.7(f) (Geometric/Arithmetic Mean Inequality)**  $(\prod_{i=1}^n x_i)^{1/n} \leq (\sum_{i=1}^n x_i)/n$  for all  $x \in \mathbb{R}^n$ ,  $x > 0$ , with equality if and only if  $x_i = \lambda$  for  $1 \leq i \leq n$  where  $\lambda \in \mathbb{R}$  is positive.

## 7.8 Exercises

---

### Exercise 7.1

Let  $P$  be a polyhedron,  $F_1$  be a  $k$ -dimensional face of  $P$  and  $F_2 \subseteq F_1$ .  $F_2$  is an  $h$ -dimensional face of  $F_1$  if and only if  $F_2$  is an  $h$ -dimensional face of  $P$  where  $h \leq k$ .

---

Let  $P = \{x \in \mathbb{R}^n : Hx \leq h\}$ . Since  $F_1$  is a  $k$  dimensional face of  $P$  we have by point 7.2(e)  $F_1 = \{x \in \mathbb{R}^n : H_1x = h_1, Hx \leq h\}$  where  $(H_1, h_1)$  is a submatrix of  $(H, h)$  with  $r(H_1) = n - k$ .

Suppose that  $F_2$  is an  $h$  dimensional face of  $P$ . Then  $F_2 = \{x \in \mathbb{R}^n : H_2x = h_2, Hx \leq h\}$  where  $(H_2, h_2)$  is a submatrix of  $(H, h)$  and  $r(H_2) = n - h$ . Since  $F_2 \subseteq F_1$  we have that  $F_2 = \{x \in F_1 : H_2x = h_2\}$ , i.e. there exists a partitioning of the constraint matrix of  $F_1$  such that  $F_2 = \{x \in F_1 : H_2x = h_2\}$  and  $r(H_2) = n - h$ . Thus by point 7.2(e)  $F_2$  is an  $h$  dimensional face of  $F_1$ .

On the other hand, suppose that  $F_2$  is an  $h$  dimensional face of  $F_1$ , i.e.  $F_2 = \{x \in F_1 : H_2x = h_2\}$  and  $r(H_2) = n - h$ . Then  $F_2 = \{x \in \mathbb{R}^n : H_1x = h_1, H_2x = h_2, Hx \leq h\}$  and  $r\begin{pmatrix} H_1 \\ H_2 \end{pmatrix} = n - h$  since  $F_2 \subseteq F_1$ . Thus  $(H, h)$  is partitioned accordingly and by point 7.2(e)  $F_2$  is an  $h$  dimensional face of  $P$ .

---

### Exercise 7.2

- (i) Given the polyhedron  $P = \{x \in \mathbb{R}^3 : x_2 - x_3 \leq 1, -x_2 - x_3 \leq -1\}$  find its lineality space  $L_P$  and an ideal description of  $P^0$ . Do the same for  $P = \{x \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, -x_1 - x_2 - x_3 \leq -1\}$  and for  $P = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 1\}$ . In which cases does every ideal description of  $P^0$  yield a linear description of  $P$ ?
  - (ii) The polyhedron  $C_n = \{x \in \mathbb{R}^n : 0 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$  is the  $n$ -dimensional unit cube. Show that  $\dim C_n = n$  and that its linear description is ideal.
  - (iii) Prove that the linear description of the  $n$ -dimensional simplex  $S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_{j=1}^n x_j \leq 1\}$  is ideal.
-

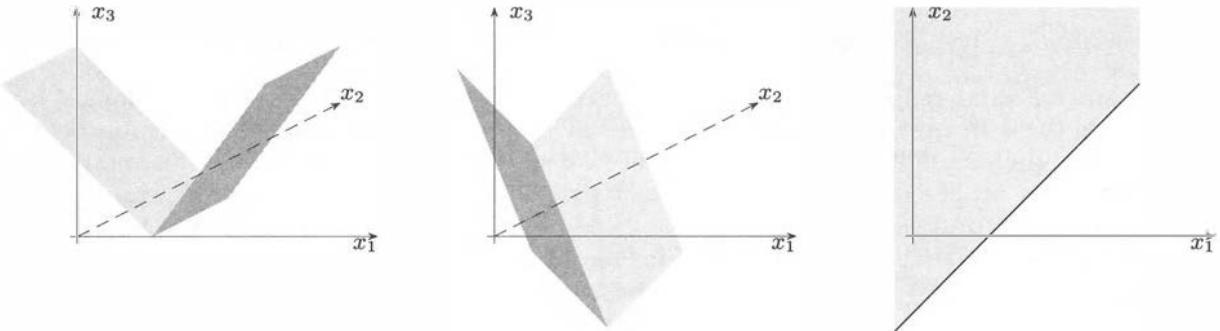


Fig. 7.4. The polyhedra of Exercise 7.2.(i)

(i) Let  $P = \{x \in \mathbb{R}^3 : x_2 - x_3 \leq 1, -x_2 - x_3 \leq -1\}$ . The linearity space is given by  $L_P = \{x \in \mathbb{R}^n : x_2 - x_3 = 0, -x_2 - x_3 = 0\}$ . A basis of  $L_P$  is  $\{(1, 0, 0)\}$  and thus  $L_P^\perp = \{x \in \mathbb{R}^3 : x_1 = 0\}$ . Hence a linear description of  $P^0$  is  $P^0 = P \cap L_P^\perp = \{x \in \mathbb{R}^3 : x_2 - x_3 \leq 1, -x_2 - x_3 \leq -1, x_1 = 0\}$  which is also minimal since removing any constraint changes  $P^0$ .

Let  $P = \{x \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, -x_1 - x_2 - x_3 \leq -1\}$ . The linearity space is given by  $L_P = \{x \in \mathbb{R}^n : x_1 + x_2 - x_3 = 0, -x_1 - x_2 - x_3 = 0\} = \{x \in \mathbb{R}^n : x_1 + x_2 = 0, x_3 = 0\}$ . A basis of  $L_P$  is  $\{(1, -1, 0)\}$  and thus  $L_P^\perp = \{x \in \mathbb{R}^3 : x_1 - x_2 = 0\}$ . Hence a linear description of  $P^0$  is  $P^0 = P \cap L_P^\perp = \{x \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, -x_1 - x_2 - x_3 \leq -1, x_1 - x_2 = 0\}$  which again is minimal since all inequalities are nonredundant.

Let  $P = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 1\}$ . The linearity space is given by  $L_P = \{x \in \mathbb{R}^2 : x_1 - x_2 = 0\}$ . A basis of the linearity space is  $\{(1, 1)\}$  and thus  $L_P^\perp = \{x \in \mathbb{R}^2 : x_1 + x_2 = 0\}$ . Hence a linear description of  $P^0$  is  $P^0 = P \cap L_P^\perp = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 1, x_1 + x_2 = 0\}$  which again is nonredundant and thus minimal.

In none of the cases does *every* ideal description of  $P^0$  yield a linear description of  $P$ . E.g.  $P^0 = \{x \in \mathbb{R}^2 : x_1 \leq 1/2, x_1 + x_2 = 0\} = \{x \in \mathbb{R}^2 : x_1 \leq 1/2\} \cap L_P^\perp \neq P \cap L_P^\perp$ , but the description  $x_1 \leq 1/2, x_1 + x_2 = 0$  is ideal for  $P^0$ .

(ii) We write  $C_n = \{x \in \mathbb{R}^n : Hx \leq h\}$ , where  $H = \begin{pmatrix} I_n \\ -I_n \end{pmatrix}$ ,  $h = \begin{pmatrix} e_n \\ 0_n \end{pmatrix}$ ,  $e_n$  and  $0_n$  are the vectors of

all ones and all zeros, respectively, in  $\mathbb{R}^n$ . Since  $r(H) = n$ , it follows that  $L_P = \{0\}$  and  $L_P^\perp = \mathbb{R}^n$ , hence  $P^0 = P$ . Since the point with  $x_j = 1/2$  for all  $1 \leq j \leq n$  is in  $C_n$  we have that  $H^\perp$  is empty and thus  $\dim C_n = n$ . To show that the linear description is ideal we show first that all inequalities are facet defining. Consider the inequality  $x_k \geq 0$  for some  $1 \leq k \leq n$  which defines the face  $F_k = \{x \in C_n : x_k = 0\}$ . Since the unit vector  $u_k \notin F_k$  and  $u_k \in C_n$ ,  $F$  is a proper face and since  $0 \in F$  it is a nonempty face. The  $n - 1$  unit vectors  $u_j$  for  $j = 1, \dots, n$ ,  $j \neq k$  and the zero vector are in  $F_k$  and are affinely independent. Thus  $\dim F_k = n - 1 = \dim C_n - 1$  and  $F_k$  is a facet of  $C_n$ . To show that inequalities  $x_k \leq 1$  define facets we show WROG that  $x_1 \leq 1$  does so. Consider the matrix  $X$  with rows  $x^1 = u_1^T$ ,  $x^i = u_1^T + u_i^T$  for  $i = 2, \dots, n$ . It follows that  $X$  is a lower triangular matrix and thus it has full rank. Thus the vectors  $x^i$  for  $i = 1, \dots, n$  are affinely independent and moreover, they lie on the face  $F_1 = \{x \in C_n : x_1 = 1\}$ . Hence  $\dim F_1 = n - 1 = \dim C_n - 1$ . Moreover,  $F_1$  is a proper face since  $0 \in C_n$  and  $0 \notin F_1$ . Thus  $F_1$  is a facet of  $C_n$ . Removing any of the inequalities  $0 \leq x_j$  or  $x_j \leq 1$  changes the polyhedron and thus the description of  $C_n$  is ideal.

(iii) We write  $S_n = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}\mathbf{x} \leq \mathbf{h}\}$  where  $\mathbf{H} = \begin{pmatrix} -I_n \\ \mathbf{e}_n^T \end{pmatrix}$ ,  $\mathbf{h} = \begin{pmatrix} \mathbf{0}_n \\ 1 \end{pmatrix}$ ,  $\mathbf{e}_n$  and  $\mathbf{0}_n$  are the vectors of all ones and all zeros, respectively, in  $\mathbb{R}^n$  and thus  $r(\mathbf{H}) = n$ . Since  $\mathbf{0} \in S_n$  and  $\mathbf{u}_k \in S_n$  for  $1 \leq k \leq n$  where  $\mathbf{u}_k$  is the  $k$ -th unit vector in  $\mathbb{R}^n$  we have  $\dim S_n = n$ . To prove that the description is ideal we prove that all inequalities are facet defining. Consider the inequality  $x_k \geq 0$  for some  $1 \leq k \leq n$  which defines a proper nonempty face  $F_k = \{\mathbf{x} \in S_n : x_k = 0\}$  of  $S_n$ , since  $\mathbf{0} \in F_k$ , and  $\mathbf{u}_k \notin F_k$ ,  $\mathbf{u}_k \in S_n$ . The  $n-1$  unit vectors  $\mathbf{u}_i$  for  $i = 1, \dots, n$ ,  $i \neq k$  and the zero vector form a set of  $n$  affinely independent vectors that lie on  $F_k$ . Thus  $\dim F_k = n-1 = \dim S_n - 1$  and  $F_k$  is a facet of  $S_n$ . Next consider the inequality  $\sum_{j=1}^n x_j \leq 1$  and let  $F$  be the face it defines, i.e.  $F = \{\mathbf{x} \in S_n : \sum_{j=1}^n x_j = 1\}$ . Since  $\mathbf{0} \in S_n$  and  $\mathbf{0} \notin F$ , and  $\mathbf{u}_1 \in F$ ,  $F$  is a nonempty proper face of  $S_n$ . The  $n$  unit vectors  $\mathbf{u}_i$ ,  $1 \leq i \leq n$  lie on  $F$  and are affinely independent. Thus  $\dim F = n-1 = \dim S_n - 1$  and  $F$  is a facet of  $S_n$ . Removing any of the defining inequalities changes the polyhedron and thus the description of  $S_n$  is ideal.

---

### Exercise 7.3

Given a polyhedron  $P = P(\mathbf{H}, \mathbf{h})$  let  $HP = HP(\mathbf{H}, \mathbf{h}) = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \mathbf{H}\mathbf{x} - \mathbf{h}x_{n+1} \leq 0, -x_{n+1} \leq 0\}$ . Let  $L_P$ ,  $L_{HP}$  be the lineality space of  $P$  and  $HP$ , respectively,  $P^0 = P \cap L_P^\perp$ ,  $HP^0 = HP \cap L_{HP}^\perp$  and remember that we write  $(\mathbf{x} \ x_{n+1})$  rather than  $\begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}$ . Let  $C_\infty = C_\infty(\mathbf{H})$  be the asymptotic cone of  $P$  and  $C_\infty^0 = C_\infty \cap L_P^\perp$ . Show:

- (i)  $\mathbf{x} \in L_P$  if and only if  $(\mathbf{x}, 0) \in L_{HP}$ .
  - (ii)  $\mathbf{x}^0$  is an extreme point of  $P^0$  if and only if  $((\mathbf{x}^0, 1))$  is an extreme ray of  $HP^0$ .
  - (iii)  $(\mathbf{x})$  is an extreme ray of  $C_\infty^0$  if and only if  $((\mathbf{x}, 0))$  is an extreme ray of  $HP^0$ .
  - (iv)  $\mathbf{h}^i \mathbf{x} \leq h_i$  is redundant for  $P$  if and only if  $\mathbf{h}^i \mathbf{x} - h_i x_{n+1} \leq 0$  is redundant for  $HP$ .
  - (v)  $\mathbf{h}^i \mathbf{x} = h_i$  is a valid equation for  $P$  if and only if  $\mathbf{h}^i \mathbf{x} - h_i x_{n+1} = 0$  for all  $(\mathbf{x}, 1) \in HP$ .
- 

(i) Suppose  $\mathbf{x} \in L_P$ , i.e.,  $\mathbf{H}\mathbf{x} = \mathbf{0}$ . Then  $\mathbf{H}\mathbf{x} - \mathbf{h}x_{n+1} = \mathbf{0}$  with  $x_{n+1} = 0$ , i.e.  $(\mathbf{x}, 0) \in L_{HP}$ . On the other hand, if  $(\mathbf{x}, 0) \in L_{HP}$ , we have  $\mathbf{H}\mathbf{x} - \mathbf{h}x_{n+1} = \mathbf{0}$  and  $x_{n+1} = 0$ . Eliminating  $x_{n+1} = 0$  we get  $\mathbf{H}\mathbf{x} = \mathbf{0}$ , i.e.  $\mathbf{x} \in L_P$ .

(ii) First we note that if the rows of  $G$  form a basis of  $L_P$  then  $L_P^\perp = \{\mathbf{x} \in \mathbb{R}^n : G\mathbf{x} = \mathbf{0}\}$  and by part (i)  $L_{HP}^\perp = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : G\mathbf{x} + 0x_{n+1} = \mathbf{0}\} = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : G\mathbf{x} = \mathbf{0}\}$ . We write  $HP^0 = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \mathbf{H}' (\mathbf{x} \ x_{n+1}) \leq \mathbf{0}, G\mathbf{x} = \mathbf{0}\}$ , where  $\mathbf{H}' = \begin{pmatrix} \mathbf{H} & -\mathbf{h} \\ \mathbf{0} & -1 \end{pmatrix}$ . It follows that  $r(\mathbf{H}') = r(\mathbf{H}) + 1$ . Suppose that  $((\mathbf{x}^0, 1))$  is an extreme ray of  $HP^0$ . Then by point 7.2( $\ell$ ) there exist  $r(\mathbf{H}') - 1$  linearly independent rows  $(\mathbf{h}^i, -h_i)$  of  $\mathbf{H}'$ , with  $1 \leq i \leq r(\mathbf{H}') - 1$ , such that  $\mathbf{h}^i \mathbf{x}^0 - h_i = 0$ . Note that the last row of  $\mathbf{H}'$  is not one of these rows. So, there exists a submatrix  $(\mathbf{H}^=, \mathbf{h}^=)$  of  $(\mathbf{H}, \mathbf{h})$  such that  $\mathbf{H}^= \mathbf{x}^0 = \mathbf{h}^=$  and  $r(\mathbf{H}^=) = r(\mathbf{H}') - 1 = r(\mathbf{H})$ . Thus the constraints

of  $P^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}\mathbf{x} \leq \mathbf{h}, \mathbf{G}\mathbf{x} = \mathbf{0}\}$  that are satisfied at equality by  $\mathbf{x}^0$  form a matrix with rank  $r(\mathbf{H}^=) + r(\mathbf{G}) = r(\mathbf{H}) + n - r(\mathbf{H}) = n$ . Hence by point 7.2(b)  $\mathbf{x}^0$  is an extreme point of  $P^0$ .

On the other hand, if  $\mathbf{x}^0$  is an extreme point of  $P^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}\mathbf{x} \leq \mathbf{h}, \mathbf{G}\mathbf{x} = \mathbf{0}\}$  then since  $r(\mathbf{G}) = n - r(\mathbf{H})$ , there exists a submatrix of  $(\mathbf{H}^=, \mathbf{h}^=)$  of  $(\mathbf{H}, \mathbf{h})$  such that  $r(\mathbf{H}^=) = r(\mathbf{H})$  and  $\mathbf{H}^=\mathbf{x}^0 = \mathbf{h}^=$ . Then  $(\mathbf{x}^0, 1)$  satisfies  $\mathbf{H}^=\mathbf{x} - \mathbf{h}_n x_{n+1} = \mathbf{0}$ ,  $x_{n+1} \geq 0$  and  $\mathbf{G}\mathbf{x} = \mathbf{0}$ , i.e.  $(\mathbf{x}^0, 1) \in HP^0 = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \mathbf{H}' \begin{pmatrix} \mathbf{x} & x_{n+1} \end{pmatrix} \leq \mathbf{0}, \mathbf{G}\mathbf{x} = \mathbf{0}\}$ . Since  $r(\mathbf{H}^=) = r(\mathbf{H}) = r(\mathbf{H}') - 1$ , there exist  $r(\mathbf{H}') - 1$  linearly independent rows  $(\mathbf{h}^i, -h_i)$  of  $\mathbf{H}'$  such that  $\mathbf{h}^i \mathbf{x}^0 - h_i = 0$ , i.e.  $((\mathbf{x}^0, 1))$  is an extreme ray of  $HP^0$ , by point 7.2(l).

**(iii)** As we showed in (ii), the linearity spaces of  $P$  and  $HP$  share the same basis. Thus we write  $C_\infty^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{H}\mathbf{x} \leq \mathbf{0}, \mathbf{G}\mathbf{x} = \mathbf{0}\}$  and  $HP^0 = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \mathbf{H}' \begin{pmatrix} \mathbf{x} & x_{n+1} \end{pmatrix} \leq \mathbf{0}, \mathbf{G}\mathbf{x} = \mathbf{0}\}$ ,

where  $\mathbf{H}' = \begin{pmatrix} \mathbf{H} & -\mathbf{h} \\ \mathbf{0} & -1 \end{pmatrix}$ . Suppose that  $(\mathbf{x})$  is an extreme ray of  $C_\infty^0$ . Then by point 7.2(l), there exist  $r(\mathbf{H}) - 1$  linearly independent rows  $i$  of  $\mathbf{H}$  such that  $\mathbf{h}^i \mathbf{x} = 0$ . The point  $(\mathbf{x}, 0) \in \mathbb{R}^{n+1}$  satisfies  $\mathbf{H}\mathbf{x} - \mathbf{h}_n x_{n+1} \leq \mathbf{0}$ ,  $-x_{n+1} \leq 0$ ,  $\mathbf{G}\mathbf{x} = \mathbf{0}$ , i.e.  $(\mathbf{x}, 0) \in HP^0$ . Moreover, it satisfies  $\mathbf{0}\mathbf{x} - x_{n+1} = 0$ , and since the last row of  $\mathbf{H}'$  is linearly independent from the  $r(\mathbf{H}) - 1$  rows of matrix  $\mathbf{H}$  such that  $\mathbf{h}^i \mathbf{x} = 0$ , it follows that there exist  $r(\mathbf{H}) = r(\mathbf{H}') - 1$  rows of  $\mathbf{H}'$  such that  $\mathbf{h}' \begin{pmatrix} \mathbf{x} & 0 \end{pmatrix} = 0$ , i.e.  $((\mathbf{x}, 0))$  is an extreme ray of  $HP^0$ .

On the other hand suppose that  $((\mathbf{x}, 0))$  is an extreme ray of  $HP^0$ . Then there exist  $r(\mathbf{H}') - 1$  linearly independent rows  $(\mathbf{h}')^i$  of  $\mathbf{H}'$  such that  $(\mathbf{h}')^i \begin{pmatrix} \mathbf{x} & 0 \end{pmatrix} = 0$ , i.e.  $\mathbf{h}\mathbf{x} = 0$ . Since  $r(\mathbf{H}') = r(\mathbf{H}) + 1$ , we have that  $r(\mathbf{H}) - 1$  of the linearly independent rows of  $\mathbf{H}'$  are rows of  $\mathbf{H}$  and thus  $(\mathbf{x})$  is an extreme ray of  $C_\infty^0$ .

**(iv)** Assuming that  $(\mathbf{H}, \mathbf{h})$  is partitioned like in (7.1), we have by point 7.2(h) that  $\mathbf{h}^i \mathbf{x} \leq h_i$  is redundant for  $P$  if and only if

$$\mathcal{U}_1 = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^m : \mathbf{u}\mathbf{H}^< + \mathbf{v}\mathbf{H}^= = \mathbf{0}, \mathbf{u}\mathbf{h}^< + \mathbf{v}\mathbf{h}^= \leq \mathbf{0}, u_i = -1, u_k \geq 0 \text{ for all } k \neq i\} \neq \emptyset.$$

By part (v) (see below) the sets of valid equalities for the two polyhedra are the same and thus the partitioning of the constraint matrix of  $HP$  is (row-wise) the same with that of  $(\mathbf{H}, \mathbf{h})$ . Applying point 7.2(h) we have that  $\mathbf{h}^i \mathbf{x} - h_n x_{n+1} \leq 0$  is redundant for  $HP$  if and only if

$$\mathcal{U}_2 = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^m : \mathbf{u}\mathbf{H}^< + \mathbf{v}\mathbf{H}^= = \mathbf{0}, -\mathbf{u}\mathbf{h}^< - \mathbf{v}\mathbf{h}^= \geq \mathbf{0}, \mathbf{u}\mathbf{0} + \mathbf{v}\mathbf{0} \leq \mathbf{0}, u_i = 1, u_k = 0 \text{ for all } k \neq i\} \neq \emptyset.$$

Since the third constraint of  $\mathcal{U}_2$  is trivially satisfied by all  $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^m$  and the second constraint can be written as  $\mathbf{u}\mathbf{h}^< + \mathbf{v}\mathbf{h}^= \leq \mathbf{0}$  we have  $\mathcal{U}_1 = \mathcal{U}_2$  and thus the assertion follows.

**(v)** First we show that  $\mathbf{x} \in P$  if and only if  $(\mathbf{x}, 1) \in HP$ . Suppose that  $(\mathbf{x}, 1) \in HP$ . Then  $\mathbf{H}\mathbf{x} - \mathbf{h} \leq \mathbf{0}$ , i.e.  $\mathbf{H}\mathbf{x} \leq \mathbf{h}$  and thus  $\mathbf{x} \in P$ . On the other hand, suppose that  $\mathbf{x} \in P$ . Then  $\mathbf{H}\mathbf{x} \leq \mathbf{h}$  and thus  $\mathbf{H}\mathbf{x} - \mathbf{h} \leq \mathbf{0}$  and since  $x_{n+1} = 1 > 0$ , we have  $(\mathbf{x}, 1) \in HP$ , and the claim follows. Now assume that  $\mathbf{h}^i \mathbf{x} = h_i$  is a valid equality for  $P$ , i.e.  $\mathbf{h}^i \mathbf{x} = h_i$  for all  $\mathbf{x} \in P$ , and suppose that there exists  $(\mathbf{x}^*, 1) \in HP$  satisfying  $\mathbf{h}^i \mathbf{x} - h_n x_{n+1} \neq 0$ , i.e.  $\mathbf{h}^i \mathbf{x}^* \neq h_i$ , which contradicts the validity of the equality, since by the previous claim  $(\mathbf{x}^*, 1) \in HP$  implies  $\mathbf{x}^* \in P$ . On the other hand, assume that  $\mathbf{h}^i \mathbf{x} - h_n x_{n+1} = 0$  for all  $(\mathbf{x}, 1) \in HP$  and suppose that there exists  $\mathbf{x}^* \in P$  such that  $\mathbf{h}^i \mathbf{x}^* \neq h_i$ . Then  $(\mathbf{x}^*, 1) \in HP$  and thus  $\mathbf{h}^i \mathbf{x}^* - h_i \neq 0$  which contradicts the assumption.

**Exercise 7.4**

- (i) Let  $S \subseteq \mathbb{R}^n$  be a convex set, i.e. for all  $\mathbf{x}^1, \mathbf{x}^2 \in S$  and all  $0 \leq \mu \leq 1$  we have  $\mu\mathbf{x}^1 + (1-\mu)\mathbf{x}^2 \in S$ . Show that  $S$  is convex if and only if for all  $1 \leq t < \infty$ ,  $\mathbf{x}^1, \dots, \mathbf{x}^t \in S$  and all  $\boldsymbol{\mu} \in \mathbb{R}^t$  such that  $\boldsymbol{\mu} \geq \mathbf{0}$  and  $\sum_{i=1}^t \mu_i = 1$  we have  $\sum_{i=1}^t \mu_i \mathbf{x}^i \in S$ . (Hint: Use induction on  $t$  and the fact that  $(1 - \mu_t)^{-1} \sum_{i=1}^{t-1} \mu_i = 1$  if  $\mu_t \neq 1$  and  $\sum_{i=1}^t \mu_i = 1$ .)
- (ii) Let  $S, T \subseteq \mathbb{R}^n$  be two convex sets. Show that  $S+T = \{z \in \mathbb{R}^n : z = \mathbf{x} + \mathbf{y} \text{ for some } \mathbf{x} \in S, \mathbf{y} \in T\}$  is convex.
- 

**(i)** Suppose that  $S$  is a convex set. For  $t = 1$ ,  $\mu_1 = 1$  we get  $\mathbf{x}^1 \in S$ . For  $t = 2$ ,  $\mu_1 + \mu_2 = 1$  we get  $\mu_1 \mathbf{x}^1 + \mu_2 \mathbf{x}^2 = \mu_1 \mathbf{x}^1 + (1 - \mu_1) \mathbf{x}^2 \in S$  from the definition of a convex set. Assume that the assertion is true for  $t = k$ , i.e. for all  $1 \leq k < \infty$ ,  $\mathbf{x}^1, \dots, \mathbf{x}^k \in S$  and all  $\boldsymbol{\mu} \in \mathbb{R}^k$  such that  $\boldsymbol{\mu} \geq \mathbf{0}$  and  $\sum_{i=1}^k \mu_i = 1$  we have  $\sum_{i=1}^k \mu_i \mathbf{x}^i \in S$ . We claim that the assertion is true for  $t = k + 1$ . Let  $\boldsymbol{\mu} \in \mathbb{R}^{k+1}$  be such that  $\sum_{i=1}^{k+1} \mu_i = 1$ . If  $\mu_{k+1} = 0$  then the assertion is true by the inductive hypothesis. If  $\mu_{k+1} = 1$  the assertion follows since  $\mathbf{x}^{k+1} \in S$ . So assume  $0 < \mu_{k+1} < 1$ . Then  $1 - \mu_{k+1} \neq 0$  and from  $\sum_{i=1}^{k+1} \mu_i = 1$  we calculate  $(1 - \mu_{k+1})^{-1} \sum_{i=1}^k \mu_i = 1$  and by the inductive hypothesis the point  $\mathbf{x}' = (1 - \mu_{k+1})^{-1} \sum_{i=1}^k \mu_i \mathbf{x}^i \in S$ . Since  $S$  is convex and  $\mathbf{x}', \mathbf{x}^{k+1} \in S$  we have that  $(1 - \mu_{k+1})\mathbf{x}' + \mu_{k+1}\mathbf{x}^{k+1} \in S$  and after substituting  $\mathbf{x}'$  we get  $\sum_{i=1}^{k+1} \mu_i \mathbf{x}^i \in S$  which proves the assertion.

On the other hand, suppose that for all  $1 \leq t < \infty$ ,  $\mathbf{x}^1, \dots, \mathbf{x}^t \in S$  and all  $\boldsymbol{\mu} \in \mathbb{R}^t$  such that  $\boldsymbol{\mu} \geq \mathbf{0}$  and  $\sum_{i=1}^t \mu_i = 1$  we have  $\sum_{i=1}^t \mu_i \mathbf{x}^i \in S$ . For  $t = 2$  we get the definition of a convex set and thus  $S$  is convex.

**(ii)** We have to show that  $\mathbf{z} = \mu\mathbf{z}^1 + (1 - \mu)\mathbf{z}^2 \in S + T$  where  $\mathbf{z}^1, \mathbf{z}^2 \in S + T$  and  $0 \leq \mu \leq 1$ . Since  $\mathbf{z}^1 \in S + T$  there exist  $\mathbf{x}^1 \in S$  and  $\mathbf{y}^1 \in T$  such that  $\mathbf{z}^1 = \mathbf{x}^1 + \mathbf{y}^1$  and similarly for  $\mathbf{z}^2$  there exist  $\mathbf{x}^2 \in S$  and  $\mathbf{y}^2 \in T$  such that  $\mathbf{z}^2 = \mathbf{x}^2 + \mathbf{y}^2$ . Thus we have  $\mathbf{z} = \mu\mathbf{z}^1 + (1 - \mu)\mathbf{z}^2 = \mu\mathbf{x}^1 + \mu\mathbf{y}^1 + (1 - \mu)\mathbf{x}^2 + (1 - \mu)\mathbf{y}^2 = \mathbf{x} + \mathbf{y}$  where we have set  $\mathbf{x} = \mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2$  and  $\mathbf{y} = \mu\mathbf{y}^1 + (1 - \mu)\mathbf{y}^2$ . From the convexity of  $S$  and  $T$  it follows that  $\mathbf{x} \in S$  and  $\mathbf{y} \in T$ , respectively, and thus  $\mathbf{z} \in S + T$ .

---

**Exercise 7.5**

- (i) Let  $\mathcal{XY} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p+q} : \mathbf{Hx} + \mathbf{Gy} \leq \mathbf{h}\}$  where  $\mathbf{H}$  and  $\mathbf{G}$  are matrices of size  $m \times p$  and  $m \times q$ , respectively. Show that under the projection  $\mathbf{z} = (\mathbf{I}_p \ \mathbf{O})(\mathbf{x}, \mathbf{y})$  from  $\mathbb{R}^{p+q}$  onto the subspace  $\mathbb{R}^p$  the image of  $\mathcal{XY}$  is given by  $\mathcal{X} = \{\mathbf{z} \in \mathbb{R}^p : \mathbf{vH}\mathbf{z} \leq \mathbf{vh} \text{ for all extreme rays } \mathbf{v} \in C\}$  and that  $C = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{vG} = \mathbf{0}, \mathbf{v} \geq \mathbf{0}\}$  is a pointed polyhedral cone.
- (ii) Let  $\mathcal{XY}^+ = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p+q} : \mathbf{Hx} + \mathbf{Gy} = \mathbf{h}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}\}$ . Show that under the same projection  $\mathbf{z} = (\mathbf{I}_p \ \mathbf{O})(\mathbf{x}, \mathbf{y})$  as in part (i) the image is  $\mathcal{X}^+ = \{\mathbf{z} \in \mathbb{R}^p : \mathbf{z} \geq \mathbf{0}, \mathbf{vH}\mathbf{z} \leq \mathbf{vh} \text{ for all } \mathbf{v} \in C\}$  where  $C = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{vG} \geq \mathbf{0}\}$ . Use this to show that the projection of  $\mathcal{X} = \{(\mathbf{x}, \mathbf{s}) \in \mathbb{R}^{n+m} : \mathbf{Ax} + \mathbf{s} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}\}$  onto the subspace of  $x$ -variables yields precisely  $\mathcal{X}^\leq = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  where  $\mathbf{A}$  is of size  $m \times n$ .

- (iii) Let  $\mathcal{X} = \{x \in \mathbb{R}^3 : -x_1 + 2x_3 \leq 1, x_2 + 2x_3 \geq 1, 0 \leq x_3 \leq 1, x_1 \geq 0, x_2 \geq 0\}$ . Show that the image of  $\mathcal{X}$  under the projection  $z_1 = x_1, z_2 = x_2$  is the nonnegative orthant  $\mathbb{R}_+^2$ . Show that the image of  $\mathcal{X}$  under the transformation  $z_1 = x_1 + x_2, z_2 = x_3$  is given by  $z_1 + 2z_2 \geq 1, -z_1 + 2z_2 \leq 1, z_1 \geq 0, 0 \leq z_2 \leq 1$ .
- (iv) Let  $\mathcal{X} = \{x \in \mathbb{R}^3 : -x_1 + x_3 \leq 0, x_2 + x_3 \geq 1, 0 \leq x_3 \leq 1\}$ . Show that the image of  $\mathcal{X}$  under the projection  $z_1 = x_1, z_2 = x_2$  is given by  $z_1 + z_2 \geq 1, z_1 \geq 0, z_2 \geq 0$ . Show that the image of  $\mathcal{X}$  under the transformation  $z_1 = x_1 + x_2, z_2 = x_3$  is given by  $z_1 \geq 1, 0 \leq z_2 \leq 1$ .
- (v) Show that the image of a polytope under an affine transformation is a polytope.
- 

**(i)** From (7.8) with  $L_1 = I_p, L_2 = \mathbf{O}, f = \mathbf{0}$  and void matrix  $H^\pm$ , we get that the image  $\mathcal{X}$  of  $\mathcal{XY}$  is

$$\mathcal{X} = \{z \in \mathbb{R}^n : uHz \leq uh \text{ for all } u \in C\},$$

where  $C = \{u \in \mathbb{R}^m : uG = \mathbf{0}, u \geq \mathbf{0}\}$ . The lineality space of  $C$  is  $L_C = \{\mathbf{0}\}$  and thus  $C$  is a pointed cone. Thus every  $v \in C$  can be written as a nonnegative linear combination of the extreme rays of  $C$ . Hence  $\mathcal{X} = \{z \in \mathbb{R}^p : vHz \leq vh \text{ for all extreme rays } v \in C\}$  since all other inequalities are redundant.

**(ii)** We write  $\mathcal{XY}^\pm = \{s \in \mathbb{R}^{p+q} : Ks \leq k\}$  where  $s = (x, y)$ ,

$$K = \begin{pmatrix} H & G \\ -I_p & \mathbf{O} \\ \mathbf{O} & -I_q \end{pmatrix}, \quad f = \begin{pmatrix} h \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$

and the first  $m$  rows of  $(K, k)$  correspond to  $(H^\pm, h^\pm)$  in the partitioning of (7.1). Now from (7.8) with  $L_1 = I_p, L_2 = \mathbf{O}$  and  $f = \mathbf{0}$  we get that the image  $\mathcal{X}^\pm$  of  $\mathcal{XY}^\pm$  is given by

$$\mathcal{X}^\pm = \{z \in \mathbb{R}^p : (vH - u)z \leq vh \text{ for all } (v, u, w) \in C'\},$$

where  $C' = \{(v, u, w) \in \mathbb{R}^{m+p+q} : vG - w = \mathbf{0}, u \geq \mathbf{0}, w \geq 0\}$ . We can replace the cone  $C'$  by  $C'' = \{(v, u) \in \mathbb{R}^{m+p} : vG \geq \mathbf{0}, u \geq \mathbf{0}\}$  since for every  $(v, u) \in C''$  we have  $(v, u, vG) \in C'$  and vice versa. The lineality space  $L_C$  of  $C''$  is given by  $\{(v, 0) \in \mathbb{R}^{m+p} : vG = \mathbf{0}\}$ . Every vector  $(0, u^i)$  is an extreme ray of the cone  $C^0 = C'' \cap L_C^\perp$  where  $u^i$  is the  $i$ th unit vector of  $\mathbb{R}^p$  and  $1 \leq i \leq p$ . Moreover, it follows from rank considerations that every extreme ray  $(v, u)$  of  $C^0$  with  $u \neq \mathbf{0}$  is of this form. Consequently,

$$\mathcal{X}^\pm = \{z \in \mathbb{R}^p : z \geq \mathbf{0}, vHz \leq vh \text{ for all } v \in C\},$$

where  $C = \{v \in \mathbb{R}^m : vG \geq \mathbf{0}\}$ . We can describe  $\mathcal{X}^\pm$  also as follows: Let  $v^1, \dots, v^r$  be a basis of the lineality space  $L_C = \{v \in \mathbb{R}^m : vG = \mathbf{0}\}$  of  $C$  and  $v^{r+1}, \dots, v^s$  the extreme rays of  $C \cap L_C^\perp$ . Then

$$\mathcal{X}^\pm = \{z \in \mathbb{R}^p : z \geq \mathbf{0}, v^i Hz = v^i h \text{ for } 1 \leq i \leq r, v^i Hz \leq v^i h \text{ for } r+1 \leq i \leq s\}.$$

For  $\mathcal{X} = \{(x, s) \in \mathbb{R}^{n+m} : Ax + s = b, x \geq \mathbf{0}, s \geq \mathbf{0}\}$  we apply the above result with  $H = A$ ,  $G = I_m$  and  $h = b$  to get that

$$\mathcal{X}^\leq = \{z \in \mathbb{R}^n : z \geq \mathbf{0}, vAz \leq vb \text{ for all } v \in C\} \quad \text{where } C = \{v \in \mathbb{R}^m : v \geq \mathbf{0}\}.$$

Thus the cone  $C$  is pointed, the extreme rays of  $C$  are the unit vectors  $\mathbf{u}^i$  of  $\mathbb{R}^m$  for  $i = 1, \dots, m$  and hence we have  $\mathcal{X}^\leq = \{z \in \mathbb{R}^n : Az \leq b, z \geq 0\}$ .

(iii) We write  $\mathcal{X} = \{x \in \mathbb{R}^3 : Hx \leq h\}$  where we have set

$$H = \begin{pmatrix} -1 & 0 & 2 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad h = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Applying the transformation with  $L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ ,  $L_1 = I_2$ ,  $L_2 = 0$  the image  $P$  of  $\mathcal{X}$  is given by

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : (-u_1 - u_4)z_1 + (-u_2 - u_5)z_2 \leq u_1 - u_2 + u_3 \text{ for all extreme rays } \mathbf{u} \in C\},$$

where  $C = \{\mathbf{u} \in \mathbb{R}^6 : 2u_1 - 2u_2 + u_3 - u_6 = 0, \mathbf{u} \geq 0\}$ . To find the extreme rays of  $C$ , we first simplify the cone by eliminating the variable  $u_6$ , i.e.  $C = \{\mathbf{u} \in \mathbb{R}^5 : 2u_1 - 2u_2 + u_3 \geq 0, \mathbf{u} \geq 0\}$ . The extreme rays of  $C$  are  $\mathbf{u}^4$ ,  $\mathbf{u}^5$ ,  $\mathbf{u}^1 + \mathbf{u}^2$ ,  $\mathbf{u}^3$ ,  $\mathbf{u}^1$ ,  $\mathbf{u}^2 + 2\mathbf{u}^3$  and give rise to the inequalities  $-z_1 \leq 0$ ,  $-z_2 \leq 0$ ,  $-z_1 - z_2 \leq 0$ ,  $0 \leq 1$ ,  $-z_1 \leq 0$ ,  $-z_2 \leq 2$ , respectively. The intersection of these inequalities is  $z_1 \geq 0$ ,  $z_2 \geq 0$  and thus  $P = \{z \in \mathbb{R}^2 : z \geq 0\}$ .

For the transformation  $z_1 = x_1 + x_2$ ,  $z_2 = x_3$ , we have

$$L = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad L_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

where  $x_2$  corresponds to  $L_2$  and thus the corresponding partition of the constraint matrix is

$$H_1^< = \begin{pmatrix} 0 & 2 \\ -1 & -2 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H_2^< = \begin{pmatrix} -1 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}.$$

With this information we calculate that the image  $P$  of  $\mathcal{X}$  is given by

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : (-u_2 - u_5)z_1 + (2u_1 - 2u_2 + u_3 - u_6)z_2 \leq u_1 - u_2 + u_3 \text{ for all extreme rays } \mathbf{u} \in C\},$$

where  $C = \{\mathbf{u} \in \mathbb{R}^6 : -u_1 + u_2 - u_4 + u_5 = 0, \mathbf{u} \geq 0\}$ . To calculate the extreme rays of  $C$ , we first eliminate variable  $u_4$  to get  $C = \{\mathbf{u} = (u_1, u_2, u_3, u_5, u_6) \in \mathbb{R}^5 : -u_1 + u_2 + u_5 \geq 0, \mathbf{u} \geq 0\}$ . The extreme rays of  $C$  are  $(0, 0, 1, 0, 0)$ ,  $(0, 0, 0, 0, 1)$ ,  $(1, 1, 0, 0, 0)$ ,  $(1, 0, 0, 1, 0)$ , and  $(0, 1, 0, 0, 0)$ , which give rise to the inequalities  $z_2 \leq 1$ ,  $-z_2 \leq 0$ ,  $-z_1 \leq 0$ ,  $-z_1 + 2z_2 \leq 1$ , and  $-z_1 - 2z_2 \leq -1$ , respectively. Thus we get

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : -z_1 + 2z_2 \leq 1, z_1 + 2z_2 \geq 1, 0 \leq z_2 \leq 1, z_1 \geq 0\}.$$

**(iv)** The matrix of the transformation  $z_1 = x_1$ ,  $z_2 = x_2$  is  $\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$  and  $\mathbf{L}_1 = \mathbf{I}_2$ ,  $\mathbf{L}_2 = \mathbf{0}$ . Thus we have that the image  $P$  of  $\mathcal{X}$  is given by

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : -u_1 z_1 - u_2 z_2 \leq -u_2 + u_3 \text{ for all extreme rays } \mathbf{u} \in C\}$$

where  $C = \{\mathbf{u} \in \mathbb{R}^4 : u_1 - u_2 + u_3 - u_4 = 0, \mathbf{u} \geq \mathbf{0}\}$ . To find the extreme rays of  $C$  we simplify the cone by eliminating variable  $u_4$  to get  $C = \{\mathbf{u} \in \mathbb{R}^3 : u_1 - u_2 + u_3 \geq 0, \mathbf{u} \geq \mathbf{0}\}$ . The extreme rays of  $C$  are:  $(1, 0, 0)$ ,  $(0, 0, 1)$ ,  $(1, 1, 0)$ , and  $(0, 1, 1)$ , which give rise to the inequalities  $-z_1 \leq 0$ ,  $0 \leq 1$ ,  $-z_1 - z_2 \leq -1$ , and  $-z_2 \leq 0$ , respectively. Thus we have

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : z_1 + z_2 \geq 1, z_1 \geq 0, z_2 \geq 0\}.$$

For the transformation  $z_1 = x_1 + x_2$ ,  $z_2 = x_3$  we proceed as in (iii) to find

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : -u_2 z_1 + (u_1 - u_2 + u_3 - u_4) z_2 \leq -u_2 + u_3 \text{ for all extreme rays } \mathbf{u} \in C\},$$

where  $C = \{\mathbf{u} \in \mathbb{R}^4 : u_1 - u_2 = 0, \mathbf{u} \geq \mathbf{0}\}$ . The extreme rays of  $C$  are  $(0, 0, 1, 0)$ ,  $(0, 0, 0, 1)$ , and  $(1, 1, 0, 0)$ , which give rise to the inequalities  $z_2 \leq 1$ ,  $-z_2 \leq 0$ , and  $-z_1 \leq -1$ , respectively. Thus we have

$$P = \{(z_1, z_2) \in \mathbb{R}^2 : z_1 \geq 1, 0 \leq z_2 \leq 1\}.$$

**(v)** By point 7.3(b), since  $P$  is a polytope, we have  $P = \text{conv}(S)$  where  $S = \{\mathbf{x}^1, \dots, \mathbf{x}^q\}$  is the set of extreme points of  $P$ . Let  $Q = \{z \in \mathbb{R}^p : \exists \mathbf{x} \in P \text{ such that } z = f + \mathbf{Lx}\}$  for some affine transformation  $z = f + \mathbf{Lx}$  mapping  $\mathbb{R}^n$  to  $\mathbb{R}^p$  with  $1 \leq p \leq n$ . By point 7.3(g)  $Q$  is a polyhedron. We show that  $Q = \text{conv}(R)$ . Let  $z \in Q$  and  $R = \{z^1, \dots, z^q\}$  where  $z^i = f + \mathbf{Lx}^i$  for  $1 \leq i \leq q$ . Then there exists  $\mathbf{x} \in P$  such that  $z = f + \mathbf{Lx}$ . Since  $P$  is a polytope,  $\mathbf{x} = \sum_{i=1}^q \mu_i \mathbf{x}^i$  with  $\mu_i \geq 0$  and  $\sum_{i=1}^q \mu_i = 1$ , thus  $z = \sum_{i=1}^q \mu_i z^i$ , i.e.  $z \in \text{conv}(R)$ , and hence  $Q \subseteq \text{conv}(R)$ . On the other hand, let  $z \in \text{conv}(R)$ . Then  $z = \sum_{i=1}^q \mu_i z^i = f + \mathbf{L}(\sum_{i=1}^q \mu_i \mathbf{x}^i) = f + \mathbf{Lx}$  for some  $\mathbf{x} \in P$ , i.e.  $\text{conv}(R) \subseteq Q$ , and thus  $Q = \text{conv}(R)$ . Consequently,  $Q$  is a bounded polyhedron and thus a polytope.

## Exercise 7.6

- (i) Find a minimal generator and a linear description for the polyhedron  $P \subseteq \mathbb{R}^2$  generated by  $(\{(1, 0), (1, 1), (0.5, 0.5), (0, 1)\}, \{(0.5, 1), (1, 1), (1, 0.5)\})$ .
- (ii) Let  $P \subseteq \mathbb{R}^2$  be the polyhedron generated by  $(\emptyset, \{(-1, 0), (0, -1), (-1, 2), (1, -2)\})$ . Show that  $(\{\mathbf{0}\}, T_0 \cup T_1)$  is a minimal generator for  $P$  where  $T_0 = \{(-1, 2)\}$  and  $T_1 = \{(-1, 0), (1, -2)\}$  and that  $2x_1 + x_2 \leq 0$  is an ideal description of  $P$ . Find a basis of its lineality space  $L_P$  and calculate the extreme points and extremal directions of the polyhedron  $P^0 = P \cap L_P^\perp$ .
- (iii) Do the same as under (ii) for the polyhedron  $P \subseteq \mathbb{R}^3$  that is generated by  $(\{(1, 0, 0)\}, \{(1, -1, 0), (-1, 1, 0), (1, 0, 1), (-1, 0, 1)\})$  and whose linear description is given by  $x_1 + x_2 - x_3 \leq 1$ ,  $-x_1 - x_2 - x_3 \leq -1$ . (Hint: Draw pictures in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , respectively.)

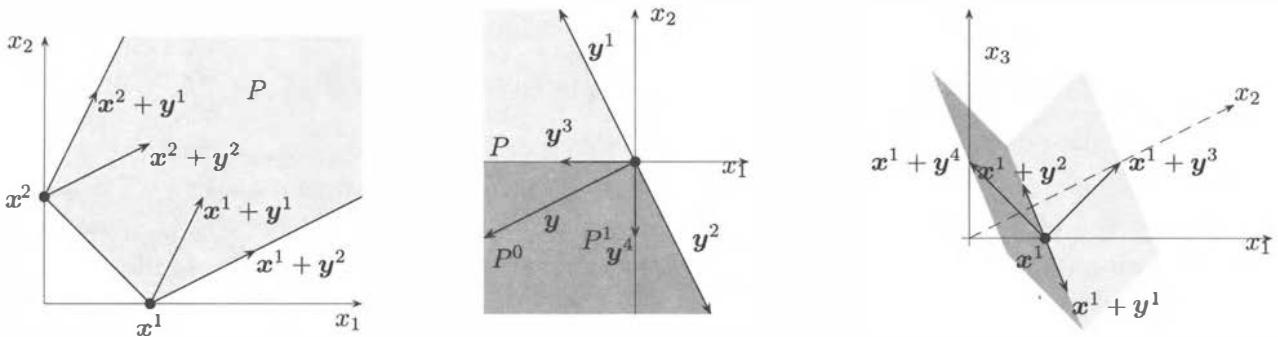


Fig. 7.5. The polyhedra of Exercise 7.6

**(i)** Let  $x^1 = (1, 0)$ ,  $x^2 = (0, 1)$ ,  $x^3 = (1, 1)$ ,  $x^4 = (0.5, 0.5)$  and  $y^1 = (0.5, 1)$ ,  $y^2 = (1, 0.5)$ ,  $y^3 = (1, 1)$ . Since  $\{\lambda \in \mathbb{R}^3 : \lambda_1 y^1 + \lambda_2 y^2 + \lambda_3 y^3 = 0, \lambda \geq 0\} = \{0\}$  it follows from point 7.3(j) that  $P = P(S, T)$  is line free where  $S = \{x^1, \dots, x^4\}$  and  $T = \{y^1, \dots, y^3\}$ . Since  $x^4 = \frac{1}{2}x^1 + \frac{1}{2}x^2$ ,  $x^3 = x^4 + \frac{1}{2}y^3$  and  $y^3 = \frac{2}{3}y^1 + \frac{2}{3}y^2$ , the points  $x^3$ ,  $x^4$  and  $y^3$  are not needed in a minimal generator for  $P$ . Consequently,  $P = P(\widehat{S}, \widehat{T})$  where  $\widehat{S} = \{x^1, x^2\}$  and  $\widehat{T} = \{y^1, y^2\}$ . Now  $x^1 = x^2 + \lambda_1 y^1 + \lambda_2 y^2$  has no solution with  $\lambda_1 \geq 0$ ,  $\lambda_2 \geq 0$ , nor does  $x^2 = x^1 + \lambda_1 y^1 + \lambda_2 y^2$ . Consequently,  $x^1$  and  $x^2$  are extreme points of  $P$  by point 7.3(l). Likewise neither  $y^1 = \lambda_1 y^2$  nor  $y^2 = \lambda_1 y^1$  has a nonnegative solution  $\lambda_1 \geq 0$  and thus  $y^1$ ,  $y^2$  are extremal directions of  $P$ . Consequently,  $(\widehat{S}, \widehat{T})$  is a minimal generator for  $P$ . Now we can “graph” the polyhedron  $P \subseteq \mathbb{R}^2$  and find that a linear description of  $P$  is given by  $x_1 + x_2 \geq 1$ ,  $-2x_1 + x_2 \leq 1$  and  $x_1 - 2x_2 \leq 1$ . In Chapter 7.4 you find an algebraic method to compute a linear description of a polyhedron from its pointwise description.

**(ii)** Let  $y^1 = (-1, 2)$ ,  $y^2 = (1, -2)$ ,  $y^3 = (-1, 0)$ ,  $y^4 = (0, -1)$  and  $S = \emptyset$ ,  $T = \{y^1, y^2, y^3, y^4\}$ . Then  $P = P(S, T)$  is a cone since  $S$  is empty. Since  $(1, 1, 0, 0) \in \{\lambda \in \mathbb{R}^4 : \lambda_1 y^1 + \lambda_2 y^2 + \lambda_3 y^3 + \lambda_4 y^4 = 0, \lambda \geq 0\}$ ,  $P$  has a line. Thus  $T_0 = \{y^1\}$ ,  $T_1 = \{y^2, y^3, y^4\}$  and iterating the procedure stated before point 7.3(k) we find that  $T_1$  cannot be reduced anymore. Consequently, the lineality space  $L_P$  of  $P$  is given by  $L_P = \text{cone}(\{y^1, -y^1\})$ ,  $P^1 = \text{cone}(T_1)$  is line free and the vector  $y^1 = (-1, 2)$  forms a basis of  $L_P$ . Since  $S = \emptyset$  we only need to check all points in  $T_1$  for extremality like in paragraph preceding point 7.3(l). Since  $y^4 = \frac{1}{2}y^2 + \frac{1}{2}y^3$  we can drop  $y^4$  from  $T_1$  and thus  $T_1 = \{y^2, y^3\}$ . Neither  $y^2 = \lambda_1 y^3$  nor  $y^3 = \lambda_1 y^2$  has a solution  $\lambda_1 \geq 0$  and thus  $y^2$ ,  $y^3$  are extremal directions of  $P^1$ , i.e.  $(\{0\}, T_0 \cup T_1)$  is a minimal generator for  $P$ . Now we can “graph”  $P^1$  and  $P$  and find that  $P = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \leq 0\}$  is a linear description of  $P$ . Moreover,  $L_P = \{x \in \mathbb{R}^2 : 2x_1 + x_2 = 0\}$  is the lineality space of  $P$  with basis  $y^1 = (-1, 2)$ . Forming  $P^0 = P \cap L_P^\perp$  we get  $P^0 = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \leq 0, -x_1 + 2x_2 = 0\}$ . Consequently,  $x = (0, 0)$  is the only extreme point of  $P^0$  and  $y = (-2, -1)$  the only extremal direction of  $P^0$ . It follows that  $P = \text{cone}(\{y^1, -y^1, y\})$  is also a minimal generator of  $P$ , which we call the “canonical” generator of  $P$ . Thus, in particular, minimal generators of polyhedra need not be unique.

**(iii)** Let  $x^1 = (1, 0, 0)$ ,  $y^1 = (1, -1, 0)$ ,  $y^2 = (-1, 1, 0)$ ,  $y^3 = (1, 0, 1)$ ,  $y^4 = (-1, 0, 1)$  and  $S = \{x^1\}$ ,  $T = \{y^1, y^2, y^3, y^4\}$  and  $P = P(S, T)$ . Since  $(1, 1, 0, 0) \in \{\lambda \in \mathbb{R}^4 : \lambda_1 y^1 + \lambda_2 y^2 + \lambda_3 y^3 + \lambda_4 y^4 = 0, \lambda \geq 0\}$ ,  $P$  has a line. Thus  $T_0 = \{y^1\}$ ,  $T_1 = \{y^2, y^3, y^4\}$  and iterating the procedure stated before point 7.3(k) we find that  $T_1$  cannot be reduced anymore. Consequently, the lineality space  $L_P$  of  $P$  is given by  $L_P = \text{cone}(y^1, -y^1)$ ,  $P^1 = \text{conv}(S) + \text{cone}(T_1)$  is line free and the vector  $y^1 = (1, -1, 0)$  forms

a basis of  $L_P$ . Since  $|S| = 1$  it follows that  $\mathbf{x}^1$  is the only extreme point of  $P^1$ . Checking the points in  $T_1$  for extremality like in the paragraph preceding point 7.3( $\ell$ ) we find that  $T_1$  cannot be reduced any further. Thus  $S^* = \{\mathbf{x}^1\}$ ,  $T^0 = \{\mathbf{y}^1\}$ ,  $T_1 = \{\mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\}$  is a minimal generator of  $P$  by point 7.3( $\ell$ ). Now we can “graph”  $P^1$  and  $P$  and find that  $P = \{\mathbf{x} \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, x_1 + x_2 + x_3 \geq 1\}$  is a linear description of  $P$ , where we have determined the two planes passing through  $\mathbf{x}^1$ ,  $\mathbf{x}^1 + \mathbf{y}^2$ ,  $\mathbf{x}^1 + \mathbf{y}^4$  and  $\mathbf{x}^1, \mathbf{x}^1 + \mathbf{y}^2, \mathbf{x}^1 + \mathbf{y}^3$ , respectively. Moreover,  $L_P = \{\mathbf{x} \in \mathbb{R}^3 : x_1 + x_2 - x_3 = 0, x_1 + x_2 + x_3 = 0\}$  is the linearity space of  $P$  with basis  $\mathbf{y}^1 = (1, -1, 0)$ . Forming  $P^0 = P \cap L_P^\perp$  we get  $P^0 = \{\mathbf{x} \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, x_1 + x_2 + x_3 \geq 1, x_1 - x_2 = 0\}$ . Consequently,  $\mathbf{x} = (0.5, 0.5, 0)$  is the only extreme point of  $P^0$  and  $\hat{\mathbf{y}}^1 = (0.5, 0.5, 1)$ ,  $\hat{\mathbf{y}}^2 = (-0.5, -0.5, 1)$  the two extremal directions of  $P^0$ . It follows that  $P = \mathbf{x} + \text{cone}(\{\mathbf{y}^1, -\mathbf{y}^1, \hat{\mathbf{y}}^1, \hat{\mathbf{y}}^2\})$ , i.e.  $\widehat{S} = \{\mathbf{x}\}$ ,  $\widehat{T} = \{\mathbf{y}^1, -\mathbf{y}^1, \hat{\mathbf{y}}^1, \hat{\mathbf{y}}^2\}$ , is also a minimal generator of  $P$ , which we call the “canonical” generator of  $P$ . Like in part (ii) we see that minimal generators of polyhedra need not be unique.

---

### Exercise 7.7

- (i) Let  $H_n$  be the  $2^n \times n$  matrix corresponding to all zero-one vectors in  $\mathbb{R}^n$  (including the zero vector). Show that a minimal generator  $(S, T)$  for the polyhedron  $H_n = \{\mathbf{x} \in \mathbb{R}^n : H_n \mathbf{x} \leq \mathbf{e}\}$  is given by  $S = \{\mathbf{u}^i \in \mathbb{R}^n : 1 \leq i \leq n\}$  and  $T = -S$ , where  $\mathbf{u}^i$  is the  $i^{\text{th}}$  unit vector in  $\mathbb{R}^n$  and  $\mathbf{e}$  is the vector with  $2^n$  components equal to 1. What is an ideal description of  $H_n$ ? (Hint: Use induction, the structure of the matrix  $H_n$  and point 7.3(n).)
  - (ii) Consider the polytope  $O_n = \text{conv}(\{\mathbf{x} \in \{0, 1\}^n : \sum_{j \in N} x_j = \text{odd}\})$  where  $N = \{1, \dots, n\}$  and  $\{0, 1\}^n = \{\mathbf{x} \in \mathbb{R}^n : x_j \in \{0, 1\} \text{ for all } j \in N\}$ . Show that  $O_n = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1 \text{ for all } j \in N, \sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j \leq |N_1| - 1 \text{ for all } N_1 \subseteq N \text{ with } |N_1| = \text{even}\}$  and that for all  $n \geq 4$  the linear description is ideal. (Hint: For any  $N_1 \subseteq N$  define  $\mathbf{a} \in \mathbb{R}^n$  by  $a_j = 1$  for  $j \in N_1$ ,  $a_j = 0$  otherwise. What is an ideal description of  $\text{conv}(\{\mathbf{x} \in \{0, 1\}^n : \mathbf{x} \neq \mathbf{a}\})$ ? )
  - (iii) Consider the polytope  $P_n^r = \text{conv}(\{\mathbf{x} \in \{0, 1\}^n : \sum_{j \in N} x_j = kr \text{ for some integer } k\})$  where  $n \geq r \geq 3$  are integers. Find an (ideal) linear description of  $P_n^r$ . [This problem is optional!]
  - (iv) Let  $\pi_P$  be a linear transformation on  $\mathbb{R}^n$  and  $S, T \subseteq \mathbb{R}^n$  be any sets. Show that  $\pi_P(\text{conv}(S)) = \text{conv}(\pi_P S)$ ,  $\pi_P(\text{cone}(T)) = \text{cone}(\pi_P T)$  and  $\pi_P(S + T) = \pi_P S + \pi_P T$ .
- 

**(i)** We write recursively  $H_n = \begin{pmatrix} H_{n-1} & \mathbf{0} \\ H_{n-1} & e_p \end{pmatrix}$  for all  $n \geq 2$  where  $H_1 = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$  and  $e_p$  is the column vector with  $p = 2^{n-1}$  entries equal to one. Clearly,  $r(H_1) = 1$ ; so suppose  $r(H_n) = n$  and let  $H$  be any submatrix of  $H_n$  with  $r(H) = n$ . Let  $\mathbf{h}$  be any row of  $H_n$ . Then the matrix

$$H' = \begin{pmatrix} H & \mathbf{0} \\ \mathbf{h} & 1 \end{pmatrix}$$

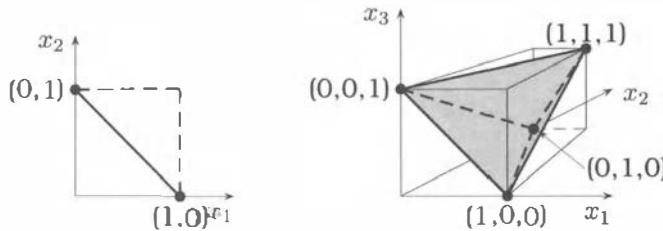
has rank  $r(H') = n + 1$  and thus  $r(H_{n+1}) = n + 1$  which proves that  $r(H_n) = n$  for all  $n \geq 1$ . Consequently the linearity space of  $H_n$  is the origin of  $\mathbb{R}^n$ . So  $H_n$  is line free, nonempty since

$\mathbf{0} \in H_n$  and thus pointed for all  $n \geq 1$ . Moreover, the linear transformation  $\pi_P$  given by (7.12) is the identity and point 7.3(n) simplifies accordingly. Let  $\mathbf{u}^i \in \mathbb{R}^n$  be the  $i$ th unit vector, where  $1 \leq i \leq n$ . Then  $H_n \mathbf{u}^i \leq \mathbf{e}$  and the  $n \times n$  submatrix  $H$  of  $H_n$  with rows  $\mathbf{h}^j = (\mathbf{u}^i + \mathbf{u}^j)^T$  for  $1 \leq j \neq i \leq n$  and  $\mathbf{h}^i = (\mathbf{u}^i)^T$ , satisfies  $H\mathbf{u}^i = \mathbf{e}_n$  and  $r(H) = n$ , since – modulo row and column permutations –  $H$  is lower triangular with ones on the main diagonal and thus nonsingular. Consequently, by point 7.3(n) all unit vectors  $\mathbf{u}^i \in \mathbb{R}^n$  are extreme points of  $H_n$ . On the other hand,  $H_n(-\mathbf{u}^i) \leq \mathbf{0}$  and the  $(n-1) \times n$  submatrix  $H$  of  $H_n$  with rows  $\mathbf{h}^j = (\mathbf{u}^j)^T$  for  $1 \leq j \neq i \leq n$  satisfies  $H(-\mathbf{u}^i) = \mathbf{0}$  and  $r(H) = n-1$ . Moreover, every other row  $\mathbf{h}$  of  $H_n$  with  $\mathbf{h}(-\mathbf{u}^i) = \mathbf{0}$  is a linear combination of the rows of  $H$  and thus by point 7.3(n) the vectors  $-\mathbf{u}^i \in \mathbb{R}^n$  are extremal directions of  $H_n$  for  $1 \leq i \leq n$ . Let  $\mathbf{u} \in \mathbb{R}^n$  be the direction vector of any halfline of  $H_n$ . It follows that  $\mathbf{u}^T = (u_1, \dots, u_n) \leq \mathbf{0}$ ,  $\mathbf{u} \neq \mathbf{0}$ , since  $H_n$  contains all  $n$  unit row vectors, and thus  $\mathbf{u} = \sum_{i=1}^n \lambda_i (-\mathbf{u}^i)$  with  $\lambda_i = |u_i| \geq 0$  which shows that  $-\mathbf{u}^1, \dots, -\mathbf{u}^n$  are precisely the extremal directions of  $H_n$ . Suppose that  $\mathbf{x} \in H_n$  and let  $\mathbf{x}^+ = \max\{\mathbf{0}, \mathbf{x}\}$ ,  $\mathbf{x}^- = \min\{\mathbf{0}, \mathbf{x}\}$ . It follows that  $\mathbf{x}^+ \in H_n$ . For, suppose not. Then  $\mathbf{h}\mathbf{x}^+ > 1$  and  $\mathbf{h}\mathbf{x}^+ + \mathbf{h}\mathbf{x}^- \leq 1$  for some row  $\mathbf{h}$  of  $H_n$  and thus  $h_j = 1$  for some  $j$  with  $x_j^- < 0$ . But the row  $\mathbf{h}^*$  with  $h_j^* = 0$  for all  $j$  with  $x_j^- < 0$ ,  $h_j^* = h_j$  otherwise is also in  $H_n$  and thus  $\mathbf{h}\mathbf{x}^+ = \mathbf{h}^*\mathbf{x}^+ \leq 1$  which is a contradiction. Consequently, every extreme point  $\mathbf{x} \in H_n$  satisfies  $\mathbf{x} \geq \mathbf{0}$ ,  $\mathbf{x} \neq \mathbf{0}$ , and thus  $\mathbf{x} = \sum_{j=1}^n x_j \mathbf{u}^j$ . Let  $\mathbf{h}$  be the row with  $h_j = 1$  for all  $x_j > 0$ ,  $h_j = 0$  otherwise. Since  $\mathbf{x}$  is an extreme point of  $H_n$ ,  $\mathbf{h}'\mathbf{x} = 1$  for some row  $\mathbf{h}'$  of  $H_n$  and thus clearly  $\mathbf{h}\mathbf{x} = 1$ . Hence  $\mathbf{x}$  is a convex combination of the extreme points  $\mathbf{u}^1, \dots, \mathbf{u}^n$  and consequently,  $(S = \{\mathbf{u}^1, \dots, \mathbf{u}^n\}, T = \{-\mathbf{u}^1, \dots, -\mathbf{u}^n\})$  is a minimal generator of  $H_n$ .

Let  $\mathbf{h} \neq \mathbf{0}$  be any nonnull row of  $H_n$  and  $N_1 = \{j \in N : h_j = 1\}$  where  $N = \{1, \dots, n\}$ . It follows that  $\mathbf{h}\mathbf{u}^j = 1$  for all  $j \in N_1$  and  $\mathbf{h}(\mathbf{u}^{j_0} - \mathbf{u}^k) = 1$  for all  $k \in N - N_1$  where  $j_0 \in N_1$  is arbitrary. The matrix  $\mathbf{U}$  with rows  $\mathbf{u}^j$  for  $j \in N_1$ ,  $\mathbf{u}^{j_0} - \mathbf{u}^k$  for  $k \in N - N_1$  is of size  $n \times n$  and  $r(\mathbf{U}) = n$ . Since  $\mathbf{u}^j \in H_n$  for  $j \in N_1$  and  $\mathbf{u}^{j_0} - \mathbf{u}^k \in H_n$  for  $k \in N - N_1$  it follows that  $F = \{\mathbf{x} \in H_n : \mathbf{h}\mathbf{x} = 1\}$  is a face of dimension  $n-1$  of  $H_n$  (see Definition FA, p. 123) and thus a facet of  $H_n$ , because  $\mathbf{0} \in H_n$  and  $\mathbf{h}\mathbf{0} < 1$ . Let  $\mathbf{h} \neq \mathbf{0}$  and  $\mathbf{h}' \neq \mathbf{0}$  be any two different nonnull rows of  $H_n$ . Then there exists  $j \in N$  such that e.g.  $h_j = 0$  and  $h'_j = 1$  or vice versa. Thus  $\mathbf{h}\mathbf{u}^j = 0$ , but  $\mathbf{h}'\mathbf{u}^j = 1$  shows that the facets defined by any two distinct nonnull rows of  $H_n$  are different. Consequently, letting  $H'_n$  be the submatrix of  $H_n$  that is obtained by deleting the zero row of  $H_n$  we have  $H_n = \{\mathbf{x} \in \mathbb{R}^n : H'_n \mathbf{x} \leq \mathbf{e}'\}$  and this linear description of  $H_n$  is ideal.

**(ii)** We first derive some facts about the unit-hypercube  $C_n = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$  which is a polytope in  $\mathbb{R}^n$ . By Exercise 7.2(ii) we know that  $\dim C_n = n$  and that its linear description is ideal for all  $n \geq 1$ . Using point 7.2(b) it is straightforward to show that every 0-1 vector is an extreme point of  $C_n$  and vice versa, i.e.  $C_n$  has precisely  $2^n$  extreme points and  $C_n = \text{conv}(\{0,1\}^n)$ . Using the second half of point 7.2(n) you prove that two extreme points  $\mathbf{x}^0, \mathbf{x}^1 \in C_n$  are adjacent if and only if they differ in exactly one component, i.e.  $\mathbf{x}^0 - \mathbf{x}^1 = \pm \mathbf{u}^j$  for some  $j \in N$  where  $\mathbf{u}^j$  is the  $j$ th unit vector in  $\mathbb{R}^n$ . Consequently, every zero-one vector  $\mathbf{x}^0 \in C_n$  has precisely  $n$  adjacent extreme points (the “neighbors” of  $\mathbf{x}^0$ ) which are of the form  $\mathbf{x}^0 - \mathbf{u}^j$  for  $j \in N_1$ ,  $\mathbf{x}^0 + \mathbf{u}^j$  for  $j \in N - N_1$  where  $N_1 = \{j \in N : x_j^0 = 1\}$ . Moreover, the neighbors of  $\mathbf{x}^0$  are affinely independent. It follows that all neighbors of  $\mathbf{x}^0 \in \{0,1\}^n$  with  $\sum_{j=1}^n x_j^0$  even have an odd number of ones and likewise, all neighbors of  $\mathbf{x}^0 \in \{0,1\}^n$  with  $\sum_{j=1}^n x_j^0$  odd, have an even number of ones. Since the  $n$  neighbors of  $\mathbf{x}^0 \in \{0,1\}^n$  are affinely independent they determine a hyperplane

$$\sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j = |N_1| - 1, \quad (1)$$



**Fig. 7.6.** The polyhedra  $O_2$  and  $O_3$  of Exercise 7.7(ii)

where  $N_1 = \{j \in N : x_j^0 = 1\}$ . Now let

$$C_n(\mathbf{x}^0) = C_n \cap \{\mathbf{x} \in \mathbb{R}^n : \sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j = |N_1| - 1\}.$$

Then evidently  $\mathbf{x}^0 \notin C_n(\mathbf{x}^0)$ , but  $\mathbf{x} \in C_n(\mathbf{x}^0)$  for all  $\mathbf{x} \in \{0,1\}^n$  with  $\mathbf{x} \neq \mathbf{x}^0$ . Here  $\mathbf{x}^0 \notin C_n(\mathbf{x}^0)$  because  $\sum_{j \in N_1} x_j^0 - \sum_{j \in N - N_1} x_j^0 = |N_1| > |N_1| - 1$  and the rest of the assertion follows likewise. Moreover, we claim

$$C_n(\mathbf{x}^0) = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1, \text{ for all } j \in N, \sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j \leq |N_1| - 1\}$$

has precisely  $2^n - 1$  extreme points which are all zero-one vectors. Since  $\mathbf{x}^0$  is the only zero-one vector that is “cut-off” by the inequality corresponding to (1),  $C_n(\mathbf{x}^0)$  has at least  $2^n - 1$  extreme points because  $C_n(\mathbf{x}^0) \subseteq C_n$  and thus every extreme point of  $C_n$  that belongs to  $C_n(\mathbf{x}^0)$  is *a fortiori* an extreme point of  $C_n(\mathbf{x}^0)$ . Suppose that  $\mathbf{x}^f \in C_n(\mathbf{x}^0)$  is an extreme point of  $C_n(\mathbf{x}^0)$  that is not zero-one valued. It follows that  $\mathbf{x}^f$  satisfies (1) and that  $\mathbf{x}^f$  is contained in a face  $F$  of dimension 1 of  $C_n$  that also contains  $\mathbf{x}^0$ . But then there must exist a neighbor  $\mathbf{x}^1$  of  $\mathbf{x}^0$  which satisfies (1) as a strict less-than inequality which is a contradiction since all neighbors of  $\mathbf{x}^0$  satisfy equation (1). Hence all extreme points of  $C_n(\mathbf{x}^0)$  are zero-one valued, i.e.

$$C_n(\mathbf{x}^0) = \text{conv}(\{0,1\}^n - \mathbf{x}^0)$$

and thus the linear description of  $\text{conv}(\{0,1\}^n - \mathbf{x}^0)$  given by  $C_n(\mathbf{x}^0)$  is complete. For  $n = 2$  the linear description is not minimal: e.g.  $x_1 \leq 1$  is implied by  $x_1 + x_2 \leq 1$  and  $x_2 \geq 0$  if  $\mathbf{x}^0 = (1,1)^T$ . However, for all  $n \geq 3$  the above linear description of  $C_n(\mathbf{x}^0)$  is minimal as well. You prove that by showing that all inequalities of the linear description define distinct facets of  $C_n(\mathbf{x}^0)$  for all  $n \geq 3$ .

Now we are ready to discuss the polyhedra  $O_n = \text{conv}(\{\mathbf{x} \in \{0,1\}^n : \sum_{j \in N} x_j = \text{odd}\})$ . Let

$$O'_n = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1, \forall j \in N, \sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j \leq |N_1| - 1, \forall N_1 \subseteq N, |N_1| \text{ even}\}. \quad (2)$$

Let  $\mathbf{x}^0 \in \{0,1\}^n$  with  $\sum_{j=1}^n x_j$  odd. Then  $\sum_{j \in N_1} x_j - \sum_{j \in N - N_1} x_j < |N_1|$  for all  $N_1 \subseteq N$  with  $|N_1|$  even and thus  $\mathbf{x}^0 \in O'_n$ . Consequently,  $O_n \subseteq O'_n$ . Since  $O'_n \subseteq C_n$  it follows as before that every  $\mathbf{x} \in O'_n \cap \{0,1\}^n$  is an extreme point of  $O'_n$ . Moreover, every zero-one extreme point of  $O'_n$  has an odd number of ones. Thus it suffices to show that  $O'_n$  has no *fractional* extreme points, i.e. that there exist no extreme points  $\mathbf{x}^f \in O'_n$  with  $0 < x_j^f < 1$  for some  $j \in N$ . We do so as follows.

Let  $E = \{\mathbf{x}^1, \dots, \mathbf{x}^p\}$  be the set of all zero-one vectors having an even number of ones where the indexing of  $\mathbf{x}^i$  for  $1 \leq i \leq p = 2^{n-1}$  is arbitrary. Note that by definition

$$O_n = \text{conv}(\{0, 1\}^n - E).$$

Now let  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^k) = \text{conv}(\{0, 1\}^n - E_k)$  where  $E_k = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$  for  $1 \leq k \leq p$ . We claim that

$$C_n(\mathbf{x}^1, \dots, \mathbf{x}^k) = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq 1, \text{ for all } j \in N, \sum_{j \in N_i} x_j - \sum_{j \in N - N_i} x_j \leq |N_i| - 1 \text{ for all } 1 \leq i \leq k\},$$

where  $N_i = \{j \in N : x_j^i = 1\}$  for all  $1 \leq i \leq p$ . For  $k = 1$  we know that the claim is true. So suppose that it is true for some  $1 \leq k < p$ . Then all extreme points of  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$  are zero-one valued and in particular,  $\mathbf{x}^{k+1}$  is a zero-one extreme point of this polytope. Since  $\mathbf{x}^{k+1}$  has an even number of ones,  $\mathbf{x}^{k+1}$  has exactly  $n$  affinely independent neighbors  $\mathbf{y}^1, \dots, \mathbf{y}^n$  having an odd number of ones on the unit cube  $C_n$  (= "odd" neighbors). But  $\mathbf{y}^i \in C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$  are extreme points and thus *a fortiori* neighbors of  $\mathbf{x}^{k+1}$  on the polytope  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$ . Suppose there exists a zero-one point  $\mathbf{y} \in C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$  that is adjacent to  $\mathbf{x}^{k+1}$  on the polytope  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$ , but not adjacent to  $\mathbf{x}^{k+1}$  on the unit cube  $C_n$ . It follows that

$$\sum_{j \in N_i} x_j^{k+1} - \sum_{j \in N - N_i} x_j^{k+1} = |N_i| - 1 \text{ for some } 1 \leq i \leq k, \quad (3)$$

which is impossible since  $N_{k+1} \neq N_i$  for all  $1 \leq i \leq k$ , the only zero-one vectors satisfying the equations (3) are the odd neighbors of  $\mathbf{x}^1, \dots, \mathbf{x}^k$  and  $\mathbf{x}^{k+1}$  has an even number of ones. Consequently,  $\mathbf{x}^{k+1}$  has exactly  $n$  affinely independent odd neighbors on the polytope  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^k)$ . We conclude thus like in the case of  $C_n(\mathbf{x}^0)$  that  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^{k+1})$  has only zero-one extreme points and thus the claim follows. Since  $C_n(\mathbf{x}^1, \dots, \mathbf{x}^p) = O'_n$  it follows from the induction that  $O'_n$  has only zero-one extreme points and thus  $O'_n = O_n$  for all  $n \geq 1$ . For  $n = 2$  and  $n = 3$

$$O_2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 = 1\},$$

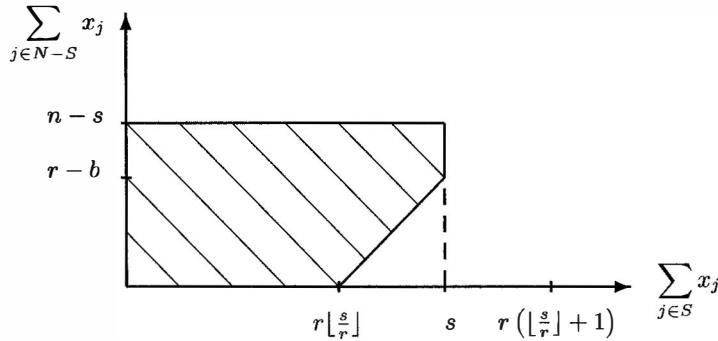
$$O_3 = \{\mathbf{x} \in \mathbb{R}^3 : x_1 + x_2 - x_3 \leq 1, x_1 - x_2 + x_3 \leq 1, -x_1 + x_2 + x_3 \leq 1, -x_1 - x_2 - x_3 \leq -1\}$$

are ideal, i.e. minimal and complete linear descriptions; see the figures. For  $n \geq 4$  we have shown completeness. Since  $\mathbf{u}^j \in O_n$  for  $1 \leq j \leq n$  and  $\mathbf{u}^1 + \mathbf{u}^2 + \mathbf{u}^3 \in O_n$  for all  $n \geq 3$  it follows that  $\dim O_n = n$  for all  $n \geq 3$ . You prove that for  $n \geq 4$  all inequalities of the linear description (2) define distinct facets of  $O_n$  and thus the linear description of  $O_n$  is ideal for all  $n \geq 4$ .

**(iii)** The polytope  $P_n^r$  can be written as  $P_n^r = \text{conv}\{\{0, 1\}^n : \sum_{j \in N} x_j \equiv 0 \pmod{r}\}$ . A class of polytopes where the congruence constraint is replaced by  $\sum_{j \in N} x_j \equiv t \pmod{r}$  where  $0 \leq t \leq r - 1$  has been studied in Alevras, D. and M. P Rijal "The convex hull of a linear congruence relation in zero-one variables", *ZOR-Mathematical Methods of Operations Research* (1995) 41, 1-23. The polytope  $P_n^r$  is derived as a special case when  $t = 0$ . We give here a summary of the results of this paper without proofs.

The ideal description of  $P_n^r$  is given by the following system of inequalities

$$\begin{aligned} x_j &\geq 0 && \text{for } j \in N, \text{ if } r \leq n - 2 \\ x_j &\leq 1 && \text{for } j \in N, \text{ if } n \geq 2r \\ \sum_{j \in N} x_j &\leq r \lfloor \frac{n}{r} \rfloor && \text{if } n \not\equiv 0 \pmod{r} \\ a \sum_{j \in S} x_j - b \sum_{j \in N - S} x_j &\leq c && \text{for } S \in \mathcal{N}_r \end{aligned} \quad (4)$$



**Fig. 7.7.** Illustration of divisibility conditions in Exercise 7.7(iii)

where  $a = r - b$ ,  $b = s - r\lfloor \frac{s}{r} \rfloor$ ,  $c = (s - b)(r - b)$ ,  $s = |S|$  and  $\mathcal{N}_r = \{S : \emptyset \neq S \subset N, s = |S| \text{ satisfies (5)}\}$ :

$$s = 1 \quad \text{or} \quad s = n - 1, \quad n \equiv 0 \pmod{r} \quad \text{or} \quad r < s < \min\{r\lfloor \frac{n}{r} \rfloor, n - 1\}, \quad \lfloor \frac{s}{r} \rfloor < \frac{n}{r} - 1, \quad s \not\equiv 0 \pmod{r}. \quad (5)$$

These inequalities were found using the double description algorithm of Chapter 7.4 for polytopes  $P_n^r$  with small  $n$  and varying  $r \leq n$ , and generalizing the derived descriptions for general  $n$  and  $r$ . Inequality (4) can be interpreted intuitively as follows. In the case  $s = 1$  it reads  $(r - 1)x_j \leq \sum_{k \in N-S} x_k$  and in the case that  $s = n - 1$  and  $n \equiv 0 \pmod{r}$  it can be brought into the form  $(r - 1)(1 - x_j) \leq \sum_{k \in N-S} (1 - x_k)$ , both of which speak for themselves. In the remaining case we have that  $|S| = s \not\equiv 0 \pmod{r}$  and that  $s$  satisfies the two other conditions. If  $\sum_{j \in S} x_j \leq r\lfloor \frac{s}{r} \rfloor$ , then inequality (4) imposes no restriction at all; if, however,  $\sum_{j \in S} x_j = r\lfloor \frac{s}{r} \rfloor + \tau$  with  $0 \leq \tau \leq b$ , then the inequality implies that  $\sum_{j \in N-S} x_j \geq (r - b)\tau/b$  and thus e.g. for  $\tau = b$  we get  $\sum_{j \in N-S} x_j \geq r - b$  which is necessary (but not sufficient) to obtain divisibility of  $\sum_{j \in N} x_j$  by the number  $r$ . In Figure 7.7 we give a graphical illustration of the linearization of the divisibility conditions, for  $n = 9$ ,  $r = 4$ ,  $s = 6$ .

**(iv)** From Definition HU we write

$$\begin{aligned} \pi_P(\text{conv}(S)) &= \pi_P \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \sum_{i=1}^t \mu_i \mathbf{x}^i \text{ where } \mu_i \geq 0, \sum_{i=1}^t \mu_i = 1, \mathbf{x}^i \in S \text{ for all } 1 \leq i \leq t, 0 \leq t < \infty \} \\ &= \{ \pi_P \mathbf{x} \in \mathbb{R}^n : \pi_P \mathbf{x} = \sum_{i=1}^t \mu_i \pi_P \mathbf{x}^i \text{ where } \mu_i \geq 0, \sum_{i=1}^t \mu_i = 1, \mathbf{x}^i \in S \text{ for all } 1 \leq i \leq t, 0 \leq t < \infty \} \\ &= \{ \mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \sum_{i=1}^t \mu_i \mathbf{z}^i \text{ where } \mu_i \geq 0, \sum_{i=1}^t \mu_i = 1, \mathbf{z}^i \in \pi_P S \text{ for all } 1 \leq i \leq t, 0 \leq t < \infty \} \\ &= \text{conv}(\pi_P S) \end{aligned}$$

since  $\pi_P(\mathbf{x} + \mathbf{y}) = \pi_P\mathbf{x} + \pi_P\mathbf{y}$  for any linear transformation in  $\mathbb{R}^n$ . Likewise, one shows that  $\pi_P(\text{cone}(T)) = \text{cone}(\pi_P T)$ . Furthermore, by part (ii) of Exercise 7.4

$$\begin{aligned}\pi_P(S + T) &= \pi_P\{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \mathbf{x} + \mathbf{y} \text{ for some } \mathbf{x} \in S, \mathbf{y} \in T\} \\ &= \{\pi_P\mathbf{z} \in \mathbb{R}^n : \pi_P\mathbf{z} = \pi_P\mathbf{x} + \pi_P\mathbf{y} \text{ for some } \mathbf{x} \in S, \mathbf{y} \in T\} \\ &= \{\xi \in \mathbb{R}^n : \xi = \eta + \zeta \text{ for some } \eta \in \pi_P S, \zeta \in \pi_P T\} = \pi_P S + \pi_P T.\end{aligned}$$


---

### Exercise 7.8

Let  $P = P(S, T) \subseteq \mathbb{R}^n$  be a polyhedron where  $S = \{\mathbf{x}^1, \dots, \mathbf{x}^q\}$ ,  $T = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$ , and let  $\mathbf{X} = (\mathbf{x}^1 \cdots \mathbf{x}^q)$ ,  $\mathbf{Y} = (\mathbf{y}^1 \cdots \mathbf{y}^r)$  be the corresponding matrices.

(i) Show that  $\dim P(S, T) = r(\mathbf{X}, \mathbf{Y})$ .

(ii) Show that  $P(S, T) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{v}^i \mathbf{x} = v_i \text{ for } 1 \leq i \leq s, \mathbf{v}^i \mathbf{x} \leq v_i \text{ for } s+1 \leq i \leq t\}$  is an ideal description of  $P(S, T)$  where  $(\mathbf{v}^i, v_i)$  for  $1 \leq i \leq s$  form a basis of the linearity space  $L_{PC}$  of the cone

$$PC = \{(\mathbf{v}, v_0) \in \mathbb{R}^{n+1} : \mathbf{v}\mathbf{X} - v_0\mathbf{e} \leq \mathbf{0}, \mathbf{v}\mathbf{Y} \leq \mathbf{0}\}$$

and the vectors  $(\mathbf{v}^i, v_i)$  for  $s+1 \leq i \leq t$  are the extreme rays of the cone  $PC^0 = PC \cap L_{PC}^\perp$  satisfying  $\mathbf{v}^i \neq \mathbf{0}$ .

---

**(i)** We assume WROG that  $\mathbf{x}^1, \dots, \mathbf{x}^a$  and  $\mathbf{y}^1, \dots, \mathbf{y}^b$  form a submatrix of full rank of  $(\mathbf{X}, \mathbf{Y})$ . Since  $\mathbf{x}^i \in P(S, T)$  for  $1 \leq i \leq a$  and  $\mathbf{x}^1 + \mathbf{y}^j \in P(S, T)$  for  $1 \leq j \leq b$  (if  $a = 0$ , then  $\mathbf{y}^j \in P(S, T)$  for  $1 \leq j \leq b$ ), it follows that  $\dim P(S, T) \geq r(\mathbf{X}, \mathbf{Y})$ . On the other hand, let  $\mathbf{z}^i \in P(S, T)$  for  $1 \leq i \leq c$  be affinely independent points such that  $c$  is as large as possible, i.e.  $\dim P(S, T) = c - 1$ , see Definition DI (p. 114). Since  $\mathbf{z}^i = \sum_{j=1}^q \mu_j^i \mathbf{x}^j + \sum_{j=1}^r \lambda_j^i \mathbf{y}^j$  for some  $\mu_j^i \geq 0$ ,  $\sum_{j=1}^q \mu_j^i = 1$  and  $\lambda_j^i \geq 0$  it follows that  $\mathbf{z}^i$  is a linear combination of  $\mathbf{x}^1, \dots, \mathbf{x}^a$  and  $\mathbf{y}^1, \dots, \mathbf{y}^b$  for  $1 \leq i \leq c$ . Consequently,  $\mathbf{z}^i - \mathbf{z}^1$  for  $2 \leq i \leq c$  are linear combinations of  $\mathbf{x}^1, \dots, \mathbf{x}^a$  and  $\mathbf{y}^1, \dots, \mathbf{y}^b$ . Denote by  $\mathbf{Z}$  the matrix of  $\mathbf{z}^i - \mathbf{z}^1$  for  $2 \leq i \leq c$ , by  $\mathbf{X}_a = (\mathbf{x}^1 \cdots \mathbf{x}^a)$  and  $\mathbf{Y}_b = (\mathbf{y}^1 \cdots \mathbf{y}^b)$ . It follows that there exists an  $(a+b) \times (c-1)$  matrix  $\mathbf{\Lambda}$  of reals such that  $\mathbf{Z} = (\mathbf{X}_a, \mathbf{Y}_b)\mathbf{\Lambda}$ . Consequently, from linear algebra,  $c - 1 = r(\mathbf{Z}) \leq \min\{r(\mathbf{X}_a, \mathbf{Y}_b), r(\mathbf{\Lambda})\} \leq r(\mathbf{X}, \mathbf{Y})$ , i.e.,  $\dim P(S, T) = r(\mathbf{X}, \mathbf{Y})$ .

**(ii)** From the derivation of the linear description of  $P(S, T)$  we know that the description is complete. It remains to show that it is minimal. By assumption  $(\mathbf{v}^i, v_i)$  for  $1 \leq i \leq s$  form a basis of  $L_{PC}$  and thus the system of equations defining  $P(S, T)$  is of full rank. The rest follows because  $(\mathbf{v}^i, v_i)$  for  $s+1 \leq i \leq t$  are extreme rays of  $PC^0$ , i.e. they belong to a minimal generator of  $PC^0$ . If  $(\mathbf{0}, v_0)$  is among the extreme rays of  $PC^0$  then  $v_0 > 0$  which gives the only redundant inequality  $\mathbf{0}\mathbf{x} \leq v_0$  for  $P(S, T)$  which is why we exclude it. Thus the linear description of  $P(S, T)$  is ideal.

## Exercise 7.9

- (i) Show that the test of line (7.16) of the double description algorithm is equivalent to verifying that  $r(\mathbf{H}^{M_i \cap M_j}) = n - nb - 2$  or not, where  $nb$  is the dimension of the linearity space of the processed rows as calculated by the algorithm.
- (ii) Show that the set  $N^*$  in line (7.16) can be replaced by the set  $\{(i, j) : i \in N_+, j \in N_- \text{ such that } |M_i \cap M_j| \geq n - nb - 2 \text{ and } M_i \cap M_j \not\subseteq M_\ell \text{ for all } \ell \in NX - \{i, j\}\}$ .
- (iii) Modify the algorithm so as to find a minimal generator for cones of the form  $C = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{H}_1 \mathbf{y} = \mathbf{0}, \mathbf{H}_2 \mathbf{y} \leq \mathbf{0}\}$ .

**(i)** We denote like in Chapter 7.4.1 by  $\widehat{\mathbf{H}}$  the  $k \times n$  matrix with rows  $\mathbf{h}^0 = \mathbf{0}, \mathbf{h}^1, \dots, \mathbf{h}^{k-1}$  processed by the algorithm DDA and by  $\mathbf{h} = \mathbf{h}^k$  the row to be processed where  $k \geq 1$ . Likewise, we let with the usual conventions

$$N_+ = \{i \in NX : \mathbf{h}\mathbf{y}^i > 0\}, \quad N_- = \{j \in NX : \mathbf{h}\mathbf{y}^j < 0\}, \quad M_\ell = \{i \in \{1, \dots, k-1\} : \mathbf{h}^i \mathbf{y}^\ell = 0\}$$

for all  $\ell \in NX$ . Let  $i \in N_+$  and  $j \in N_-$ . The assertion is equivalent to proving that

$$r(\mathbf{H}^{M_i \cap M_j}) = n - nb - 2 \quad \text{if and only if} \quad M_i \cap M_j \not\subseteq M_\ell \text{ for all } \ell \in NX - \{i, j\},$$

where  $nb = \dim L$  and  $L$  is the linearity space of the cone  $C_{k-1}$ . Since  $nb = n - r(\widehat{\mathbf{H}})$  the assertion is equivalent to claim (3) of the proof of point 7.4(b), which is proven in the text.

**(ii)** By part (i) of this exercise the test in line (7.16) of the algorithm DDA is equivalent to checking that  $r(\mathbf{H}^{M_i \cap M_j}) = n - nb - 2$ . Consequently, the matrix  $\mathbf{H}^{M_i \cap M_j}$  must have at least  $n - nb - 2$  rows and the correctness of the assertion follows. While seemingly trivial, this simple sharpening of the test (7.16) speeds up the calculations considerably.

**(iii)** Denote by  $M^+ \subseteq M = \{1, \dots, m\}$  the subset of rows of the matrix  $\mathbf{H}$  that correspond to equations in the definition of the cone  $C$ . We communicate the set  $M^+$  to the algorithm DDA and modify Steps 3 and 4 as follows:

- Step 3':** Replace  $BL_{k-1}$  by  $BL_k = \{(\mathbf{h}\mathbf{b})\mathbf{b}^i - (\mathbf{h}\mathbf{b}^i)\mathbf{b} : i \in \{1, \dots, nb\} - j\}$  and set  $nb := nb - 1$ .  
 If  $k \in M^+$  replace  $EX_{k-1}$  by  $EX_k = \{(\mathbf{h}\mathbf{y}^i)\mathbf{b} - (\mathbf{h}\mathbf{b})\mathbf{y}^i : i \in NX\}$  and **go to Step 1**.  
 Otherwise, replace  $EX_{k-1}$  by  $EX_k = \{\mathbf{b}\} \cup \{(\mathbf{h}\mathbf{y}^i)\mathbf{b} - (\mathbf{h}\mathbf{b})\mathbf{y}^i : i \in NX\}$ ,  
 set  $nx := nx + 1$  and **go to Step 1**.
- Step 4':** Set  $N_0 := \{i \in NX : \mathbf{h}\mathbf{y}^i = 0\}$ ,  $N_+ := \{i \in NX : \mathbf{h}\mathbf{y}^i > 0\}$ ,  $N_- := \{i \in NX : \mathbf{h}\mathbf{y}^i < 0\}$ .  
 For all  $\ell \in NX$  calculate  $M_\ell = \{i \in \{1, \dots, k-1\} : \mathbf{h}^i \mathbf{y}^\ell = 0\}$  and  
 $N^* := \{(i, j) : i \in N_+, j \in N_- \text{ such that } M_i \cap M_j \not\subseteq M_\ell \text{ for all } \ell \in NX - \{i, j\}\}$ .  
 If  $k \in M^+$  replace  $EX_{k-1}$  by  $EX_k := \{\mathbf{y}^i : i \in N_0\} \cup \{|\mathbf{h}\mathbf{y}^j|\mathbf{y}^i + |\mathbf{h}\mathbf{y}^i|\mathbf{y}^j : (i, j) \in N^*\}$   
 and set  $nx := |N_0| + |N^*|$ .  
 If  $k \notin M^+$  replace  $EX_{k-1}$  by  $EX_k := \{\mathbf{y}^i : i \in N_0 \cup N_-\} \cup \{|\mathbf{h}\mathbf{y}^j|\mathbf{y}^i + |\mathbf{h}\mathbf{y}^i|\mathbf{y}^j : (i, j) \in N^*\}$   
 and set  $nx := |N_0| + |N_-| + |N^*|$ .  
 Set  $BL_k := BL_{k-1}$  and **go to Step 1**.

Otherwise the algorithm DDA described on page 153 remains unchanged. The changes are justified as follows. If the current row of  $H$  to be processed is an inequality, then nothing changes. Otherwise, let  $h^k y = 0$  be the current equation that is to be processed by the algorithm. We replace  $h^k y = 0$  by the pair of inequalities  $h^k y \leq 0$  and  $-h^k y \leq 0$  that are processed in this order. Steps 3' and 4' are the resulting simplifications that impose themselves when the original algorithm is executed on the pair of inequalities. In computational practice we proceed slightly differently by executing a modified algorithm that finds first a basis of the lineality space of the cone  $C$ , before determining the conical part of the minimal generator of  $C$  – see Chapters 7.4.3 and 7.4.4 and make the appropriate modifications.

---

### Exercise 7.10

- (i) Write a computer program in a language of your choice for the double description algorithm with integer data using the Euclidean algorithm to reduce the size of the numbers that the algorithm produces and compact data structures similar to the ones discussed in Chapter 5.4 to save storage space.
  - (ii) Redo all of Exercise 7.6 using your computer program.
  - (iii) Determine all extreme points of the unit cube  $C_n = \{x \in \mathbb{R}^n : 0 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$  in  $\mathbb{R}^n$  using the double description algorithm.
  - (iv) Let  $H = (h_j^i)_{j=1,\dots,n}^{i=1,\dots,n}$  be given by  $h_j^i = 0$  for all  $i < j$ ,  $h_i^i = 2$  and  $h_j^i = 1$  for all  $i > j$ . Find all extreme rays of the cone  $Hx \leq 0$  using the double description algorithm with and without the Euclidean algorithm.
- 

- (i)** The following code is an implementation of the Double Description Algorithm as a MATLAB function. For simplicity, we have not used the compact data structures as suggested in the exercise.

```

%% This is the implementation of the Double Description Algorithm (DDA)
%% as found on page. 153. We DO NOT use sparse structures.
%% The function vecgcd is used to calculate the gcd of the components
%% of a~vector.
%%
%% NAME      : dda
%% PURPOSE: Find the extreme rays of the cone H x <= 0
%% INPUT    : The matrix H
%% OUTPUT   : nb: the dimension of the lineality space
%%             B : a~basis of the lineality space
%%             nx: the number of extreme rays
%%             Y : the extreme rays of the cone (columnwise)

```

```

%%%%%
% Example matrices
% 7.6(i)
%   H=[1 0 -1; 1 1 -1; 1 1 -2; 0 1 -1; 1 2 0; 1 1 0; 2 1 0];
% 7.6(ii)
%   H=[-1 0; 0 -1; -1 2; 1 -2];
%   H=[2 1 0; 0 0 -1];
% 7.6(iii)
%   H=[1 0 0 -1; 1 -1 0 0; -1 1 0 0; 1 0 1 0; -1 0 1 0];
% 7.6 (iv)
%   H=[2 0; 1 2]; %n=2
%   H=[2 0 0; 1 2 0; 1 1 2]; %n=3
%   H=[2 0 0 0; 1 2 0 0; 1 1 2 0; 1 1 1 2]; %n=4
%   H=[2 0 0 0 0; 1 2 0 0 0; 1 1 2 0 0; 1 1 1 2 0; 1 1 1 1 2]; %n=5
%%%%%

%% Initialization

function [nb,B,nx,Y]=ddda(H)
[m,n]=size(H);
k=0;
nb=n;
B=eye(n);
nx=0;

while k < m,
    k=k+1;
    h=H(k,:);
    if nb > 0 & any(h*B) ~= 0, %% Execute Step 3 of the algorithm
        aux=find(h*B ~= 0);
        j=aux(1);
        b=B(:,j);
        if h*B(:,j) > 0, b=-b; end
        cnt=1;
        for i=1:nb,
            if i ~= j,
                NB(:,cnt)=h*b*B(:,i)-h*B(:,i)*b;
                g=vecgcd(NB(:,cnt));
                NB(:,cnt)=NB(:,cnt)/g;    % Euclidean reduction
                cnt=cnt+1;
            end
        end
        B=zeros(n,nb);
        B=NB;
        NB=zeros(n,nb);
    end
end

```

```

nb=nb-1;
for i=1:nx,
    Y(:,i)=h*Y(:,i)*b-h*b*Y(:,i);
    g=vecgcd(Y(:,i));
    Y(:,i)=Y(:,i)/g;
end
nx=nx+1;
Y(:,nx)=b;
else          %% Execute Step 4 of the algorithm
n0=find(h*Y == 0);
np=find(h*Y > 0);
nm=find(h*Y < 0);
M=H(1:k-1,:)*Y;
[au,sn0]=size(n0);
[au,snp]=size(np);
[au,snm]=size(nm);

cnt=1;
for i=1:sn0,
    NY(:,cnt)=Y(:,n0(i));
    cnt=cnt+1;
end
for i=1:snm,
    NY(:,cnt)=Y(:,nm(i));
    cnt=cnt+1;
end

for i=1:snp,
    for j=1:snm,
        subset = -1;
        aux=find(abs(M(:,np(i)))+abs(M(:,nm(j))) == 0);
        [sau,au]=size(aux);
        if nx > 2,
            for l=1:nx,
                if l ~= np(i) & l~= nm(j),
                    subset=1;
                    for t=1:sau,
                        if (M(aux(t),l) ~= 0),
                            subset=0;
                            break;
                        end
                    end
                    if subset == 1, break; end
                end
            end
        end
    end
end

```

```

elseif nx == 2,      % if nx=2 then N*=NX in (7.16)
    subset = 0;
else
    subset = 1;
end
if subset == 0,      % (i,j) satisfies condition (7.16)
    yi=Y(:,np(i));
    yj=Y(:,nm(j));
    NY(:,cnt)=abs(h*yj)*yi+abs(h*yi)*yj;
    g=vecgcd(NY(:,cnt));
    NY(:,cnt)=NY(:,cnt)/g;
    cnt=cnt+1;
end
end
if cnt >= 2,
    clear Y;
    Y=NY;
    clear NY;
end
nx=cnt-1;
end
end

```

The implementation of the function `vecgcd(x)` which finds the greatest common divisor of the components of a vector  $x$  is as follows.

```

function [g] = vecgcd(x)

if round(x) ~= x,
    error('Requires integer vector components')
end

[m,n]=size(x);
if m > 1, x=x'; end
y=find(x ~= 0 );
g=x(y(1));
for i=2:max(size(y)),
    g=gcd(g,x(y(i)));
end
g=abs(g);

```

The following program shows how to use the function and format the output.

```
H=[1 0 -1; 1 1 -1; 1 1 -2; 0 1 -1; 1 2 0; 1 1 0; 2 1 0];
```

```
[nb,B,nx,Y]=ddda(H);
```

```

fprintf('The dimension of the lineality space is: %d \n',nb)
if nb > 0,
    fprintf('A basis of the lineality space is: \n')
    for i=1:nb,
        fprintf('%3d ',i)
        fprintf('%4d',B(:,i))
        fprintf('\n')
    end
end
if nx > 0,
    fprintf('The extreme rays of the cone are: \n%d')
    for i=1:nx,
        fprintf('%3d ',i)
        fprintf('%4d',Y(:,i))
        fprintf('\n')
    end
end

```

**(ii)** To find linear descriptions for the polyhedra of Exercise 7.6, we give the following matrices as input to the function dda

```

7.6(i) H=[1 0 -1; 1 1 -1; 1 1 -2; 0 1 -1; 1 2 0; 1 1 0; 2 1 0];
7.6(ii) H=[-1 0; 0 -1; -1 2; 1 -2];
7.6(iii) H=[1 0 0 -1; 1 -1 0 0; -1 1 0 0; 1 0 1 0; -1 0 1 0];

```

For part (i) we get

The dimension of the lineality space is: 0

The extreme rays of the cone are:

- 1) 0 0 1
- 2) -1 -1 -1
- 3) -2 1 1
- 4) 1 -2 1

which translates to the following linear description

$$-x_1 - x_2 \leq -1, -2x_2 + x_1 \leq 1, x_1 - 2x_2 \leq 1.$$

For part (ii) we get similarly one extreme ray  $(2, 1)$  that gives the inequality  $2x_1 + x_2 \leq 0$ , and for part (iii) we get three extreme rays  $(0, 0, 0, 1)$ ,  $(-1, -1, -1, -1)$  and  $(1, 1, -1, 1)$ , which give the description  $-x_1 - x_2 - x_3 \leq -1$ ,  $x_1 + x_2 - x_3 \leq 1$ .

To find a minimal generator for part (i) we apply dda once more on the cone  $HP$  that results after homogenization (see (7.5)), i.e., with input matrix  $H=[-1 -1 1; -2 1 -1; 1 -2 -1; 0 0 -1]$ . We get

The dimension of the lineality space is: 0

The extreme rays of the cone are:

- 1) 1 0 1

- 2)    0    1    1  
 3)    2    1    0  
 4)    1    2    0

which is interpreted as follows: a minimal generator is  $((1, 0), (0, 1), (2, 1), (1, 2))$ . Similarly, for part (ii) we use as input the matrix  $H = [2 \ 1]$  and find that  $(1, -2)$  is a basis of the lineality space  $L_P = \text{cone}\{(1, -2), (-1, 2)\}$ , and  $(-1, 0)$  is its extreme ray. Thus a minimal description is  $\{\{0\}, \{(1, -2), (-1, 2), (-1, 0)\}$ . Calculating the pointwise description of  $P^0 = P \cap L_P^\perp$ , using input matrix  $H = [2 \ 1 \ 0; -1 \ 2 \ 0; 1 \ -2 \ 0; 0 \ 0 \ -1]$  we get that  $P^0$  has one extreme point  $(0, 0)$  and one extreme ray  $(-2, -1)$ . Doing the same for part (iii) and input matrix  $H = [-1 \ -1 \ -1 \ 1; 1 \ 1 \ -1 \ -1; 0 \ 0 \ 0 \ -1]$  we get that a basis of the lineality space  $L_P$  is  $(1, -1, 0)$ . Thus  $P^0 = P \cap L_P^\perp = \{-x_1 - x_2 - x_3 \leq -1, x_1 + x_2 - x_3 \leq 1, x_1 - x_2 = 0\}$  and using dda with input matrix  $H = [-1 \ -1 \ -1 \ 1; 1 \ 1 \ -1 \ -1; 1 \ -1 \ 0 \ 0; 0 \ 0 \ 0 \ -1]$  we get that  $P^0$  has one extreme point  $(0.5, 0.5)$  and three extreme rays  $(1, 1, 2)$ ,  $(-1, -1, 2)$  and  $(-1, 1, 0)$ .

**(iii)** After the homogenization, the input matrix is

$$H = [\text{eye}(n) \ -\text{ones}(n, 1); -\text{eye}(n) \ \text{zeros}(n, 1); \ \text{zeros}(1, n) \ -1]$$

and using dda e.g. for  $n = 3$ , we get the following output:

The dimension of the lineality space is: 0

The extreme rays of the cone are:

- 1)    0    1    1    1  
 2)    1    1    1    1  
 3)    0    0    1    1  
 4)    1    0    1    1  
 5)    0    1    0    1  
 6)    1    1    0    1  
 7)    0    0    0    1  
 8)    1    0    0    1

and thus the  $C_3$  is a polytope with extreme points all the zero-one vectors in  $\mathbb{R}^3$ .

To prove the statement in full generality, we order the constraints of  $C_n$  after homogenization as follows:  $h^j x = -x_j \leq 0$  for  $1 \leq j \leq n+1$ ,  $h^{n+1+j} x = x_j - x_{n+1} \leq 0$  for  $1 \leq j \leq n$ . We claim that in iteration  $k$  with  $0 \leq k \leq n+1$  we get

$$BL_k = \{(-1)^k u_j : k+1 \leq j \leq n+1\}, \quad EX_k = \{u_j : 1 \leq j \leq k\}$$

by executing the algorithm DDA on p.153, where  $u_j \in \mathbb{R}^{n+1}$  is the  $j$ -th unit vector. This is correct for  $k = 0$ . Suppose the assertion is true for some  $k \geq 0$ ,  $k < n+1$ . In Step 1 the algorithm then picks  $h = -u_{k+1}^T$  as the next row to be processed. In Step 2 we get  $b = u_{k+1}$  and thus we calculate  $BL_{k+1} = \{(-1)^{k+1} u_j : k+2 \leq j \leq n+1\}$  and  $EX_{k+1} = \{u_j : 1 \leq j \leq k+1\}$ . Consequently, the assertion is correct and thus in iteration  $k = n+1$  we find  $BL_{n+1} = \emptyset$ , i.e., the cone is pointed, and  $EX_{n+1} = \{u_j : 1 \leq j \leq n+1\}$ . There remain  $n$  rows of the constraint matrix to be processed. For notational convenience, we reset the counter  $k$  of the algorithm to zero and claim that

$$EX_{n+1+k} = \{u_\ell : k+1 \leq \ell \leq n\} \cup \{u_{n+1} + \sum_{\ell \in S} u_\ell : S \subseteq \{1, \dots, k\}\}$$

and  $|EX_{n+1+k}| = n - k + 2^k$  for  $0 \leq k \leq n$ . This is correct for  $k = 0$ , i.e., for iteration  $n + 1$  of the algorithm. Suppose the assertion is correct for some  $k \geq 0$ ,  $k < n$ . In Step 1 of the algorithm we then pick  $\mathbf{h} = \mathbf{u}_{k+1}^T - \mathbf{u}_{n+1}^T$  as the next row to be processed. Since  $nb = 0$  in Step 2 we go to Step 4. We calculate  $N_0$  to be the set of indices corresponding to the elements  $\mathbf{u}_\ell$  for  $k + 2 \leq \ell \leq n$  of  $EX_{n+1+k}$ .  $N_+$  consists of the single index corresponding to  $\mathbf{u}_{k+1}$  and  $N_-$  consists of all remaining elements of  $EX_{n+1+k}$ . Consequently, forming the set  $N^*$ , we find that every element in  $N_-$  has to be combined with the single element of  $N_+$ , because the rank of the submatrix of the processed rows that are simultaneously annulled by  $\mathbf{u}_{k+1}$  and by  $\mathbf{u}_{n+1} + \sum_{\ell \in S} \mathbf{u}_\ell$  equals  $n - 1$  for every  $S \subseteq \{1, \dots, k\}$  and thus by Exercise 7.9(i) all possible combinations  $(k + 1, j)$  for  $j \in N_-$  pass the test in line (7.16) of the algorithm. It follows that

$$EX_{n+1+k+1} = \{\mathbf{u}_\ell : k + 2 \leq \ell \leq n\} \cup \{\mathbf{u}_{n+1} + \sum_{\ell \in S} \mathbf{u}_\ell : S \subseteq \{1, \dots, k + 1\}\}$$

and  $|EX_{n+2+k}| = n - (k + 1) + 2^{k+1}$  as claimed. By Exercise 7.3, we hence get exactly the  $2^n$  extreme points of the unit cube  $C_n$  at the termination of the algorithm.

**(iv)** We first apply the algorithm DDA without Euclidean reduction to the cone  $C = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Hx} \leq 0\}$  and claim – as done in the text on page 161 – that

$$BL_k = \{(-1)^k 2^{2^k-1} \mathbf{u}_i : k + 1 \leq i \leq n\}, \quad EX_k = \{2^{2^k-2^{i-1}-k+i-1} \mathbf{z}_i^k : 1 \leq i \leq k\}$$

where  $\mathbf{u}_i \in \mathbb{R}^n$  is the  $i$ -th unit vector and  $\mathbf{z}_i^k$  is given by

$$\mathbf{z}_i^k = -2^{k-i} \mathbf{u}_i + \sum_{j=i+1}^k 2^{k-j} \mathbf{u}_j$$

for  $1 \leq i \leq k$  and  $0 \leq k \leq n$ . The assertion is true for  $k = 0$ . So suppose that it is true for  $k \geq 0$ ,  $k < n$ . In Step 1 of the algorithm we pick thus the row  $\mathbf{h}^{k+1}$  of  $\mathbf{H}$  with

$$h_\ell^{k+1} = 1 \text{ for } 1 \leq \ell \leq k, \quad h_{k+1}^{k+1} = 2, \quad h_\ell^{k+1} = 0 \text{ for } k + 2 \leq \ell \leq n,$$

as the next row to be processed. In Step 2 of the algorithm we select  $\mathbf{b} = -2^{2^k-1} \mathbf{u}_{k+1}$  and go to Step 3. We calculate with  $\mathbf{h} = \mathbf{h}^{k+1}$

$$(\mathbf{h}\mathbf{b})\mathbf{b}^i - (\mathbf{h}\mathbf{b}^i)\mathbf{b} = -2^{2^k} (-1)^k 2^{2^k-1} \mathbf{u}_i + \mathbf{0} = (-1)^{k+1} 2^{2^{k+1}-1} \mathbf{u}_i$$

for  $k + 2 \leq i \leq n$  and thus the formula for  $BL_k$  follows by induction. Likewise we calculate the elements of  $EX_{k+1}$  by

$$(\mathbf{h}\mathbf{y}^i)\mathbf{b} - (\mathbf{h}\mathbf{b})\mathbf{y}^i = 2^{2^k-2^{i-1}-k+i-1} 2^{2^k-1} (-(\mathbf{h}\mathbf{z}_i^k)\mathbf{u}_{k+1} + 2\mathbf{z}_i^k) = 2^{2^{k+1}-2^{i-1}-(k+1)+i-1} \mathbf{z}_i^{k+1}$$

for  $1 \leq i \leq k + 1$ , where we have used that  $\sum_{j=i+1}^k 2^{k-j} = 2^{k-i} - 1$  and thus  $\mathbf{h}^{k+1}\mathbf{z}_i^k = -1$ . Since  $\mathbf{z}_{k+1}^{k+1} = \mathbf{b}$  the formula for  $EX_k$  follows by induction as well. Thus after  $n$  iterations we find that the cone  $C$  is pointed and that it has precisely the  $n$  extreme rays given by  $EX_n$ .

We show next by induction that the iterative application of the Euclidean reduction in the double description algorithm produces the sets

$$BL_k = \{(-1)^k \mathbf{u}_i : k + 1 \leq i \leq n\}, \quad EX_k = \{\mathbf{z}_i^k : 1 \leq i \leq k\}$$

for all  $0 \leq k \leq n$  as asserted in the text on page 161 for the example cone of this exercise. The assertion is true for  $k = 0$ . So suppose that it is true for  $k \geq 0$ ,  $k < n$ . In Step 1 of the algorithm we pick as above the  $\mathbf{h}^{k+1}$  of  $\mathbf{H}$  as the next row to be processed. In Step 2 of the algorithm we select  $\mathbf{b} = -\mathbf{u}_{k+1}$  and go to Step 3. We calculate with  $\mathbf{h} = \mathbf{h}^{k+1}$

$$(\mathbf{h}\mathbf{b})\mathbf{b}^i - (\mathbf{h}\mathbf{b}^i)\mathbf{b} = (-2)(-1)^k \mathbf{u}_i + \mathbf{0} = 2(-1)^{k+1} \mathbf{u}_i$$

for  $k+2 \leq i \leq n$ . Clearing the common divisor of 2 in each one of the elements of  $BL_{k+1}$ , the changed formula for  $BL_k$  follows by induction. Likewise we calculate the changed  $EX_{k+1}$  by

$$(\mathbf{h}\mathbf{y}^i)\mathbf{b} - (\mathbf{h}\mathbf{b})\mathbf{y}^i = (-1)(-\mathbf{u}_{k+1}) - (-2)(-2^{k-i} \mathbf{u}_i + \sum_{j=i+1}^k 2^{k-j} \mathbf{u}_j) = -2^{k+1-i} \mathbf{u}_i + \sum_{j=i+1}^{k+1} 2^{k+1-j} \mathbf{u}_j$$

for  $1 \leq i \leq k+1$  and  $\mathbf{z}_{k+1}^{k+1} = \mathbf{b}$ . So the changed formula for  $EX_k$  follows as well by induction. Thus after  $n$  iterations we find that the cone  $C$  is pointed and that it has precisely  $n$  extreme rays as stated in the changed set  $EX_n$  which has numbers of a considerably smaller size than those that are produced by “blindly” applying the double description algorithm without Euclidean reduction. Reread the text on pages 160-162 to fully appreciate the reduction in the “digital size” of the numbers that can be achieved when you use Euclidean reduction in the algorithm DDA.

---

### Exercise 7.11

Let  $E = J = \{1, \dots, k\}$  and  $\ell > k$ . Denote by  $\mathbf{H}_J^{E-r+\ell}$  the  $k \times k$  matrix that is obtained from  $\mathbf{H}_J^E$  by deleting row  $r$  and adding an arbitrary row vector  $\mathbf{h}^\ell \in \mathbb{R}^k$  as the last row. Show that  $\det \mathbf{H}_J^{E-s+\ell} \det \mathbf{H}_{J-j}^{E-r} - \det \mathbf{H}_J^{E-r+\ell} \det \mathbf{H}_{J-j}^{E-s} = \det \mathbf{H}_J^E \det \mathbf{H}_{J-j}^{E-\{r,s\}+\ell}$  for all  $1 \leq r < s \leq k$  and  $1 \leq j \leq k$ . (Hints: Use  $\det \mathbf{B}^T = \det \mathbf{B}$  and point 7.4(f).)

---

First we observe that since  $\det \mathbf{B}^T = \det \mathbf{B}$  and  $E = J$  we have  $\det \mathbf{H}_{J-s}^{E-j} = \det \mathbf{H}_{J-j}^{E-s}$ . Thus we calculate

$$\begin{aligned} \det \mathbf{H}_J^E \det \mathbf{H}_{J-j}^{E-\{r,s\},\ell} &= \det \mathbf{H}_J^E \det \mathbf{H}_{J-\{r,s\}+\ell}^{E-j} = \det \mathbf{H}_{J-s+\ell}^E \det \mathbf{H}_{J-r}^{E-j} - \det \mathbf{H}_{J-r+\ell}^E \det \mathbf{H}_{J-s}^{E-j} \\ &= \det \mathbf{H}_J^{E-s+\ell} \det \mathbf{H}_{J-j}^{E-r} - \det \mathbf{H}_J^{E-r+\ell} \det \mathbf{H}_{J-j}^{E-s}, \end{aligned}$$

where we have used point 7.4(f) in the second equality.

---

### Exercise 7.12

Write a computer program in a language of your choice for the basis algorithm as stated and one for the modified basis algorithm with the Euclidean algorithm using compact data structures like those of Chapter 5.4.

---

The following program is an implementation of the basis algorithm as stated in the book, as a MATLAB function. As before, we do not use compact data structures.

```
%%%%%%%
%% This is the implementation of the Basis Algorithm
%% as found on pages 164-5. We DO NOT use sparse structures.
%% The function vecgcd is used to calculate the gcd of the components
%% of a~vector.
%%
%% NAME      : basis
%% PURPOSE: Find the rank of a~matrix H and a~basis of the lineality
%%             space of the cone H x <= 0
%% INPUT    : The matrix H
%% OUTPUT   : nb   : the dimension of the lineality space
%%             B    : a~basis of the lineality space
%%             r    : the rank of the matrix H
%%             E    : row set such that |E| is the rank of H
%%             J    : column set such that |J| is the rank of H
%%             Det  : the determinant of the matrix H_J^E
%%             nx   : the number of extreme rays of the cone H^E x <=0
%%             Y    : the extreme rays (columnwise) of the cone H^E x <=0
%%             Hinv: the inverse of the matrix HE
%%%%%%%
function [nb,B,nx,Y,r,E,J,Det,Hinv]=basis(H)
[m,n]=size(H);
k=0;
nb=n;
B=eye(n);
nx=0;
A=ones(1,m);
R=ones(1,n);
jcnt=0;
ecnt=0;
d=1;

while k < m,
  k=k+1;
  aux=find(A ~= 0);
  r=aux(1);
  clear aux
  A(r)=0;
  h=H(r,:);
  if any(h*B) ~= 0,
    aux=find(h*B ~= 0);
    j=aux(1);
    b=B(:,j);
    sigma=1;
```

```

if h*B(:,j) > 0, sigma=-1; end
dold = d;
d=h*b;
ecnt=ecnt+1;
E(ecnt)=r;
jcnt=jcnt+1;
for l=1:n,
    if R(l) == 1 & B(l,j) ~= 0, break; end
end
J(jcnt)=l;
R(l)=0;
cnt=1;
for i=1:nb,
    if i ~= j,
        NB(:,cnt)=(d*B(:,i)-h*B(:,i)*b)/dold;
        cnt=cnt+1;
    end
end
B=zeros(n,nb);
B=N;
NB=zeros(n,nb);
nb=nb-1;
for i=1:nx,
    Y(:,i)=sigma*(h*Y(:,i)*b-d*Y(:,i))/abs(dold);
end
nx=nx+1;
Y(:,nx)=sigma*b;
end
if nb == 0, break; end
end
r=ecnt;
Det=d;
Hinv=-Y/abs(d);

```

The following program demonstrates the use of the function:

```

H=[ 1 4 5; 1 2 0; 1 2 1];
[m,n]=size(H);
[nb,B,nx,Y,r,E,J,Det,Hinv]=basis(H);

fprintf('The rank of the matrix is: %d \n',r)
fprintf('A rank %2d submatrix HE of determinant %4d is:\n',r,Det)
HE=H(E,J);
for i=1:r,
    for j=1:r,
        fprintf('%4d',HE(i,j))
    end

```

```

    fprintf('\n')
end
fprintf('and its inverse is:\n')
for i=1:r,
    for j=1:r,
        fprintf('%7.3f',Hinv(i,j))
    end
    fprintf('\n')
end
fprintf('The dimension of the lineality space is: %d \n',nb)
if nb > 0,
    fprintf('A basis of the lineality space is: \n')
    for i=1:nb,
        fprintf('%3d ',i)
        fprintf('%4d',B(:,i))
        fprintf('\n')
    end
end
if nx > 0,
    fprintf('The extreme rays of the cone HE <= 0 are: \n')
    for i=1:nx,
        fprintf('%3d ',i)
        fprintf('%4d',Y(:,i))
        fprintf('\n')
    end
end

```

The output for the matrix  $H = [1\ 4\ 5; 1\ 2\ 0; 1\ 2\ 1]$  is the following one:

The rank of the matrix is: 3

A~rank 3 submatrix HE of determinant -2 is:

```

1   4   5
1   2   0
1   2   1

```

and its inverse is:

```

-1.000 -3.000  5.000
 0.500  2.000 -2.500
 0.000 -1.000  1.000

```

The dimension of the lineality space of the cone  $H \leq 0$  is: 0

The extreme rays of the cone  $HE \leq 0$  are:

- 1) 2 -1 0
- 2) 6 -4 2
- 3) -10 5 -2

The implementation of the algorithm using Euclidean reduction is as follows:

```
%%%%%%%%%%%%%%%
%% This is the implementation of the Basis Algorithm
```

```

%% as found on pages 164-5, modified to use Euclidean reduction
%% as suggested on page 166. We DO NOT use sparse structures.
%% The function vecgcd is used to calculate the gcd of the components
%% of a~vector.
%%
%% NAME    : eubasis
%% PURPOSE: Find the rank of a~matrix H and a~basis of the lineality
%%           space of the cone H x <= 0
%% INPUT   : The matrix H
%% OUTPUT  : nb   : the dimension of the lineality space
%%           B    : a~basis of the lineality space
%%           r    : the rank of the matrix H
%%           E    : row set such that |E| is the rank of H
%%           J    : column set such that |J| is the rank of H
%%           Det  : the determinant of the matrix H_J^E
%%           nx   : the number of extreme rays of the cone H^E x <=0
%%           Y    : the extreme rays (columnwise) of the cone H^E x <=0
%%           Hinv: the inverse of the matrix HE
%%%%%
function [nb,B,nx,Y,r,E,J,Det,Hinv]=eubasis(H)
[m,n]=size(H);
k=0;
nb=n;
B=eye(n);
drem=ones(n);
nx=0;
A=ones(1,m);
R=ones(1,n);
jcnt=0;
ecnt=0;
d=1;
while k < m,
  k=k+1;
  aux=find(A ~= 0);
  r=aux(1);
  clear aux
  A(r)=0;
  h=H(r,:);
  if any(h*B) ~= 0,
    aux=find(h*B ~= 0);
    j=aux(1);
    b=B(:,j);
    sigma=1;
    if h*B(:,j) > 0, sigma=-1; end
    dold = d;
    d=h*b;
    for i=1:n,
      if B(i,j) ~= 0,
        B(i,:)=B(i,:)-sigma*(dold*B(i,:));
      end
    end
    if sigma<0, B=r*B;
  end
end

```

```

Det=d*drem(j);
ecnt=ecnt+1;
E(ecnt)=r;
jcnt=jcnt+1;
for l=1:n,
    if R(l) == 1 & B(l,j) ~= 0, break; end
end
J(jcnt)=l; R(l)=0;
cnt=1;
for i=1:nb,
    if i ~= j,
        NB(:,cnt)=d*B(:,i)-h*B(:,i)*b;
        div=vecgcd(NB(:,cnt));
        NB(:,cnt)=NB(:,cnt)/div;
        drem(cnt)=drem(i)*div;
        cnt=cnt+1;
    end
end
B=zeros(n,nb);
B=NB;
NB=zeros(n,nb);
nb=nb-1;
for i=1:nx,
    Y(:,i)=sigma*(h*Y(:,i)*b-d*Y(:,i));
    Y(:,i)=Y(:,i)/vecgcd(Y(:,i));
end
nx=nx+1;
Y(:,nx)=sigma*b;
end
if nb == 0, break; end
end
r=ecnt;
clear aux
aux=diag(Y(J,:)*H(E,J));
Hinv=Y(J,:)*diag(1 ./ aux);

```

The following MATLAB program, eubacall.m shows the use of the above function (as usual, we assume that the function is stored in a file with the same name as the function,in our case eubasis.m). As examples, we use the matrix  $H$  of Exercise 7.10(iv) with  $n = 5$  and the matrix  $K_{3n}$  with  $n = 3$  (see page 169 in the text).

```

H=[2 0 0 0 0; 1 2 0 0 0; 1 1 2 0 0; 1 1 1 2 0; 1 1 1 1 2];
%T=[1 1 0; 1 0 1; 0 1 1];
%Z=zeros(3,3);
%H=[T Z Z; Z T Z; Z Z T];

```

```
[m,n]=size(H);
[nb,B,nx,Y,r,E,J,Det,Hinv]=eubasis(H);

fprintf('The rank of the matrix is: %g \n',r)
fprintf('A rank %2g submatrix HE of determinant %4g is:\n',r,Det)
HE=H(E,J);
for i=1:r,
    for j=1:r,
        fprintf('%4g',HE(i,j))
    end
    fprintf('\n')
end
fprintf('and its inverse is:\n')
for i=1:r,
    for j=1:r,
        fprintf('%7.4f ',Hinv(i,j))
    end
    fprintf('\n')
end
fprintf('The dimension of the lineality space of the cone Hx<=0 is: %g\n',nb)
if nb > 0,
    fprintf('A basis of the lineality space of the cone H x <= 0 is: \n')
    for i=1:nb,
        fprintf('%3g ',i)
        for k=1:n,
            fprintf('%4g',B(k,i))
        end
        fprintf('\n')
    end
end
if nx > 0,
    fprintf('The extreme rays of the cone HE x<= 0 are: \n')
    for i=1:nx,
        fprintf('%3g ',i)
        for k=1:n,
            fprintf('%4g',Y(k,i))
        end
        fprintf('\n')
    end
end
```

We run the basis algorithm with and without Euclidean reduction on the two matrices mentioned above and get the following results:

```
>>clear
>>eubacall
```

The rank of the matrix is: 9

A~rank 9 submatrix HE of determinant -8 is:

1	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	1

and its inverse is:

0.5000	0.5000	-0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.5000	-0.5000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.5000	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.5000	0.5000	-0.5000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.5000	-0.5000	0.5000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-0.5000	0.5000	0.5000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	0.5000	-0.5000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	-0.5000	0.5000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.5000	0.5000

The dimension of the lineality space of the cone  $H x \leq 0$  is: 0

The extreme rays of the cone  $HE x \leq 0$  are:

1)	-1	-1	1	0	0	0	0	0
2)	-1	1	-1	0	0	0	0	0
3)	1	-1	-1	0	0	0	0	0
4)	0	0	0	-1	-1	1	0	0
5)	0	0	0	-1	1	-1	0	0
6)	0	0	0	1	-1	-1	0	0
7)	0	0	0	0	0	0	-1	-1
8)	0	0	0	0	0	0	-1	1
9)	0	0	0	0	0	0	1	-1

>>clear

>>bascall

The rank of the matrix is: 9

A~rank 9 submatrix HE of determinant -8 is:

1	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	1

and its inverse is:

```

0.500  0.500 -0.500  0.000  0.000  0.000  0.000  0.000  0.000
0.500 -0.500  0.500  0.000  0.000  0.000  0.000  0.000  0.000
-0.500  0.500  0.500  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.500  0.500 -0.500  0.000  0.000  0.000
 0.000  0.000  0.000  0.500 -0.500  0.500  0.000  0.000  0.000
 0.000  0.000  0.000 -0.500  0.500  0.500  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.500  0.500 -0.500
 0.000  0.000  0.000  0.000  0.000  0.000  0.500 -0.500  0.500
 0.000  0.000  0.000  0.000  0.000 -0.500  0.500  0.500  0.500

```

The dimension of the lineality space of the cone  $H x \leq 0$  is: 0

The extreme rays of the cone  $HE \leq 0$  are:

- 1) -4 -4 4 0 0 0 0 0 0
- 2) -4 4 -4 0 0 0 0 0 0
- 3) 4 -4 -4 0 0 0 0 0 0
- 4) 0 0 0 -4 -4 4 0 0 0
- 5) 0 0 0 -4 4 -4 0 0 0
- 6) 0 0 0 4 -4 -4 0 0 0
- 7) 0 0 0 0 0 0 -4 -4 4
- 8) 0 0 0 0 0 0 -4 4 -4
- 9) 0 0 0 0 0 0 4 -4 -4

>>clear

>>eubacall

The rank of the matrix is: 5

A~rank 5 submatrix HE of determinant 32 is:

```

2  0  0  0  0
1  2  0  0  0
1  1  2  0  0
1  1  1  2  0
1  1  1  1  2

```

and its inverse is:

```

0.5000  0.0000  0.0000  0.0000  0.0000
-0.2500  0.5000  0.0000  0.0000  0.0000
-0.1250 -0.2500  0.5000  0.0000  0.0000
-0.0625 -0.1250 -0.2500  0.5000  0.0000
-0.0313 -0.0625 -0.1250 -0.2500  0.5000

```

The dimension of the lineality space of the cone  $H x \leq 0$  is: 0

The extreme rays of the cone  $HE \leq 0$  are:

- 1) -16 8 4 2 1
- 2) 0 -8 4 2 1
- 3) 0 0 -4 2 1
- 4) 0 0 0 -2 1
- 5) 0 0 0 0 -1

>>clear

>>bascall

The rank of the matrix is: 5

A~rank 5 submatrix HE of determinant 32 is:

2	0	0	0	0
1	2	0	0	0
1	1	2	0	0
1	1	1	2	0
1	1	1	1	2

and its inverse is:

0.500	0.000	0.000	0.000	0.000
-0.250	0.500	0.000	0.000	0.000
-0.125	-0.250	0.500	0.000	0.000
-0.063	-0.125	-0.250	0.500	0.000
-0.031	-0.063	-0.125	-0.250	0.500

The dimension of the lineality space of the cone  $H \ x \leq 0$  is: 0

The extreme rays of the cone  $HE \ x \leq 0$  are:

- 1) -16 8 4 2 1
- 2) 0 -16 8 4 2
- 3) 0 0 -16 8 4
- 4) 0 0 0 -16 8
- 5) 0 0 0 0 -16

>>

---

### Exercise 7.13

Let  $\mathbf{c} \in \mathbb{R}^n$  be any row vector. Find optimal solutions to  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in S_n\}$ ,  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in C_n\}$ ,  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in H_n\}$  and  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in O_n\}$  where  $C_n$ ,  $S_n$  are defined in Exercise 7.2,  $H_n$ ,  $O_n$  in Exercise 7.7. State your optimality criterion clearly in each case.

---

For the first optimization problem we have  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in S_n\} = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \geq 0, \sum_{j=1}^n x_j \leq 1\}$ . We introduce a slack variable  $x_{n+1}$  with cost  $c_{n+1} = 0$  to bring the LP in standard form. Since  $\mathbf{x} = 0$ ,  $x_{n+1} = 1$  is a feasible solution to the problem and the feasible region of the LP is bounded, we know that there exists a finite optimal solution. Let  $k$  be the index of the variable that is in the optimal basis. From the optimality of the basis it follows that the reduced costs of all variables are nonpositive, i.e.  $c_j - c_k \leq 0$  for all  $1 \leq j \leq n+1$ . Therefore, the solution  $x_k = 1$ ,  $x_j = 0$  otherwise, where  $k$  is such that  $c_k \geq c_j$  for all  $1 \leq j \leq n+1$  is optimal. See also Exercise 5.3.

For the second problem we have  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in C_n\} = \max\{\mathbf{c}\mathbf{x} : 0 \leq x_j \leq 1 \text{ for } 1 \leq j \leq n\}$ . The problem has a finite optimal solution since  $\mathbf{x} = 0$  is a feasible solution and moreover, the feasible region is bounded. Let  $N_+ = \{j \in N : c_j > 0\}$  where  $N = \{1, \dots, n\}$ . Then the solution  $x_j = 1$  for  $j \in N_+$ ,  $x_j = 0$  for  $j \in N - N_+$  is optimal since

$$\mathbf{c}\mathbf{x} = \sum_{j \in N} c_j x_j \leq \sum_{j \in N_+} c_j x_j \leq \sum_{j \in N_+} c_j$$

where the first inequality follows from the nonnegativity of  $x_j$ ,  $j \in N$  and the second from the inequalities  $x_j \leq 1$  for  $j \in N$ .

For the third problem we know from Exercise 7.7(i) that the extreme points of  $H_n$  are the  $n$  unit vectors  $\mathbf{u}^i$ ,  $1 \leq i \leq n$ , in  $\mathbb{R}^n$  and its extreme rays are  $-\mathbf{u}^i$ ,  $1 \leq i \leq n$ . It follows that the polyhedron is not bounded from below and therefore, if there exists  $j \in N$  such that  $c_j < 0$  then since  $-\mathbf{u}^j$  is an extreme direction of  $H_n$  the problem is unbounded. So we assume that  $c \geq 0$ . In this case an optimal solution exists and the corresponding optimal value is achieved at one of the extreme points of  $H_n$ , i.e. one of the unit vectors in  $\mathbb{R}^n$ . Let  $k$  be such that  $c_k \geq c_j$  for all  $1 \leq j \leq n$ . It follows that  $\mathbf{x} = \mathbf{u}^k$  is the optimal solution to the problem.

For the last optimization problem, we have  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in O_n\} = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in \{0,1\}^n, \sum_{j=1}^n x_j = \text{odd}\}$ . Since the unit vectors in  $\mathbb{R}^n$  are in  $O_n$  and  $O_n$  is bounded, the optimization problem has a finite optimal solution. WROG we assume that the components of the cost vector are decreasing, i.e.  $c_1 \geq c_2 \geq \dots \geq c_n$ . Let  $N_+ = \{j \in N : c_j > 0\}$  and  $k = |N_+|$ . If  $k$  is odd then the optimal solution is given by  $x_j = 1$  for  $1 \leq j \leq k$ ,  $x_j = 0$  for  $k \leq j \leq n$  because  $\mathbf{c}\mathbf{x} = \sum_{j \in N} c_j x_j \leq \sum_{j \in N_+} c_j x_j \leq \sum_{j \in N_+} c_j$  for all  $\mathbf{x} \in O_n$ . If  $k$  is even then if  $c_{k+1} = 0$  or  $c_k + c_{k+1} > 0$ , then the optimal solution is given by  $x_j = 1$  for  $1 \leq j \leq k+1$ ,  $x_j = 0$  for  $k+2 \leq j \leq n$ . Otherwise, the optimal solution is given by  $x_j = 1$  for  $1 \leq j \leq k-1$ ,  $x_j = 0$  for  $k \leq j \leq n$ . This follows since, as before,  $\mathbf{c}\mathbf{x} \leq \sum_{j \in N_+} c_j$  for all  $\mathbf{x} \in O_n$  and since  $|N_+|$  is odd we must either drop or add one variable at value one.

---

### \*Exercise 7.14

Let  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{c} \neq \mathbf{0}$  have rational components and  $P \subseteq \mathbb{R}^n$  be a polytope of facet complexity  $\phi$ .

- (i) Show that there exists an integer  $\lambda > 0$  such that  $\tilde{\mathbf{d}} = \lambda \mathbf{c}$  has integer components  $\tilde{d}_j$  with  $\langle \tilde{d}_j \rangle \leq \langle \mathbf{c} \rangle - 2n$  and thus  $\langle \tilde{\mathbf{d}} \rangle \leq n \langle \mathbf{c} \rangle - 2n^2$ .
  - (ii) Let  $\mathbf{x}, \mathbf{y} \in P$  be two extreme points of  $P$  with  $\mathbf{c}\mathbf{x} > \mathbf{c}\mathbf{y}$ . Show that  $\mathbf{c}\mathbf{x} > \mathbf{c}\mathbf{y} + 2^{-8n^2\phi - \langle \mathbf{c} \rangle}$ .
  - (iii) Let  $\Delta \geq 1 + 2^{4n\phi + 8n^2\phi + \langle \mathbf{c} \rangle + 1}$ . Define  $\tilde{d}_j = \Delta^n c_j + \Delta^{n-j}$  for  $1 \leq j \leq n$  and  $\tilde{\mathbf{d}} = (\tilde{d}_1, \dots, \tilde{d}_n)$ . Show that  $\max\{\tilde{\mathbf{d}}\mathbf{x} : \mathbf{x} \in P\}$  has a unique maximizer  $\mathbf{x}^{max}$ , say, and that  $\mathbf{c}\mathbf{x}^{max} = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$ .
  - (iv) Let  $\tilde{\mathbf{d}} \in \mathbb{R}^n$  and  $\mathbf{x}^{max} \in P$  be as defined in (iii) and set  $\mathbf{d} = \tilde{\mathbf{d}} / \|\tilde{\mathbf{d}}\|_\infty$  where  $\|\cdot\|_\infty$  is the  $\ell_\infty$ -norm. Show that  $\langle \mathbf{d} \rangle \leq 3.5n(n-1)\lceil \log_2 \Delta \rceil + 2(n-1)\langle \mathbf{c} \rangle$  and  $\mathbf{d}\mathbf{x}^{max} \geq \mathbf{d}\mathbf{y} + 2^{-8n^2\phi - \langle \mathbf{d} \rangle}$  for every extreme point  $\mathbf{y} \in P$ ,  $\mathbf{y} \neq \mathbf{x}^{max}$ .
  - (v) Show that if  $\mathbf{c} = \mathbf{0}$  and  $P \neq \emptyset$  then the linear optimization problem  $\max\{\tilde{\mathbf{d}}\mathbf{x} : \mathbf{x} \in P\}$  finds the lexicographically maximal extreme point of  $P$ , where  $\tilde{\mathbf{d}}$  is defined in (ii).
- 

- (i) Since  $\mathbf{c}$  is rational, each component  $c_j$  of  $\mathbf{c}$  can be written as  $p_j/q_j$  where  $p_j$  and  $q_j \geq 1$  are relatively prime integer numbers. Let  $\lambda = \prod_{j=1}^n q_j$  and  $\hat{d}_j = \lambda c_j = p_j \prod_{k \neq j} q_k$ . Then  $\hat{d}_j$  is integer for

$1 \leq j \leq n$  and we estimate:

$$\begin{aligned} 1 + \log_2(1 + |p_j| \prod_{k \neq j} q_k) &\leq 1 + \log_2((1 + |p_j|) \prod_{k \neq j} q_k) = 1 + \sum_{k \neq j} \log_2 q_k + \log_2(1 + |p_j|) \\ &\leq \sum_{k \neq j} (1 + \log_2(1 + q_k)) - (n - 1) + 1 + \log_2(1 + |p_j|) \\ &\leq \sum_{k=1}^n (1 + \lceil \log_2(1 + q_k) \rceil + 1 + \lceil \log_2(1 + |p_k|) \rceil) - 2n = \langle c \rangle - 2n. \end{aligned}$$

Consequently,  $\langle \hat{d}_j \rangle \leq \langle c \rangle - 2n$  and  $\langle \hat{d} \rangle \leq n\langle c \rangle - 2n^2$ .

(iii) Let  $c_j = p_j/q_j$  with  $p_j, q_j$  relatively prime integers and  $q_j \geq 1$  for  $1 \leq j \leq n$ . Since  $x, y$  are extreme points of a rational polytope  $P$  of facet complexity  $\phi$  we can write componentwise  $x_j = r_j/s_j$  and  $y_j = u_j/v_j$  where  $r_j, s_j$  and  $u_j, v_j$ , respectively, are relatively prime integers satisfying  $1 \leq s_j, v_j < 2^{4n\phi}$ ; see point 7.5(b). Since by assumption  $cx > cy$  it follows that the integer number  $(\prod_{j=1}^n q_j s_j v_j) c(x - y) \geq 1$ . We calculate

$$(\prod_{j=1}^n q_j)(\prod_{j=1}^n s_j)(\prod_{j=1}^n v_j) < (\prod_{j=1}^n q_j) 2^{8n^2\phi} < 2^{\langle c \rangle + 8n^2\phi}$$

and thus  $cx > cy + 2^{-8n^2\phi - \langle c \rangle}$ .

(iii) Let  $x^0 \in P$  be any extreme point of  $P$  such that  $cx^0 = \max\{cx : x \in P\}$ . Define  $S^+ = \{x : cx = cx^0, x \text{ is an extreme point of } P\}$  and  $S$  to be the set of all extreme points of  $P$ . Let  $x \in S - S^+$  be arbitrary. Then we estimate

$$\begin{aligned} \tilde{d}(x^0 - x) &= \Delta^n c(x^0 - x) + \sum_{j=1}^n \Delta^{n-j} (x_j^0 - x_j) > \Delta^n 2^{-8n^2\phi - \langle c \rangle} - \left| \sum_{j=1}^n \Delta^{n-j} (x_j^0 - x_j) \right| \\ &> \Delta^n 2^{-8n^2\phi - \langle c \rangle} - 2^{4n\phi+1} (\Delta^n - 1) / (\Delta - 1) > \Delta^n 2^{-8n^2\phi - \langle c \rangle} - (\Delta^n - 1) 2^{-8n^2\phi - \langle c \rangle} = 2^{-8n^2\phi - \langle c \rangle}, \end{aligned}$$

where we have used part (ii) of this exercise and the definition of  $\Delta$ . Consequently,  $\tilde{d}x^0 > dx$  for all  $x \in S - S^+$ . Let  $x^{\max} \in S^+$  be such that  $\tilde{d}x^{\max} \geq \tilde{d}x$  for all  $x \in S^+$ . For  $x \in S^+$  with  $x \neq x^{\max}$  let  $k$  be the smallest index such that  $x_k \neq x_k^{\max}$ . We claim that  $x_k^{\max} > x_k$ . For suppose not. Then we calculate

$$\begin{aligned} \sum_{j=1}^n \Delta^{n-j} (x_j^{\max} - x_j) &= \sum_{j=k}^n \Delta^{n-j} (x_j^{\max} - x_j) \\ &\leq \Delta^{n-k} (x_k^{\max} - x_k) + 2^{4n\phi+1} (\Delta^{n-k} - 1) / (\Delta - 1) \\ &\leq \Delta^{n-k} (x_k^{\max} - x_k) + 2^{-8n^2\phi - \langle c \rangle} (\Delta^{n-k} - 1) \leq -2^{-8n^2\phi - \langle c \rangle}, \end{aligned}$$

since  $x_k^{\max} - x_k < 0$  implies by point 7.5(b) that  $x_k^{\max} - x_k \leq -2^{-8n\phi} \leq -2^{-8n^2\phi - \langle c \rangle}$ . This contradicts  $\tilde{d}x^{\max} \geq \tilde{d}x$  and consequently,  $x_k^{\max} > x_k$ . Thus  $x_k^{\max} - x_k \geq 2^{-8n\phi}$  and to finish the proof we

calculate

$$\begin{aligned}\tilde{\mathbf{d}}(\mathbf{x}^{\max} - \mathbf{x}) &= \Delta^{n-k}(x_k^{\max} - x_k) + \sum_{j=k+1}^n \Delta^{n-j}(x_j^{\max} - x_j) \\ &\geq \Delta^{n-k}(x_k^{\max} - x_k) - \left| \sum_{j=k+1}^n \Delta^{n-j}(x_j^{\max} - x_j) \right| \\ &\geq \Delta^{n-k} 2^{-8n\phi} - 2^{-8n^2\phi - \langle \mathbf{c} \rangle} (\Delta^{n-k} - 1) = 2^{-8n^2\phi - \langle \mathbf{c} \rangle}\end{aligned}$$

for all  $\mathbf{x} \in S^{\neq} - \mathbf{x}^{\max}$  which proves the assertion.

**(iv)** Let  $c_j = p_j/q_j$  with  $p_j$  and  $q_j$  relatively prime integers and  $q_j \geq 1$ , and let  $\ell$  be such that  $|\Delta^n c_\ell + \Delta^{n-\ell}| \geq |\Delta^n c_j + \Delta^{n-j}|$  for  $1 \leq j \leq n$ . We estimate

$$\begin{aligned}\langle d_j \rangle &= \langle \frac{\tilde{d}_j}{\tilde{d}_\ell} \rangle = \langle (\Delta^n p_j + \Delta^{n-j} q_j) q_\ell \rangle + \langle (\Delta^n p_\ell + \Delta^{n-\ell} q_\ell) q_j \rangle \\ &\leq \langle \Delta^n p_j + \Delta^{n-j} q_j \rangle + \langle q_\ell \rangle + \langle \Delta^n p_\ell + \Delta^{n-\ell} q_\ell \rangle + \langle q_j \rangle.\end{aligned}$$

Using  $\log_2(1 + |a + b|) \leq \log_2(1 + |a|) + \log_2(1 + b)$  for all  $b \geq 1$  and  $\log_2(1 + ab) \leq \log_2 a + \log_2(1 + b)$  for all  $a \geq 1, b \geq 0$  we find

$$\langle \Delta^n p_j + \Delta^{n-j} q_j \rangle \leq (2n - j) \lceil \log_2 \Delta \rceil + \langle p_j \rangle + \langle q_j \rangle - 1 \quad \text{for } 1 \leq j \leq n.$$

Using  $\langle d_\ell \rangle = 2$  and rather rough estimates we get

$$\langle \mathbf{d} \rangle \leq 3.5n(n-1) \lceil \log_2 \Delta \rceil + 2(n-1) \langle \mathbf{c} \rangle.$$

From the first two equations we get

$$\langle d_j \rangle \geq \langle q_j \rangle + \langle \Delta^n p_\ell + \Delta^{n-\ell} q_\ell \rangle \text{ for } 1 \leq j \neq \ell \leq n$$

and thus we have

$$\langle \mathbf{d} \rangle \geq \log_2(|\Delta^n p_\ell + \Delta^{n-\ell} q_\ell|) + \sum_{j \neq \ell} \log_2 q_j.$$

To prove  $\mathbf{d}(\mathbf{x}^{\max} - \mathbf{y}) \geq 2^{-8n^2\phi - \langle \mathbf{d} \rangle}$  we observe that by part (iii)  $\tilde{\mathbf{d}}(\mathbf{x}^{\max} - \mathbf{y}) \geq 2^{-8n^2\phi} (\prod_{j=1}^n q_j)^{-1}$ . Since  $\|\tilde{\mathbf{d}}\|_\infty = |\Delta^n c_\ell + \Delta^{n-\ell}|$ , the assertion follows from the lower bound on  $\langle \mathbf{d} \rangle$  that we just derived, and thus the proof of (iv) is complete.

**(v)** If  $\mathbf{c} = \mathbf{0}$  then  $S = S^{\neq}$  in part (iii) of this exercise and the assertion follows immediately from what we have proved under (iii) and the definition of a lexicographically maximal element of a set.

### Exercise 7.15

Write a computer program in a language of your choice for the division free Gaussian algorithm with and without Euclidean reduction.

The following program is an implementation of the division free Gaussian algorithm as a MATLAB function. For simplicity, we use the formula given in the statement of the iterative step of the algorithm on page 195 in the text, instead of using the rowwise and columnwise structures for the matrix  $B$ . The program is in a file called dfgauss.m.

```

%%%%%
%% This is the implementation of the Division-Free Gaussian Algorithm;
%% as found on pages 200-1. We DO NOT use sparse structures.
%%
%% NAME    : dfgauss
%% PURPOSE: Solve the system B x = a~where B is an m x m matrix
%%
%% INPUT   : The matrix B, the vector a.
%% OUTPUT  : stat: solution status, 1 if solved, 0 if infeasible
%%           x   : the solution to the system
%%           rk  : the rank of the matrix B
%%           E   : the set of rows in the order of pivoting
%%           J   : the set of columns in the order of pivoting
%%           Det : the determinant of the matrix B_J^E
%%           L,U : the L and U factors of B
%%           D   : the matrix D in B=LD^{-1}U
%%%%%
function [stat,x,xnum,xden,rk,Det,E,J,L,U,D]= dfgauss(B,a);
[m,n]=size(B);
k=0;
E=zeros(1,m);
J=zeros(1,m);
D=zeros(1,m);
Jind=zeros(1,m);
L=zeros(m,m);
U=zeros(m,m);
b=zeros(1,m);
rk=0;

h=1; j=1; d=1;
[h,j]=pivotel(E,J,B);
while ( h*j > 0)
    rk=rk+1;
    dold=d;
    d=B(h,j);
    D(k+1)=d;
    Det=d;
    l=B(:,j);
    u=B(h,:);
    L(:,k+1)=l;
    U(k+1,:)=u;
    % PIVOTING
    for i=h+1:m
        if abs(l(i))>1e-10
            E(i,:)=l;
            J(i)=j;
            L(i,:)=L(i,:)/l(i);
            U(i,:)=U(i,:)/l(i);
            for j=1:m
                if j~=i
                    L(i,j)=L(i,j)*l(j);
                    U(i,j)=U(i,j)*l(j);
                end;
            end;
        end;
    end;
end;

```

```

E(h)=1;
J(j)=1;
Jind(k+1)=j;
B=(d*B-l*u)/dold;
b(k+1)=a(h);
a=(d*a-b(k+1)*l)/dold;
k=k+1;
[h,j]=pivotel(E,J,B);
end

if (any(a ~= 0)),
    stat=0;
else
    x=zeros(m,1);
    xnum=zeros(m,1);
    xden=zeros(m,1);
    jcar=sum(J==1);
    for h=jcar:-1:1,
        psum=0;
        for i=h+1:jcar,
            psum=psum+U(h,Jind(i))*x(Jind(i));
        end
        xnum(Jind(h))=(b(h)*Det-psum);
        xden(Jind(h))=U(h,Jind(h));
        x(Jind(h))=xnum(Jind(h))/xden(Jind(h));
    end
    x=x/Det;
    stat=1;
end

```

The program dfgcall.m which follows, demonstrates the use of the function dfgauss.

```

%B=[2 0 0 0 0; 1 2 0 0 0; 1 1 2 0 0; 1 1 1 2 0; 1 1 1 1 2];
%a=[1;1;1;1;1];

T=[1 1 0; 1 0 1; 0 1 1];
Z=zeros(3,3);
B=[T Z Z; Z T Z; Z Z T];
a=ones(9,1);

[stat,x,xnum,xden,rk,Det,E,J,L,U,D]=dfgauss(B,a);
if (stat == 0),
    fprintf('The system has no feasible solution.\n')
else
    fprintf('The solution to the system is:\n')
    [m,n]=size(B);
    for i=1:m,

```

```

fprintf('x(%3g)=%10.5f',i,x(i))
fprintf('(xnum=%10.5f, xden=%10.5f, Det=%4g\n',xnum(i),xden(i),Det);
end
end

```

**Note:** The function dfgauss returns the numerator xnum and denominator xden used in the calculation of x to the system. We do that to show better the difference between this version of the algorithm and that with Euclidean reduction.

The algorithm with the Euclidean reduction is implemented in the function eudfgaus that follows:

```

%%%%%
%% This is the implementation of the Division-Free Gaussian Algorithm
%% using Euclidean reduction. We DO NOT use sparse structures.
%%
%% NAME    : dfgauss
%% PURPOSE: Solve the system B x = a~where B is an m x m matrix
%%
%% INPUT   : The matrix B, the vector a.
%% OUTPUT  : stat: solution status, 1 if solved, 0 if infeasible
%%           x   : the solution to the system
%%           rk  : the rank of the matrix B
%%           E   : the set of rows in the order of pivoting
%%           J   : the set of columns in the order of pivoting
%%           Det : the determinant of the matrix B_J^E
%%           L,U : the L and U factors of B
%%           D   : the matrix D in B=LD^{-1}U
%%%
function [stat,x,xnum,xden,rk,Det,E,J,L,U,D] = eudfgaus(B,a);
[m,n]=size(B);
k=0;
E=zeros(1,m);
J=zeros(1,m);
D=zeros(1,m);
Jind=zeros(1,m);
L=zeros(m,m);
U=zeros(m,m);
b=zeros(1,m);
rk=0;

h=1; j=1; d=1;
[h,j]=pivotel(E,J,B);
while ( h*j > 0)
    rk=rk+1;
    dold=d;
    d=B(h,j);
    % PIVOTING
    for i=h+1:m
        if abs(d) < 1e-10
            stat=0;
            break;
        end
        if abs(d) > 1e-10
            % Compute the pivot element
            pivot_element = d;
            % Compute the ratio
            ratio = -b(i)/pivot_element;
            % Update the matrix B
            for j=1:m
                B(i,j) = B(i,j) + ratio*B(h,j);
            end
            % Update the vector b
            b(i) = b(i) + ratio*b(h);
            % Update the matrix L
            for j=h+1:m
                L(i,j) = L(i,j) + ratio*L(h,j);
            end
            % Update the matrix U
            for j=h+1:m
                U(i,j) = U(i,j) + ratio*U(h,j);
            end
            % Update the matrix D
            D(i,i) = D(i,i) + ratio*D(h,h);
        end
    end
    % Swap rows
    if j < h
        % Swap rows
        for i=h+1:m
            temp = B(i,j);
            B(i,j) = B(j,i);
            B(j,i) = temp;
        end
        % Swap columns
        for i=h+1:m
            temp = L(i,j);
            L(i,j) = L(j,i);
            L(j,i) = temp;
        end
        % Swap vector b
        temp = b(j);
        b(j) = b(h);
        b(h) = temp;
    end
    % Increment row index
    h=h+1;
    j=j+1;
    d=d*sign(d);
end

```

```

D(k+1)=d;
Det=d;
l=B(:,j);
u=B(h,:);
L(:,k+1)=l;
U(k+1,:)=u;
E(h)=1;
J(j)=1;
Jind(k+1)=j;
B=d*B-l*u;
b(k+1)=a(h);
a=d*a-b(k+1)*l;
G=[B a];
for i=1:m,
    div=vecgcd(G(i,:));
    B(i,:)=B(i,:)/div;
    a(i)=a(i)/div;
end
k=k+1;
[h,j]=pivotel(E,J,B);
end

if (any(a ~= 0)),
    stat=0;
else
    x=zeros(m,1);
    xnum=zeros(m,1);
    xden=zeros(m,1);
    jcar=sum(J==1);
    for h=jcar:-1:1,
        psum=0;
        for i=h+1:jcar,
            psum=psum+U(h,Jind(i))*x(Jind(i));
        end
        xnum(Jind(h))=(b(h)*Det-psum);
        xden(Jind(h))=U(h,Jind(h));
        x(Jind(h))=xnum(Jind(h))/xden(Jind(h));
    end
    x=x/Det;
    stat=1;
end

```

and the program to call this function is edfgcall.m that follows:

```
%B=[2 0 0 0 0; 1 2 0 0 0; 1 1 2 0 0; 1 1 1 2 0; 1 1 1 1 2];
%a=[1;1;1;1;1];
```

```

T=[1 1 0; 1 0 1; 0 1 1];
Z=zeros(3,3);
B=[T Z Z; Z T Z; Z Z T];
a=ones(9,1);
[stat,x,xnum,xden,rk,Det,E,J,L,U,D]=eudfgaus(B,a);
if (stat == 0),
    fprintf('The system has no feasible solution.\n')
else
    fprintf('The solution to the system is:\n')
    [m,n]=size(B);
    for i=1:m,
        fprintf('x(%3g)=%10.5f',i,x(i))
        fprintf(' (xnum=%10.5f, xden=%10.5f, Det=%4g\n',xnum(i),xden(i),Det);
    end
end

```

We solve two systems of the form  $B\mathbf{x} = \mathbf{e}$  where in the first  $B$  is the matrix  $H$  of Exercise 7.10(iv) for  $n = 5$  and in the second  $B = K_{3n}$  with  $n = 3$  (see page 169 in the text). Running the two algorithms with these data we get:

```

>>clear
>>dfgacall
The solution to the system is:
x( 1)= 0.50000 (xnum= 32.00000, xden= 2.00000, Det= 32)
x( 2)= 0.25000 (xnum= 32.00000, xden= 4.00000, Det= 32)
x( 3)= 0.12500 (xnum= 32.00000, xden= 8.00000, Det= 32)
x( 4)= 0.06250 (xnum= 32.00000, xden= 16.00000, Det= 32)
x( 5)= 0.03125 (xnum= 32.00000, xden= 32.00000, Det= 32)
>>clear
>>edfgcall
The solution to the system is:
x( 1)= 0.50000 (xnum= 32.00000, xden= 2.00000, Det= 32)
x( 2)= 0.25000 (xnum= 32.00000, xden= 4.00000, Det= 32)
x( 3)= 0.12500 (xnum= 32.00000, xden= 8.00000, Det= 32)
x( 4)= 0.06250 (xnum= 32.00000, xden= 16.00000, Det= 32)
x( 5)= 0.03125 (xnum= 32.00000, xden= 32.00000, Det= 32)
>>clear
>>dfgacall
The solution to the system is:
x( 1)= 0.50000 (xnum= -4.00000, xden= 1.00000, Det= -8)
x( 2)= 0.50000 (xnum= 4.00000, xden= -1.00000, Det= -8)
x( 3)= 0.50000 (xnum= 8.00000, xden= -2.00000, Det= -8)
x( 4)= 0.50000 (xnum= 8.00000, xden= -2.00000, Det= -8)
x( 5)= 0.50000 (xnum= -8.00000, xden= 2.00000, Det= -8)
x( 6)= 0.50000 (xnum= -16.00000, xden= 4.00000, Det= -8)
x( 7)= 0.50000 (xnum= -16.00000, xden= 4.00000, Det= -8)
x( 8)= 0.50000 (xnum= 16.00000, xden= -4.00000, Det= -8)

```

```
x( 9) = 0.50000 (xnum= 32.00000, xden= -8.00000, Det= -8)
>>clear
>>edfgcall
The solution to the system is:
x( 1)= 0.50000 (xnum= -1.00000, xden= 1.00000, Det= -2)
x( 2)= 0.50000 (xnum= 1.00000, xden= -1.00000, Det= -2)
x( 3)= 0.50000 (xnum= 2.00000, xden= -2.00000, Det= -2)
x( 4)= 0.50000 (xnum= -1.00000, xden= 1.00000, Det= -2)
x( 5)= 0.50000 (xnum= 1.00000, xden= -1.00000, Det= -2)
x( 6)= 0.50000 (xnum= 2.00000, xden= -2.00000, Det= -2)
x( 7)= 0.50000 (xnum= -1.00000, xden= 1.00000, Det= -2)
x( 8)= 0.50000 (xnum= 1.00000, xden= -1.00000, Det= -2)
x( 9)= 0.50000 (xnum= 2.00000, xden= -2.00000, Det= -2)
>>
```

---

### Exercise 7.16

- (i) Let  $H = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ ,  $d = a - c + w$ ,  $w = \sqrt{(a - c)^2 + 4b^2}$  and  $r = \sqrt{d^2 + 4b^2}$ . Show that  $\lambda_1 = (a + c + w)/2$ ,  $\lambda_2 = (a + c - w)/2$  are the two eigenvalues of  $H$ . Show that the vectors  $x^1$  with components  $x_1^1 = d/r$ ,  $x_2^1 = 2b/r$  and  $x^2$  with components  $x_1^2 = 2b/r$ ,  $x_2^2 = -d/r$  are the eigenvectors of  $H$  for  $\lambda_1$  and  $\lambda_2$ , respectively, and that the matrix  $X = (x^1 \ x^2)$  is an orthogonal matrix.
- (ii) Let  $H$  be a positive definite matrix of size  $n \times n$  and  $H = X\Lambda X^T$  as above; see the remark after the proof of point 7.7(e) in the text. Show that  $H^{1/2} := X\Lambda^{1/2}X^T$  is the unique positive definite "square root" of  $H$ , i.e.  $H = H^{1/2}H^{1/2}$ ,  $H^{1/2}$  is positive definite (and thus symmetric) and  $H = KK$  for some  $n \times n$  matrix  $K$  implies  $K = X\Lambda^{1/2}X^T$ , where  $\Lambda^{1/2} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$ ,  $\lambda_i$  are the eigenvalues and  $X$  the orthogonal matrix of the eigenvectors of  $H$ .
- (iii) Let  $u \in \mathbb{R}^n$ ,  $\|u\| = 1$  and  $\alpha \in \mathbb{R}$ . Show that  $\det(\alpha uu^T - \lambda I_n) = (-\lambda)^{n-1}(\alpha - \lambda)$ .
- 

- (i) To find the eigenvalues  $\lambda$  of  $H$  we solve the equation  $\det(H - \lambda I) = 0$ , i.e.  $(a - \lambda)(c - \lambda) - b^2 = 0$ . Thus the two eigenvalues of  $H$  are the two roots of the quadratic equation  $\lambda^2 - (a+c)\lambda + ac - b^2 = 0$ , i.e.

$$\lambda = \frac{a + c \pm \sqrt{(a + c)^2 - 4ac + 4b^2}}{2} = \frac{a + c \pm \sqrt{(a - c)^2 + 4b^2}}{2} = \frac{a + c \pm w}{2}.$$

So let  $\lambda_1 = (a + c + w)/2$  and  $\lambda_2 = (a + c - w)/2$ . To find the eigenvectors corresponding to an eigenvalue  $\lambda$  we solve the system of equations  $Hx = \lambda x$ , i.e.  $ax_1 + bx_2 = \lambda x_1$  and  $bx_1 + cx_2 = \lambda x_2$ . Since  $\lambda$  is an eigenvalue, the determinant of the system is zero, and thus the system has an infinite number of solutions. We normalize the eigenvector by requiring  $x_1^2 + x_2^2 = 1$  to get a second equation. For  $\lambda = \lambda_1$  we have by solving  $bx_1 + cx_2 = \lambda_1 x_1$  for  $x_1$  that  $x_1 = \frac{\lambda_1 - c}{b}x_2$  and

from the normalization equation we get  $(\frac{(\lambda_1 - c)^2}{b^2} + 1)x_2^2 = 1$  which yields  $x_2^2 = \frac{b^2}{(\lambda_1 - c)^2 + b^2}$ . Now,  $\lambda_1 - c = \frac{a+c+w}{2} - c = \frac{a+w-c}{2} = \frac{d}{2}$ . Thus  $x_2^2 = \frac{4b^2}{d^2 + 4b^2} = \frac{4b^2}{r^2}$  and  $x_2 = \frac{2b}{r}$ , where we pick arbitrarily the positive square root. Substituting in the equation for  $x_1$  we get  $x_1 = \frac{\lambda_1 - c}{b} \frac{2b}{r} = \frac{d}{r}$ . Thus the eigenvector corresponding to  $\lambda_1$  is  $x^1 = (\frac{d}{r}, \frac{2b}{r})^T$ .

For  $\lambda = \lambda_2$ , we solve the first equation for  $x_2$  to get  $x_2 = \frac{\lambda_2 - a}{b}x_1$ , and substituting in  $x_1^2 + x_2^2 = 1$  and rearranging we get  $x_1^2 = \frac{b^2}{b^2 + (\lambda_2 - a)^2}$ . Now,  $\lambda_2 - a = \frac{a+c-w}{2} - a = \frac{c-w-a}{2} = -\frac{d}{2}$ . Thus  $x_1^2 = \frac{4b^2}{4b^2 + d^2} = \frac{4b^2}{r^2}$ , i.e.  $x_1 = \frac{2b}{r}$  (we arbitrarily pick the positive square root) and  $x_2 = \frac{\lambda_2 - a}{b}x_1 = -\frac{d}{2b} \frac{2b}{r} = -\frac{d}{r}$ . Thus the eigenvector corresponding to  $\lambda_2$  is  $x^2 = (\frac{2b}{r}, -\frac{d}{r})^T$ .

Finally to show that  $X = (x^1 \ x^2)$  is orthogonal we calculate  $\frac{d}{r} \frac{2b}{r} - \frac{2b}{r} \frac{d}{r} = 0$ .

**(ii)** We have  $H^{1/2}H^{1/2} = X\Lambda^{1/2}X^TX\Lambda^{1/2}X^T = X\Lambda^{1/2}\Lambda^{1/2}X^T = X\Lambda X^T = H$ , where we have used  $XX^T = I_n$ . To show that  $H^{1/2}$  is positive definite for any vector  $x \in \mathbb{R}^n$  we calculate  $x^T H^{1/2} x = x^T X \Lambda^{1/2} X^T x = (X^T x)^T \Lambda^{1/2} (X^T x) > 0$  since  $\Lambda^{1/2} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$  is positive definite. To show that  $H^{1/2}$  is unique, suppose that  $H = KK$  for some  $K \neq H^{1/2}$ . From  $H = X\Lambda X^T$  we have  $X^T H X = \Lambda$  and substituting  $H = KK$  we get  $X^T K^2 X = \Lambda$ . Thus  $\Lambda^{1/2} = X^T K X$  and  $H^{1/2} = X\Lambda^{1/2}X^T = XX^T K XX^T = K$ .

**(iii)** We claim more generally that  $\det(\alpha u u^T - \lambda I_n) = (-\lambda)^{n-1}(\alpha \|u\|^2 - \lambda)$  for all  $u \in \mathbb{R}^n$ . The formula is clearly correct if  $\lambda = 0$  (see also Exercise 4.1) or  $u = 0$ . So assume that  $\lambda \neq 0$  and  $u \neq 0$ . WROG let  $u_1 \neq 0$ . Multiplying row 1 of the matrix by  $-u_i/u_1$  and adding the result to row  $i$  where  $1 \leq i \leq n$  we bring the matrix  $\alpha u u^T - \lambda I_n$  by elementary row operations into the form

$$\begin{pmatrix} \alpha u_1^2 - \lambda & \alpha u_1 u \\ \frac{\lambda}{u_1} v^T & -\lambda I_{n-1} \end{pmatrix} = \begin{pmatrix} \alpha \|u\|^2 - \lambda & -\frac{\alpha}{\lambda} u_1 v \\ 0 & I_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\lambda}{u_1} v^T & -\lambda I_{n-1} \end{pmatrix}$$

where  $v = (u_2, \dots, u_n)$ . Taking the determinant the formula follows.

### \*Exercise 7.17

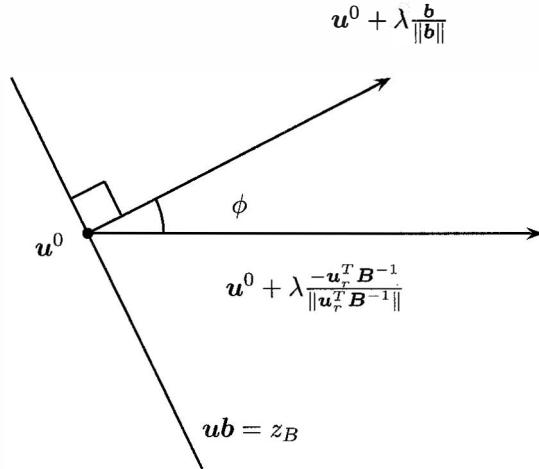
Give a geometric interpretation of the normed pivot row selection criteria for the dual simplex algorithm using the partial and/or full row norms of Exercise 6.12. (Hint: Review Chapters 6.4 and 7.6.1 first.)

The dual simplex algorithm solves the linear program in standard form  $\min\{cx : Ax = b, x \geq 0\}$  by working on the dual linear program  $\max\{ub : uA \leq c\}$ . We assume that a dual feasible basis  $B$  is at hand and that we move along edges of the dual polyhedron

$$\mathcal{U}^\leq = \{u \in \mathbb{R}^m : uA \leq c\},$$

as we maintain dual feasibility from iteration to iteration. For a geometric interpretation we can also embed  $\mathcal{U}^\leq$  in a higher dimensional space using slack variables and consider the polyhedron

$$\mathcal{U}^= = \{(u, v) \in \mathbb{R}^{m+n} : uA + v = c, v \geq 0\}.$$



**Fig. 7.8.** Steepest ascent for  $\mathcal{U}^<$  in Exercise 7.17

The polyhedron  $\mathcal{U}^<$  is obtained from  $\mathcal{U}^=$  by projecting out the slack variables (see Exercise 7.5(ii)) and we can interpret the way the dual simplex algorithm proceeds on either polyhedron.

Consider first the polyhedron  $\mathcal{U}^<$ . At every iteration we have a dual basis  $B$  and its corresponding matrix  $R$  of nonbasic columns of  $A$ . The current solution is  $u^0 = c_B B^{-1}$  and we move away from it along an “edge” of  $\mathcal{U}^<$  of the form  $u(\lambda) = u^0 - \lambda u_r^T B^{-1}$ , where  $r$  is the pivot row and  $\lambda \geq 0$ . The pivot row is selected such that  $\bar{b}_r < 0$  where  $\bar{b} = B^{-1}b$  is the current right-hand side. If the normed pivot row selection criterion using the partial norms  $\delta_i = \|u_i^T B^{-1}\|$  is used in line (6.7) of the dual simplex algorithm then we determine  $r$  such that  $\bar{b}_r/\delta_r$  is as small as possible. Consider now the dual objective function  $u(\lambda)b = z_B + \lambda(-\bar{b}_r)$  where  $z_B = c_B B^{-1}b$ . Like in the discussion of the normed pivot column selection on page 189 of the text the direction of “steepest” ascent is given by  $b/\|b\|$  where we assume WROG that  $b \neq 0$  (see Remark 3.1). This is so because  $b/\|b\|$  is perpendicular to the hyperplane  $ub = z_B$ . Thus from among the “feasible” directions for the dual simplex algorithm we want the ones that are as close as possible to this direction of steepest ascent because we are maximizing the dual LP. So we calculate the cosine of the angle  $\phi$  between  $b/\|b\|$  and a direction  $w = -u_r^T B^{-1}$  that we consider. We find

$$\cos \phi = \frac{wb}{\|b\|\|w\|} = \frac{-\bar{b}_r}{\|b\|\|u_r^T B^{-1}\|}.$$

Since  $\|b\|$  is constant, the normed pivot row selection thus finds the direction that is “as close as possible” to the direction vector  $b/\|b\|$  because we choose a row  $r$  such that  $\cos \phi$  is as large as possible, i.e., such that  $\phi$  is as close as possible to zero, see Figure 7.8. Like in the case of the normed pivot column selection discussed on page 189 of the book, degeneracy must be taken into account in this interpretation.

Suppose we interpret the dual simplex algorithm on the polyhedron  $\mathcal{U}^=$ . Then the current basis gives the feasible point  $(u^0, v^0)$  where  $u^0 = c_B B^{-1}$  and  $v^0 = c - u^0 A \geq 0$ . The current objective function value gives the hyperplane  $ub + 0v = z_B$  and we are interested in finding the

direction that is as close as possible  $\|b\|^{-1} \begin{pmatrix} b \\ 0 \end{pmatrix}$ . To keep the notation simple let us assume that

$B$  occurs in the first  $m$  columns and  $R$  in the remaining  $n - m$  columns of  $A$ . Accordingly we partition  $v$  into  $v_B$  and  $v_R$  and note that  $v_B^0 = 0$  in the current solution. The direction vectors  $w$  for a move on  $\mathcal{U}^+$  that the dual simplex algorithm considers are then of the form

$$w = (-u_r^T B^{-1}, u_r^T, u_r^T B^{-1} R),$$

and we leave it to you to verify that  $(u^0, v^0) + \lambda w \in \mathcal{U}^+$  for all  $0 \leq \lambda \leq \gamma$ , where  $\gamma$  is determined by the minimum ratio test (6.8) of the dual simplex algorithm. Consequently, calculating the cosine

of the angle  $\hat{\phi}$  between  $\|b\|^{-1} \begin{pmatrix} b \\ 0 \end{pmatrix}$  and  $w$  we find

$$\cos \hat{\phi} = \frac{w \begin{pmatrix} b \\ 0 \end{pmatrix}}{\|b\| \|w\|} = \frac{-\bar{b}_r}{\|b\| \sqrt{1 + \|u_r^T B^{-1}\|^2 + \|u_r^T B^{-1} R\|^2}}.$$

Thus the normed pivot row selection criterion “find the most negative  $\bar{b}_r/d_r$ ” selects a direction for  $\mathcal{U}^+$  that is a feasible direction of steepest ascent.

The two normed pivot row selection criteria are evidently different from each other and – empirically – behave differently as well. The partial norms  $\delta_r$  seem to perform about as good in numerical calculation as the full norms  $d_r$  and both “beat” the usual pivot row selection criterion based on a most negative  $\bar{b}_r$  on large-scale linear programs substantially; see J.J. Forrest and D. Goldfarb, “Steepest-edge simplex algorithms for linear programming”, *Mathematical Programming*, 57(1992), 341-374. From a theoretical point of view there is no reason – other than the geometry just outlined – to explain the differences in algorithmic behavior.

## 8. Projective Algorithms

Ἄστρος ὁ Θεός ὁ μέγας γεωμετρεῖ ...<sup>1</sup>  
Disciples of Euclid (c. 300 B.C.)

In this chapter we give a summary of the essentials of Chapter 8 of the text without proofs. Using an artificial variable  $x_{n+1}$  with the column  $\mathbf{a}_{n+1} = -\sum_{j=1}^n \mathbf{a}_j + \mathbf{b}$  and setting  $c_{n+1} = M$  in the objective function like in the Big-M method and redefining  $n+1$  to be  $n$ , we can ensure –if necessary– that the linear program in standard form

$$(LP) \quad \min \mathbf{c}\mathbf{x} \quad \text{subject to} \quad \mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

has a feasible solution  $\mathbf{x}^0$  with  $\mathbf{x}^0 > \mathbf{0}$ . We assume WROG that (LP) has such a solution and that  $\mathbf{c}\mathbf{x}$  is not a constant over  $\mathcal{X}$ . We rewrite (LP) as follows

$$(LP^*) \quad \min\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} - \mathbf{b}x_{n+1} = \mathbf{0}, x_{n+1} = 1, \mathbf{x} \geq \mathbf{0}, x_{n+1} \geq 0\}$$

where  $x_{n+1}$  is a “new” variable. Geometrically, we are embedding  $\mathbb{R}^n$  into  $\mathbb{R}^{n+1}$  by identifying a point  $\mathbf{x} \in \mathbb{R}^n$  with the point  $(\mathbf{x}, 1) \in \mathbb{R}^{n+1}$  (see Figure 8.1). Consider the projective transformation  $T_0$

$$y_j = \frac{x_j/x_j^0}{1 + \sum_{j=1}^n x_j/x_j^0} \quad \text{for } j = 1, \dots, n, \quad y_{n+1} = \frac{1}{1 + \sum_{j=1}^n x_j/x_j^0},$$

which maps the nonnegative “orthant”  $\{(\mathbf{x}, 1) \in \mathbb{R}^{n+1} : \mathbf{x} \geq \mathbf{0}\}$  into the  $n$ -dimensional simplex

$$S^{n+1} = \left\{ \mathbf{y} \in \mathbb{R}^{n+1} : \sum_{j=1}^{n+1} y_j = 1, \mathbf{y} \geq \mathbf{0} \right\}.$$

The point  $(\mathbf{x}^0, 1) \in \mathbb{R}^{n+1}$  is mapped into the “center”  $\mathbf{y}^0 = [1/(n+1)]\mathbf{f}$  of the simplex  $S^{n+1}$  where  $\mathbf{f}^T = (1, \dots, 1)$  is the vector with  $n+1$  components equal to one. We write  $\mathbf{y} = T_0(\mathbf{x})$  to denote the image  $\mathbf{y} \in \mathbb{R}^{n+1}$  of the point  $(\mathbf{x}, 1) \in \mathbb{R}^{n+1}$  where  $\mathbf{x} \in \mathbb{R}^n$ . Denote  $\mathbf{D} = \text{diag}(x_1^0, \dots, x_n^0)$  the  $n \times n$  matrix with diagonal elements  $x_i^0$  for  $i = 1, \dots, n$  and zeroes elsewhere. The transformation  $T_0$  becomes

$$(T_0) \quad \begin{pmatrix} \mathbf{y}_N \\ y_{n+1} \end{pmatrix} = \frac{1}{1 + \mathbf{e}^T \mathbf{D}^{-1} \mathbf{x}} \begin{pmatrix} \mathbf{D}^{-1} \mathbf{x} \\ 1 \end{pmatrix},$$

where  $\mathbf{e}^T = (1, \dots, 1)$  is the vector with  $n$  components equal to one and  $\mathbf{y}_N^T = (y_1, \dots, y_n)$  is the subvector of the  $n$  first components of the vector  $\mathbf{y} \in \mathbb{R}^{n+1}$ . The inverse of the projective transformation is

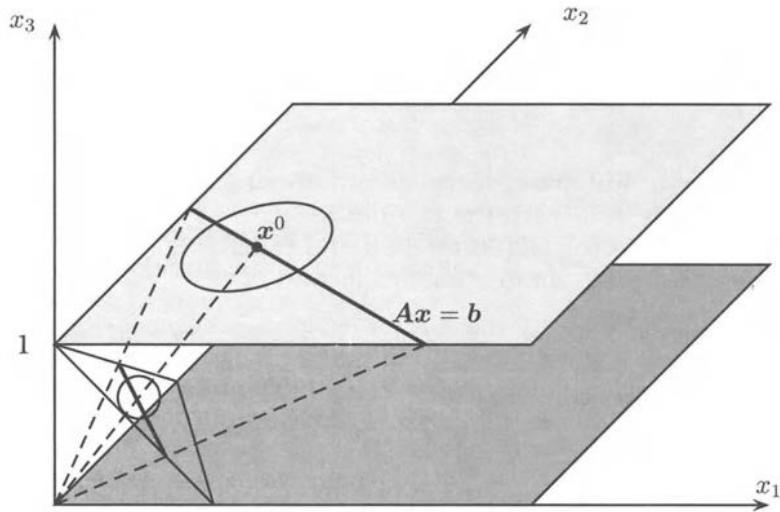
$$(T_0^{-1}) \quad \mathbf{x} = (1/y_{n+1}) \mathbf{D} \mathbf{y}_N$$

and  $x_{n+1} = 1$ . It follows that the image  $T_0(\mathcal{X})$  of the set  $\mathcal{X}$  under  $T_0$  is

$$T_0(\mathcal{X}) = \left\{ \mathbf{y} \in \mathbb{R}^{n+1} : (\mathbf{A}\mathbf{D}, -\mathbf{b})\mathbf{y} = \mathbf{0}, \mathbf{f}^T \mathbf{y} = 1, \mathbf{y} \geq \mathbf{0} \right\}.$$

---

<sup>1</sup>God the Almighty always does geometry.....



**Fig. 8.1.** Embedding of  $\mathbb{R}^n$  into  $\mathbb{R}^{n+1}$  or  $P_n$  for  $n = 2$

Furthermore,  $T_0(\mathcal{X}) \subseteq S^{n+1}$ ,  $\mathbf{y}^0 \in T_0(\mathcal{X})$  and (LP) becomes the nonlinear programming problem

$$(FLP) \quad \min \left\{ \frac{\mathbf{c}^T \mathbf{y}_N}{y_{n+1}} : \mathbf{y} \in T_0(\mathcal{X}) \right\}.$$

Denote the intersection of the  $(n+1)$ -dimensional ball with radius  $\rho$  and center  $\mathbf{y}^0$  with  $\mathbf{f}^T \mathbf{y} = 1$  by

$$B_\rho^{n+1} = \left\{ \mathbf{y} \in \mathbb{R}^{n+1} : \sum_{j=1}^{n+1} y_j = 1, \sum_{j=1}^{n+1} \left( y_j - \frac{1}{n+1} \right)^2 \leq \rho^2 \right\}$$

which gives an  $n$ -dimensional ball in the simplex  $S^{n+1}$  if the radius  $\rho$  is “small” enough. Let

$$r^2 = 1/n(n+1). \quad (8.1)$$

If  $0 \leq \rho \leq r$  then  $B_\rho^{n+1} \subseteq S^{n+1}$ ; see Exercise 8.1. We can thus replace (FLP) by

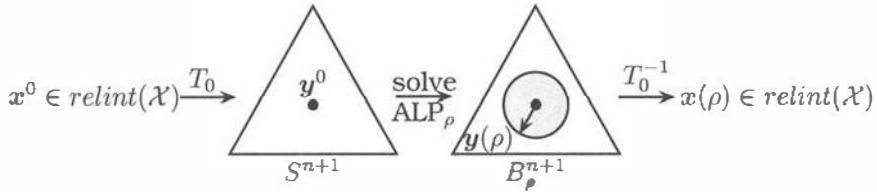
$$(FLP_\rho) \quad \min \left\{ \frac{(\mathbf{c}^T \mathbf{D}, 0) \mathbf{y}}{y_{n+1}} : (\mathbf{A} \mathbf{D}, -\mathbf{b}) \mathbf{y} = \mathbf{0}, \mathbf{y} \in B_\rho^{n+1} \right\},$$

where  $0 \leq \rho < r$  ensures that  $\mathbf{y} > \mathbf{0}$ . Thus for all  $0 \leq \rho < r$  we have

$$\min \{ \mathbf{c}^T \mathbf{x} : \mathbf{x} \in \mathcal{X} \} \leq \min \left\{ \frac{(\mathbf{c}^T \mathbf{D}, 0) \mathbf{y}}{y_{n+1}} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1} \right\}. \quad (8.2)$$

The problem (FLP $_\rho$ ) is a *restriction* of (FLP) and it is a classical nonlinear optimization problem that can be solved exactly. In the original approach the objective function of (FLP) is linearized as follows:

$$(ALP) \quad \min \{ (\mathbf{c}^T \mathbf{D}, 0) \mathbf{y} : \mathbf{y} \in T_0(\mathcal{X}) \}$$



**Fig. 8.2.** The iterative step of projective algorithms

and in lieu of problem  $(FLP_\rho)$  one solves the approximate problem

$$(ALP_\rho) \quad \min\{(cD, 0)\mathbf{y} : (AD, -b)\mathbf{y} = \mathbf{0}, \mathbf{y} \in B_\rho^{n+1}\}.$$

In either case, once a solution to  $(FLP_\rho)$  or to  $(ALP_\rho)$  has been obtained, one can use the inverse transformation  $T_0^{-1}$  of the projective transformation  $T_0$  to obtain a “new” interior point  $x^1 \in \mathcal{X}$  which gives rise to a new projective transformation  $T_1$ , etc, see Figure 8.2.

## 8.1 A Basic Algorithm

Aller Anfang ist schwer.<sup>2</sup>  
German proverb

We approximate  $(FLP)$  by  $(ALP)$  and make the **additional assumptions** that  $\mathcal{X}$  is **bounded** and that the optimal objective function value of  $(LP)$  equals **zero**. We discuss later how to remove these assumptions. It follows that the optimal objective function value of  $(ALP)$  equals zero as well no matter what interior point  $x^0 \in \mathcal{X}$  is used in the projective transformation  $T_0$ .

By the additional assumption that we have made, it follows that an optimal solution to the restriction  $(ALP_\rho)$  of  $(ALP)$  exists and its optimal objective function value is nonnegative for all  $0 \leq \rho \leq r$ . The following remark summarizes the key facts about the solution of  $(ALP_\rho)$ .

**Remark 8.1** Consider the linear program  $\min\{cz : Az = \mathbf{0}, e^T z = 1, z \geq \mathbf{0}\}$  where  $A$  is an  $m \times n$  matrix of rank  $m$ ,  $z^0 = (1/n)e$  is a nonoptimal feasible solution and the optimal objective function value equals zero, i.e.  $cz^0 > 0$ . Then for all  $\rho \geq 0$  an optimal solution to

$$(P_\rho) \quad \min\{cz : Az = \mathbf{0}, z \in B_\rho^n\}$$

is given by  $z(\rho) = (1/n)e - \rho p / \|p\|$  where  $p = (I_n - A^T(AA^T)^{-1}A - (1/n)ee^T)c^T$  is the orthogonal projection of  $c$  on the subspace  $\{z \in \mathbb{R}^n : Az = \mathbf{0}, e^T z = 0\}$ . Moreover, for all  $\rho \geq 0$  the optimal solution  $z(\rho)$  satisfies  $cz(\rho)/cz^0 \leq 1 - \rho\sqrt{n/(n-1)}$ .

### 8.1.1 The Solution of the Approximate Problem

To solve the problem  $(ALP_\rho)$  we need the orthogonal projection of the vector  $(cD, 0)$  on the subspace

$$\{\mathbf{y} \in \mathbb{R}^{n+1} : (AD, -b)\mathbf{y} = \mathbf{0}, f^T \mathbf{y} = 0\}. \quad (8.3)$$

<sup>2</sup>All beginning is difficult...

To this end we need the inverse of the matrix

$$\widehat{\mathbf{G}} = \begin{pmatrix} \mathbf{AD} & -\mathbf{b} \\ \mathbf{e}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{DA}^T & \mathbf{e} \\ -\mathbf{b}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{AD}^2\mathbf{A}^T + \mathbf{bb}^T & \mathbf{0} \\ \mathbf{0} & n+1 \end{pmatrix}.$$

Denote  $\mathbf{G} = \mathbf{AD}^2\mathbf{A}^T$  and note that  $\mathbf{G}$  is positive definite. Thus  $\mathbf{G}^{-1}$  exists and the inverse of  $\widehat{\mathbf{G}}$  is

$$\widehat{\mathbf{G}}^{-1} = \begin{pmatrix} \mathbf{G}^{-1} - (1+\beta)^{-1}(\mathbf{G}^{-1}\mathbf{b})(\mathbf{b}^T\mathbf{G}^{-1}) & \mathbf{0} \\ \mathbf{0} & (n+1)^{-1} \end{pmatrix},$$

where  $\beta = \mathbf{b}^T\mathbf{G}^{-1}\mathbf{b} \geq 0$  since  $\mathbf{G}^{-1}$  is positive definite as well. Denote

$$\mathbf{P} = \mathbf{I}_n - \mathbf{D}\mathbf{A}^T\mathbf{G}^{-1}\mathbf{AD}, \quad \mathbf{p} = \mathbf{PD}\mathbf{c}^T, \quad \mathbf{d} = \mathbf{Pe}, \quad (8.4)$$

i.e.  $\mathbf{p}$  is the orthogonal projection of  $\mathbf{D}\mathbf{c}^T$  and  $\mathbf{d}$  the orthogonal projection of  $\mathbf{e}$  on the subspace

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{ADx} = \mathbf{0}\}. \quad (8.5)$$

The projection operator  $\mathbf{Q}$  on the subspace (8.3) is calculated to be

$$\mathbf{Q} = \begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix} + (1+\beta)^{-1} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} \left( \mathbf{e}^T - \mathbf{d}^T, 1 \right) - (n+1)^{-1} \mathbf{ff}^T$$

and consequently the orthogonal projection of  $(\mathbf{cD}, 0)$  is given by

$$\mathbf{q} = \mathbf{Q} \begin{pmatrix} \mathbf{D}\mathbf{c}^T \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} + \frac{z_0 - \gamma}{1+\beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} - \frac{z_0}{n+1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix}, \quad (8.6)$$

where we have set  $z_0 = \mathbf{c}\mathbf{x}^0$  and  $\gamma = \mathbf{p}^T\mathbf{d}$ . For further reference we note that necessarily  $\mathbf{p} \neq 0$  if  $\mathbf{x}^0$  is a *nonoptimal* solution to (LP). Moreover,  $\gamma = \mathbf{p}^T\mathbf{e} = \mathbf{cDd}$  by the properties of orthogonal projections. By the definition of  $\mathbf{d}$  and  $\beta$ ,  $\beta = n - \|\mathbf{d}\|^2 = \|\mathbf{e} - \mathbf{d}\|^2$  and  $\|\mathbf{d}\|^2 \leq n$  because  $\beta \geq 0$ .  $\|\mathbf{q}\|$  is calculated using the fact that  $\|\mathbf{q}\|^2 = (\mathbf{cD}, 0) \mathbf{q}$ . It follows from Remark 8.1 that the solution to  $(\text{ALP}_\rho)$  is

$$\mathbf{y}^K(\rho) = \mathbf{y}^0 - \rho \mathbf{q} / \|\mathbf{q}\|, \quad (8.7)$$

that  $(\mathbf{cD}, 0) \mathbf{y}^K(\rho) = (z_0/(n+1)) - \rho \|\mathbf{q}\|$  and that for all  $\rho \geq 0$

$$\frac{(\mathbf{cD}, 0) \mathbf{y}^K(\rho)}{(\mathbf{cD}, 0) \mathbf{y}^0} \leq 1 - \rho \sqrt{(n+1)/n}. \quad (8.8)$$

Reversing the projective transformation we find that the new iterate  $\mathbf{x}^K(\rho) \in \text{relint}(\mathcal{X})$  is

$$\mathbf{x}^K(\rho) = \mathbf{x}^0 - t(\rho) \mathbf{D} \left( \mathbf{p} - \frac{z_0 - \gamma}{1+\beta} \mathbf{d} \right), \text{ where } t(\rho) = \frac{(1+\beta)(n+1)\rho}{(1+\beta)\|\mathbf{q}\| + \rho(\gamma(n+1) - (n-\beta)z_0)} \quad (8.9)$$

and that the objective function value of  $\mathbf{x}^K(\rho)$  is given by

$$\mathbf{c}\mathbf{x}^K(\rho) = z_0 - t(\rho) [\|\mathbf{p}\|^2 - \gamma(z_0 - \gamma)/(1+\beta)]. \quad (8.10)$$

### 8.1.2 Convergence of the Approximate Iterates

Like the solution (8.7) to  $(ALP_\rho)$ , the loci of  $\mathbf{x}^K(\rho)$  given by (8.9) form a *line* in  $\mathbb{R}^n$ . To prove convergence of the sequence of points generated by an iterative application of the basic idea consider the function

$$h(\mathbf{x}) = \mathbf{c}\mathbf{x} \left( \prod_{j=1}^n x_j \right)^{-1/(n+1)}, \quad (8.11)$$

which is the objective function divided by the *geometric mean* of the point  $(\mathbf{x}, 1) \in \mathbb{R}^{n+1}$ . It follows that

$$\frac{h(\mathbf{x}^K(\rho))}{h(\mathbf{x}^0)} \leq (1 - \rho \sqrt{(n+1)/n}) \left( \prod_{j=1}^{n+1} (n+1)y_j^K(\rho) \right)^{-1/(n+1)} \quad (8.12)$$

using (8.8) and we are left with estimating the last term for  $\mathbf{y} \in B_\rho^{n+1}$ . The next remark gives a best possible estimation, where  $\rho = \alpha r$  with  $0 \leq \alpha \leq 1$ .

**Remark 8.2** Let  $\rho = \alpha r$  where  $r^2 = 1/n(n-1)$ . Then for all  $0 < \alpha < 1$

$$\max \left\{ \left( \prod_{j=1}^n nz_j \right)^{-1/n} : \mathbf{z} \in B_\rho^n \right\} = [1 + \alpha/(n-1)]^{-1} [(1 + \alpha/(n-1))/(1 - \alpha)]^{1/n}.$$

From Remark 8.2 and (8.12) it follows that for all  $0 \leq \alpha < 1$

$$\frac{h(\mathbf{x}^K(\alpha r))}{h(\mathbf{x}^0)} \leq \frac{1 - \alpha/n}{1 + \alpha/n} \left( \frac{1 + \alpha/n}{1 - \alpha} \right)^{1/(n+1)} = \tilde{g}(\alpha, n). \quad (8.13)$$

We estimate  $\tilde{g}(\alpha, n)$  more conveniently and find

$$\tilde{g}(\alpha, n) \leq \frac{e^{-\frac{2\alpha}{n}} - \frac{\alpha}{n(n+1)} + \frac{\alpha}{n(n+1)}}{(1 - \alpha)^{1/n}} = \left( \frac{e^{-2\alpha}}{1 - \alpha} \right)^{1/n} = g(\alpha, n). \quad (8.14)$$

It follows that  $g(\alpha, n) < 1$  for all  $0 < \alpha < \alpha_0 = 0.7968\dots$ . Hence the iterative application produces a geometric convergence rate in terms of  $h(\mathbf{x})$  for any fixed “step-size”  $\alpha$  satisfying  $0 < \alpha < 0.7968\dots$ .

#### Basic Algorithm ( $\alpha, p, m, n, \mathbf{A}, \mathbf{c}, \mathbf{x}^0$ )

**Step 0:** Set  $D_0 := \text{diag}(x_1^0, \dots, x_n^0)$ ,  $z := \mathbf{c}\mathbf{x}^0$  and  $k := 0$ .

**Step 1:** Compute  $\mathbf{G} := \mathbf{A}D_k^2\mathbf{A}^T$ ,  $\mathbf{G}^{-1}$  and  $\mathbf{P} := \mathbf{I}_n - D_k\mathbf{A}^T\mathbf{G}^{-1}\mathbf{A}D_k$ .

**Step 2:** Compute  $\mathbf{p} := \mathbf{P}\mathbf{D}_k\mathbf{c}^T$ ,  $\mathbf{d} := \mathbf{P}\mathbf{e}$ ,  $\gamma := \mathbf{p}^T\mathbf{d}$ ,  $\beta := n - \|\mathbf{d}\|^2$ ,

$$\|\mathbf{q}\| := \sqrt{\|\mathbf{p}\|^2 + (z - \gamma)^2/(1 + \beta) - z^2/(n + 1)} \text{ and}$$

$$t := \frac{\alpha(1 + \beta)(n + 1)}{(1 + \beta)\sqrt{n(n + 1)}\|\mathbf{q}\| + \alpha(\gamma(n + 1) - (n - \beta)z)}.$$

**Step 3:** Set  $\mathbf{x}^{k+1} := \mathbf{x}^k - tD_k \left( \mathbf{p} - \frac{z - \gamma}{1 + \beta} \mathbf{d} \right)$ ,  $D_{k+1} := \text{diag}(x_1^{k+1}, \dots, x_n^{k+1})$ .

**Step 4:** if  $\frac{\mathbf{c}\mathbf{x}^{k+1}}{\mathbf{c}\mathbf{x}^0} < 2^{-p}$  stop “ $\mathbf{x}^{k+1}$  is a  $p$ -optimal solution to  $(LP)$ ”.

Set  $z := \mathbf{c}\mathbf{x}^{k+1}$ ; replace  $k + 1$  by  $k$ ; go to Step 1.

### 8.1.3 Correctness, Finiteness, Initialization

**Remark 8.3** For every  $0 < \alpha < 0.7968\dots$  and  $p \geq \log_2 K$  the basic algorithm iterates at most  $\mathcal{O}(np)$  times where  $K \geq 2$  is such that  $\mathcal{X} \subseteq \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq K \text{ for } j = 1, \dots, n\}$ .

The practical experience with the basic algorithm and its derivatives has been good and indeed, in terms of the number of steps, it is far better than suggested by the analysis. As we shall see in Chapter 8.5 – after an analysis of  $(FLP_p)$  – the step complexity of projective algorithms can be improved substantially.

Two issues remain to be addressed. The first one concerns how to start the basic algorithm and the second one how to get a *basic feasible* solution to the linear program. Practice and theory diverge – as they do so often – on both of these points. We will be brief and discuss the “theoretical” side of the coin only.

The second issue can be dealt with by a modification of the proof of part (b) of Theorem 1, where we showed *constructively* how to find an optimal basic solution from any finite (near) optimal solution  $\mathbf{x} \in \mathcal{X}$ .

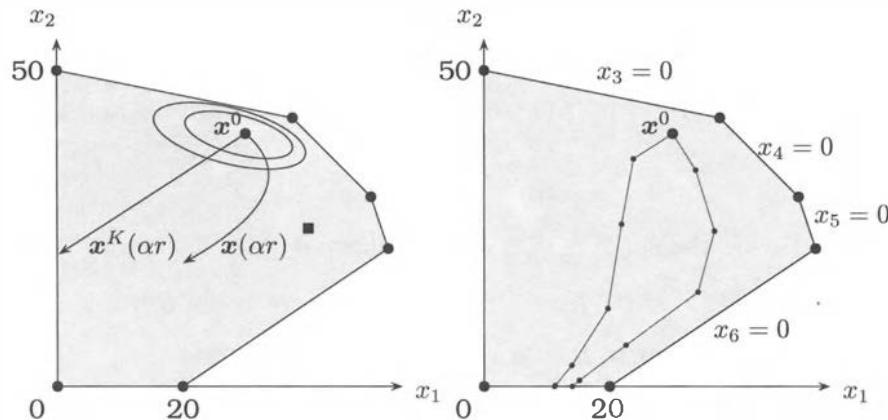
To show how to get started suppose that the *original* linear program that we wish to solve is in *canonical* form. Changing our notation, let the original linear program be  $\max\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ . From Remark 6.5 it follows that we can find an optimum solution or conclude that none exists by solving

$$\min\{-\mathbf{c}\mathbf{x} + \mathbf{b}^T \mathbf{u} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, -\mathbf{A}^T \mathbf{u} \leq -\mathbf{c}^T, \mathbf{u} \geq \mathbf{0}, \mathbf{x} \geq \mathbf{0}\},$$

where we have written the vector  $\mathbf{u}$  of dual variables as a column rather than as a row vector. By Remark 6.5 the optimum objective function value of this linear program equals zero – if it exists. Denote  $\mathbf{s} \in \mathbb{R}^m$  and  $\mathbf{v} \in \mathbb{R}^n$  the vectors of the respective slack variables for the linear inequalities. Choose any  $\hat{\mathbf{x}} > \mathbf{0}$ ,  $\hat{\mathbf{s}} > \mathbf{0}$ ,  $\hat{\mathbf{u}} > \mathbf{0}$  and  $\hat{\mathbf{v}} > \mathbf{0}$ . Then  $\mathbf{x} = \hat{\mathbf{x}}$ ,  $\mathbf{s} = \hat{\mathbf{s}}$ ,  $\mathbf{u} = \hat{\mathbf{u}}$ ,  $\mathbf{v} = \hat{\mathbf{v}}$  and  $\lambda = 1$  is a solution to

$$\begin{aligned} & \min -\mathbf{c}\mathbf{x} + \mathbf{b}^T \mathbf{u} + M\lambda \\ \text{subject to } & \mathbf{A}\mathbf{x} + \mathbf{s} + (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}} - \hat{\mathbf{s}})\lambda = \mathbf{b} \\ & \mathbf{A}^T \mathbf{u} - \mathbf{v} + (\mathbf{c}^T - \mathbf{A}^T \hat{\mathbf{u}} + \hat{\mathbf{v}})\lambda = \mathbf{c}^T \\ & \mathbf{x}, \mathbf{s}, \mathbf{u}, \lambda \geq \mathbf{0} \end{aligned}$$

that is in the relative interior of the feasible set of this (bigger) linear program as required by the basic algorithm.  $M$  must be a sufficiently large number to ensure that  $\lambda = 0$  in any optimal solution – provided that it exists. To ensure the boundedness of the feasible region we need to intersect it e.g. with a constraint that bounds the sum of  $\mathbf{x}$ ,  $\mathbf{s}$ ,  $\mathbf{u}$ ,  $\mathbf{v}$  and  $\lambda$  from above by a suitably large constant  $K$ . Adding this constraint to the linear program with a slack variable we get a problem that satisfies all of the assumptions that we have made to prove Remark 8.3. A different way of using the basic algorithm consists of utilizing the objective function as a constraint, i.e. by adding a constraint of the form  $\mathbf{c}\mathbf{x} \leq z$  where  $z$  is a parameter, and minimizing its slack; for more detail see the text.



**Fig. 8.3.** The line (8.9), the projective curve (8.17) and interior paths to optimality

## 8.2 Analysis, Algebra, Geometry

Μή μου τοὺς κύκλους πάρατε! <sup>3</sup>  
Archimedes of Syracuse (c. 287–212 B.C.)

Here the restriction  $(FLP_\rho)$  of the problem  $(FLP)$  is solved exactly. (So replace  $ALP_\rho$  by  $FLP_\rho$  in Figure 8.2.) We do so in two steps.

- First, the solution to  $(FLP_\rho)$  in the space of the variables of  $(LP)$  is derived for  $0 \leq \rho \leq r$ , see (8.1), which lends itself to direct interpretation in that space, see Figure 8.3.
- Then the problem is solved in the “transformed” space and the solution of  $(FLP_\rho)$  for values  $\rho \geq r$  is derived, which allows for a nice geometric interpretation of the solution.

This leads to an algorithm for  $(LP)$  with monotonically decreasing objective function values, without the restrictive assumptions of the basic algorithm and with a step complexity of  $O(p\sqrt{n})$ ; see Chapter 8.5. Here we are going to use the fact that every  $(n+1)$ -tuple  $(x_1, \dots, x_n, x_{n+1}) \in \mathbb{R}^{n+1}$  can also be interpreted as the *homogeneous coordinates* of some point in the  $n$ -dimensional projective space  $\mathcal{P}^n$ ; see the text.

### 8.2.1 Solution to the Problem in the Original Space

Reversing the projective transformation the nonlinear optimization problem  $(FLP_\rho)$  becomes the problem

$$(LP_\rho) \quad \min \{ \mathbf{c}\mathbf{x} : \mathbf{x} \in \mathcal{X} \cap T_0^{-1}(B_\rho^{n+1}) \}$$

and the first task is thus to find  $T_0^{-1}(B_\rho^{n+1})$ . Define a new parameter  $R = R(\rho)$  by

$$R = \rho \sqrt{(n+1)/n(r^2 - \rho^2)} . \quad (8.15)$$

<sup>3</sup>Don't touch my circles!

It follows that  $R \in [0, \infty)$  for all  $\rho \in [0, r)$  and since  $dR/d\rho = (n+1)(1-\rho^2/r^2)^{-3/2} > 0$ , the parameter change preserves strict monotonicity, i.e.  $\rho < \rho'$  if and only if  $R(\rho) < R(\rho')$ . We denote the pre-image  $T_0^{-1}(B_\rho^{n+1})$  of  $B_\rho^{n+1}$  by  $E(\mathbf{x}^0, R)$  and by  $(\text{LP}_R)$  the optimization problem  $(\text{LP}_\rho)$  in the parameter  $R$ , i.e.

$$(\text{LP}_R) \quad \min\{\mathbf{c}\mathbf{x} : \mathbf{x} \in \mathcal{X} \cap E(\mathbf{x}^0, R)\}.$$

**Remark 8.4** For  $0 \leq \rho < r$  the set  $T_0^{-1}(B_\rho^{n+1})$  is the ellipsoid

$$E(\mathbf{x}^0, R) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \mathbf{x}_C)^T \mathbf{H}(\mathbf{x} - \mathbf{x}_C) \leq R^2\}$$

where  $\mathbf{x}_C = (1+R^2)\mathbf{x}^0$  is its center,  $R = R(\rho)$  is defined in (8.15) and

$$\mathbf{H} = \mathbf{D}^{-1} (\mathbf{I}_n - (1+n+nR^2)^{-1}(1+R^2)\mathbf{e}\mathbf{e}^T) \mathbf{D}^{-1}.$$

Moreover, the inverse  $\mathbf{H}^{-1}$  of  $\mathbf{H}$  is given by

$$\mathbf{H}^{-1} = \mathbf{D} (\mathbf{I}_n + (1+R^2) \mathbf{e}\mathbf{e}^T) \mathbf{D}. \quad (8.16)$$

It follows that for  $0 \leq \rho < r$  the sets  $T_0^{-1}(B_\rho^{n+1})$  are ellipsoids  $E(\mathbf{x}^0, R)$  and satisfy

$$E(\mathbf{x}^0, R) \subset \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} > \mathbf{0}\} \text{ for all } 0 \leq R < \infty;$$

see Figures 8.1 and 8.3. For  $\rho \geq r$  the sets  $T_0^{-1}(B_\rho^{n+1})$  may be paraboloids, hyperboloids or they may simply not exist in *real* terms – which is why we work here with the parameter  $R$  rather than with  $\rho$ .

**Remark 8.5** If  $\mathbf{x}^0 > \mathbf{0}$  is a nonoptimal feasible solution to  $(\text{LP})$  then

$$\mathbf{x}(R) = \mathbf{x}^0 - (R/W)\mathbf{D} [\mathbf{p} - (1+\beta)^{-1}(z_0 - z(R) - \gamma)\mathbf{d}] \quad (8.17)$$

solves the optimization problem  $(\text{LP}_R)$  for all  $0 \leq R < \infty$  where

$$z(R) = z_0 - R((1+\beta)W - \gamma R)/(1+\beta + \beta R^2) \quad (8.18)$$

is the objective function value of  $(\text{LP}_R)$ ,

$$W = W(R) = (1+\beta)^{-1/2} \sqrt{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2} \quad (8.19)$$

and  $z_0, \beta, \gamma, \mathbf{p}, \mathbf{d}$  are defined in Chapter 8.1.1.  $z(R)$  is a strictly decreasing function of  $R$  since

$$\frac{dz}{dR} = -\frac{((1+\beta)W - \gamma R)^2}{W(1+\beta + \beta R^2)^2} < 0 \quad (8.20)$$

Different from the solution  $\mathbf{x}^K(\rho)$  obtained by approximation, which is a line in  $\mathbb{R}^n$ , the solution  $\mathbf{x}(R)$  obtained from  $(\text{FLP}_\rho)$  is a curve in  $\mathbb{R}^n$ . In the left part of Figure 8.3 we display both  $\mathbf{x}^K(\rho)$  and  $\mathbf{x}(R)$  when parameterized by  $\alpha$  for the data of Exercise 8.2 (ii) where  $\rho = \alpha r$ ,  $r = 1/\sqrt{n(n+1)}$  and we first expressed  $\mathbf{x}(R)$  in terms of  $\rho$  using (8.15) to get  $\mathbf{x}(\rho)$ .  $\mathbf{x}^0$  is the point in  $\mathbb{R}^2$  with coordinates  $x_1 = 30, x_2 = 40$  like in Exercise 8.2 (ii) and  $\mathbf{x}(\alpha r)$  is the truncation of  $\mathbf{x}(\rho) \in \mathbb{R}^6$  to  $\mathbb{R}^2$ . The point indicated by ■ has the coordinates  $x_1 = 40, x_2 = 25$ .  $x_3, \dots, x_6$  are the slack variables that correspond to the inequalities of our problem and define the respective line segments when they are equal to zero. Remember that  $r$  is the radius of the biggest ball  $B_\rho^{n+1}$  with center  $\mathbf{y}^0$  that is inscribable into the simplex  $S^{n+1}$  where  $n = 6$  in our example; see the text for more detail.

### 8.2.2 The Solution in the Transformed Space

Using the transformation  $T_0$  and expressing the result in terms of the parameter  $\rho$  rather than  $R$ , see (8.15), we calculate the solution  $\mathbf{y}(\rho)$  to  $(\text{FLP}_\rho)$  for all  $0 \leq \rho < r$  where  $r$  is defined in (8.1). We find after some algebra and simplifications that the solution to  $(\text{FLP}_\rho)$  for  $0 \leq \rho \leq r$  is

$$\mathbf{y}(\rho) = \mathbf{y}^0 - \rho \mathbf{q}(\rho) / \| \mathbf{q}(\rho) \| , \quad (8.21)$$

where  $\mathbf{y}^0$  is the center of  $S^{n+1}$  and in the previous notation we have set

$$\mathbf{q}(\rho) = \mathbf{Q} \begin{pmatrix} \mathbf{Dc}^T \\ -z(\rho) \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} + \frac{z_0 - z(\rho) - \gamma}{1 + \beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} - \frac{z_0 - z(\rho)}{n + 1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} , \quad (8.22)$$

$$z(\rho) = \frac{(\mathbf{cD}, \mathbf{0}) \mathbf{y}(\rho)}{y_{n+1}(\rho)} = z_0 - (n + 1)\rho \frac{(1 + \beta)W(\rho) - (n + 1)\gamma\rho}{1 + \beta - (n + 1)(n - \beta)\rho^2} , \quad (8.23)$$

$$W(\rho) = (1 + \beta)^{-1/2} \sqrt{(1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 - (\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2)(n + 1)\rho^2} . \quad (8.24)$$

Changing the parameter from  $R$  to  $\rho$  the quantity (8.19) becomes the quantity (8.24) multiplied by  $(1 - n(n + 1)\rho^2)^{-1/2}$  and likewise (8.18) transforms to (8.23). Thus they both exist for  $0 \leq \rho < r$ . The objective function (8.23) can also be written as

$$z(\rho) = z_0 - (n + 1)\rho \frac{(1 + \beta)\|\mathbf{p}\|^2 + \gamma^2}{(1 + \beta)W(\rho) + (n + 1)\gamma\rho} , \quad (8.25)$$

whereas the norm of  $\mathbf{q}(\rho)$  satisfies the relation

$$\|\mathbf{q}(\rho)\|^2 = \|\mathbf{p}\|^2 + \frac{(z_0 - z(\rho) - \gamma)^2}{(1 + \beta)} - \frac{(z_0 - z(\rho))^2}{(n + 1)} = \frac{(z_0 - z(\rho))^2}{(n + 1)^2\rho^2} \quad (8.26)$$

and thus,  $z(\rho) = z_0 - (n + 1)\rho\|\mathbf{q}(\rho)\|$ .

[As an aside, we can obtain the approximate solution (8.7) from the exact solution (8.21) by setting  $z(\rho)$  equal to zero in (8.22). More precisely, from (8.22) and the definition (8.6) of  $\mathbf{q}$

$$\mathbf{q}(\rho) = \mathbf{q} - z(\rho)\mathbf{r} , \text{ where } \mathbf{r} = \frac{1}{1 + \beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} - \frac{1}{n + 1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} \quad (8.27)$$

and  $\mathbf{r}$  is the orthoprojection of the  $(n + 1)^{\text{st}}$  unit vector of  $\mathbb{R}^{n+1}$  on the subspace (8.3). The “direction vectors” of (8.21) are the normalized orthogonal projections of  $(\mathbf{cD}, -z(\rho))$  on the subspace (8.3) that change as  $\rho$  varies and  $\mathbf{q}(\rho) = \mathbf{q}$  if and only if  $\beta = n$ , i.e.  $\mathbf{d} = \mathbf{0}$ . Asking the more general question under what conditions the curves (8.17) and (8.21) are straight lines you find that this is the case if and only if  $\mathbf{p}$  and  $\mathbf{d}$  are linearly dependent.]

To interpret the solution  $\mathbf{y}(\rho)$  to  $(\text{FLP}_\rho)$  geometrically, we proceed as follows. From formula (8.21) we get using (8.22), (8.25), (8.26) and by collecting the terms with  $\rho$  and  $\rho^2$ , respectively,

$$\mathbf{y}(\rho) = \mathbf{y}^0 + \frac{(1 + \beta)\rho W(\rho)}{(1 + \beta)\|\mathbf{p}\|^2 + \gamma^2} \mathbf{u} + \frac{(n + 1)\rho^2}{(1 + \beta)\|\mathbf{p}\|^2 + \gamma^2} \mathbf{v} , \quad (8.28)$$

where  $\mathbf{u}$  and  $\mathbf{v}$  and their respective norms are given by

$$\begin{aligned}\mathbf{u} &= -\begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} + \frac{\gamma}{1+\beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix}, \quad \mathbf{v} = -\frac{\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2}{n+1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} - \begin{pmatrix} \gamma\mathbf{p} - \|\mathbf{p}\|^2\mathbf{d} \\ 0 \end{pmatrix}, \\ \|\mathbf{u}\| &= \sqrt{\|\mathbf{p}\|^2 + \gamma^2/(1+\beta)}, \quad \|\mathbf{v}\| = \|\mathbf{u}\|\sqrt{(1+\beta)(\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2)/(n+1)}.\end{aligned}$$

Since  $\beta \geq 0$  and  $\|\mathbf{p}\| \neq 0$  we have  $\mathbf{u} \neq \mathbf{0}$ , while  $\mathbf{v} = \mathbf{0}$  if and only if  $\mathbf{p}$  and  $\mathbf{d}$  are linearly dependent. Moreover,  $\mathbf{u}^T \mathbf{v} = 0$ , i.e.,  $\mathbf{u}$  and  $\mathbf{v}$  are orthogonal to each other. Define two lines

$$\mathbf{u}(s) = \mathbf{y}^0 + s\mathbf{u}/\|\mathbf{u}\|, \quad \mathbf{v}(s) = \mathbf{y}^0 + s\mathbf{v}/\|\mathbf{v}\|, \quad (8.29)$$

where  $s \in \mathbb{R}$  and we make the tacit assumption that  $\|\mathbf{v}\| \neq 0$ . Both lines depend *only* on the center of  $S^{n+1}$  and the orthogonal projections  $\mathbf{p}$  and  $\mathbf{d}$  of  $D\mathbf{c}^T$  and  $\mathbf{e}$ , respectively, but *not* on the objective function value  $z_0$  of the starting point  $\mathbf{x}^0$ ; see Figure 8.4.

From (8.28) and (8.29) it follows that the solution  $\mathbf{y}(\rho)$  to  $(FLP_\rho)$  is a combination of the vectors  $\mathbf{u}(s)$  and  $\mathbf{v}(s)$  for some  $s = s(\rho) \in \mathbb{R}$ . Let us first look at the line  $\mathbf{v}(s)$ . From Exercise 8.5

$$v_{n+1}(s) = (n+1)^{-1} - s\sqrt{(\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2)/(n+1)((1+\beta)\|\mathbf{p}\|^2 + \gamma^2)} \geq 0$$

for all  $0 \leq s \leq r = 1/\sqrt{n(n+1)}$ . From the fact that  $v_{n+1}(s)$  decreases monotonically it follows that  $v_{n+1}(s_0) = 0$  for some  $s_0 \geq r$  and solving the equation we get

$$s_0 = \sqrt{((1+\beta)\|\mathbf{p}\|^2 + \gamma^2)/(n+1)(\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2)}.$$

$s_0$  exists if and only if  $\mathbf{p}$  and  $\mathbf{d}$  are linearly independent which is a temporary assumption that we make implicitly. Let  $z$  be *any* real number. Then we compute

$$(\mathbf{cD}, -z)\mathbf{v}(s) = \frac{z_0 - z}{n+1} + \frac{s}{\|\mathbf{v}\|} \left( -\frac{\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2}{n+1}(z_0 - z) \right) = (z_0 - z)(1 - s/s_0)/(n+1)$$

and thus  $(\mathbf{cD}, -z)\mathbf{v}(s_0) = 0$  for  $s = s_0$  no matter what value  $z$  assumes. It follows that,  $\mathbf{v}(s_0)$  is a “distinguished” point since *all hyperplanes*  $(\mathbf{cD}, -z)\mathbf{y} = 0$  obtained by varying  $z$  meet in this point. Reversing the projective transformation we find that  $(\mathbf{cD}, -z)\mathbf{y} = 0$  goes over into the hyperplane  $\mathbf{c}\mathbf{x} = z$  of  $\mathbb{R}^n$  which yields a set of *parallel* hyperplanes in  $\mathbb{R}^n$  if we vary the value of  $z$ . Thus after the projective transformation, these parallel hyperplanes of  $\mathbb{R}^n$  have the distinguished point  $\mathbf{v}(s_0)$  in common!

To resolve the mystery, remember that  $v_{n+1}(s_0) = 0$  as well and so  $\mathbf{v}(s_0)$  is an **improper** point of the  $n$ -dimensional projective space  $\mathcal{P}_n$  which corresponds to a “point at infinity” of  $\mathbb{R}^n$  if it corresponds to anything at all; see the text. Let  $\mathbf{w}^\infty = \mathbf{v}(s_0)$  and thus  $\mathbf{w}^\infty = \mathbf{y}^0 + \mathbf{w}$  where

$$\begin{aligned}\mathbf{w} &= \frac{-1}{\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2} \left\{ \frac{\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2}{n+1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} + \begin{pmatrix} \gamma\mathbf{p} - \|\mathbf{p}\|^2\mathbf{d} \\ 0 \end{pmatrix} \right\}, \\ \|\mathbf{w}\|^2 &= [(1+\beta)\|\mathbf{p}\|^2 + \gamma^2]/(n+1)(\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2).\end{aligned} \quad (8.30)$$

$\|\mathbf{w}^\infty\|^2 = \|\mathbf{w}\|^2 + 1/(n+1)$  and taking a sort of a limit, the point  $\mathbf{w}^\infty$  “slides off to infinity” as the vectors  $\mathbf{p}$  and  $\mathbf{d}$  “become linearly dependent” because  $\mathbf{w} = (\|\mathbf{p}\|^2\|\mathbf{d}\|^2 - \gamma^2)^{-1}\mathbf{v}$ . In this case, the line  $\mathbf{v}(s)$  degenerates into the point  $\mathbf{y}^0$  and  $\mathbf{y}(\rho)$  degenerates into the line  $\mathbf{u}(s)$ , see (8.32) below.

We write  $\|w\| = \infty$  to indicate the linear dependence of  $p$  and  $d$ . By the definition of equality of points in  $P_n$  linear dependence of  $p \neq 0 \neq d$  means *equality* of the two improper points  $(p, 0)$  and  $(d, 0)$  of  $P_n$ . Note that

$$z(\rho) = z_0 + (n+1)\mathbf{q}^T(\rho)\mathbf{w}, \quad (8.31)$$

yields another expression for the optimal objective function value of  $(FLP_\rho)$ . Rewriting formula (8.28) using  $w$  and formulas (8.24) and (8.30) we get for the solution  $\mathbf{y}(\rho)$  to  $(FLP_\rho)$  the expression

$$\mathbf{y}(\rho) = \mathbf{y}^0 + \mathbf{g}(\rho), \text{ where } \mathbf{g}(\rho) = \frac{\rho}{\|\mathbf{u}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{w}\|^2}} \mathbf{u} + \frac{\rho^2}{\|\mathbf{w}\|^2} \mathbf{w}. \quad (8.32)$$

Denoting by  $\phi(\rho)$  the *angle* between  $\mathbf{g}(\rho)$  and  $\mathbf{w}$  we get  $\cos \phi(\rho) = \rho / \|\mathbf{w}\|$  and from  $\sin^2 \alpha + \cos^2 \alpha = 1$

$$\mathbf{y}(\rho) = \mathbf{y}^0 + \rho(\sin \phi(\rho)\mathbf{u}/\|\mathbf{u}\| + \cos \phi(\rho)\mathbf{w}/\|\mathbf{w}\|).$$

### 8.2.3 Geometric Interpretations and Properties

In Figure 8.4 we give an illustration of the solution  $\mathbf{y}(\rho)$  to the problem  $(FLP_\rho)$  where  $\rho < \sigma$  and  $\mathbf{u}(s(\rho))$ ,  $\mathbf{u}(s(\sigma))$  denote the points of intersection of the line  $\mathbf{u}(s)$  with the hyperplanes  $(cD, -z(\rho))\mathbf{y} = 0$  and  $(cD, -z(\sigma))\mathbf{y} = 0$ , respectively.  $z(\rho) = (cD, 0)\mathbf{y}(\rho)/y_{n+1}(\rho)$  is the objective function value of  $\mathbf{y}(\rho)$  and  $z(\sigma)$  is defined likewise. The line  $\mathbf{u}(s)$  is the *tangent* and, by the orthogonality of  $\mathbf{u}$  and  $\mathbf{w}$ , the line  $\mathbf{v}(s)$  is the *normal* to the curve  $\mathbf{y}(\rho)$  in the point  $\mathbf{y}^0$ . Formula (8.32) is well defined for  $0 \leq \rho \leq \|\mathbf{w}\|$  and

$$\|\mathbf{y}(\rho) - (\mathbf{y}^0 + \mathbf{w}^\infty)/2\| = \|\mathbf{w}\|/2.$$

The loci of  $\mathbf{y}(\rho)$  form thus a semi-circle with center  $(\mathbf{y}^0 + \mathbf{w}^\infty)/2$  and radius  $\|\mathbf{w}\|/2$  as  $\rho$  varies from 0 to  $\|\mathbf{w}\|$ . The other half of the circle corresponds to the problem of *maximizing* the objective function of  $(FLP_\rho)$ ; see Exercise 8.6 (ii).

By the preceding we know that  $\mathbf{y}(\rho)$  is the optimal solution to  $(FLP_\rho)$  for all  $0 \leq \rho < r$  where  $r = 1/\sqrt{n(n+1)}$  is the radius of the largest ball  $B_\rho^{n+1}$  that can be *inscribed* into the simplex  $S^{n+1}$ .

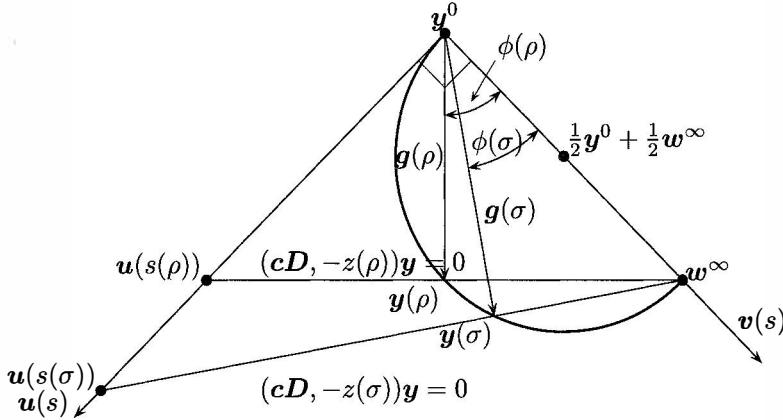
More precisely, starting from (8.21), ..., (8.26) we obtained formula (8.32) by algebraic manipulations and thus comparing (8.21) and (8.32)

$$\mathbf{g}(\rho) = -\rho\mathbf{q}(\rho)/\|\mathbf{q}(\rho)\|. \quad (8.33)$$

Thus (8.21) exists wherever  $\mathbf{g}(\rho)$  is well defined. Since  $\|d\|^2 \leq \|e\|^2 = n$  by the properties of orthogonal projections, it follows that  $\|\mathbf{w}\|^2 \geq r^2$  and equality holds if and only if  $d = e$  and  $\gamma = 0$ , which is the trivial case that we have encountered already when we defined  $\mathbf{w}^\infty$ . So in general we have  $\|\mathbf{w}\| > r$  and in the case that  $\|\mathbf{w}\| = \infty$  formula (8.32) becomes the line  $\mathbf{u}(s)$  defined in (8.29).

The question is whether or not  $\mathbf{y}(\rho)$ , as given by (8.32), remains an optimal solution to  $(FLP_\rho)$  for values of  $\rho$  greater than  $r$ . To answer it we need to know more about  $\mathbf{y}(\rho)$ . Since  $\mathbf{u}$  and  $\mathbf{w}$  determine  $\mathbf{y}(\rho)$  denote by  $L_{uw}$  the (two-dimensional) plane spanned by  $\mathbf{u}$  and  $\mathbf{w}$  and that contains  $\mathbf{y}^0$  and  $\mathbf{y}(\rho)$ , i.e.

$$L_{uw} = \{\mathbf{y} \in \mathbb{R}^{n+1} : \mathbf{y} = \mathbf{y}^0 + s\mathbf{u} + t\mathbf{w} \text{ for } s, t \in \mathbb{R}\} = \{\mathbf{y} \in \mathbb{R}^{n+1} : \mathbf{y} = \mathbf{y}^0 + sq + tr \text{ for } s, t \in \mathbb{R}\},$$



**Fig. 8.4.** The semi-circle determined by  $y(\rho)$

where  $q$  is defined in (8.6) and  $r$  in (8.27). The second equality follows because  $u$  and  $w$  are linearly independent if and only if  $q$  and  $r$  are linearly independent, where we write  $q$  and  $r$  as follows:

$$q = -\left(1 - \frac{z_0\gamma}{(1+\beta)\|u\|^2}\right)u - \frac{z_0}{n+1} \cdot \frac{w}{\|w\|^2}, \quad r = \frac{\gamma}{1+\beta} \cdot \frac{u}{\|u\|^2} - \frac{1}{n+1} \cdot \frac{w}{\|w\|^2}.$$

If  $\|w\| = \infty$ , i.e. if  $u$  and  $w$  as well as  $q$  and  $r$  are linearly dependent, then  $L_{uw}$  is simply the line  $u(s)$  and we drop  $w$  from the definition of  $L_{uw}$ . The last component of the line  $u(s)$  is

$$u_{n+1}(s) = \frac{1}{n+1} + \frac{\gamma}{1+\beta} \frac{s}{\|u\|}.$$

Thus for  $s = -(1+\beta)\|u\|/\gamma(n+1)$  we get an improper point of  $\mathcal{P}_n$  that we call  $u^\infty$ .  $u^\infty$  exists if and only if  $\gamma \neq 0$ , i.e., if  $p$  and  $d$  are *nonorthogonal*, which we assume temporarily. Write  $u^\infty = y^0 + u^0$  where

$$u^0 = -[(1+\beta)/\gamma(n+1)]u, \quad \|u^0\| = (1+\beta)\|u\|/|\gamma|(n+1). \quad (8.34)$$

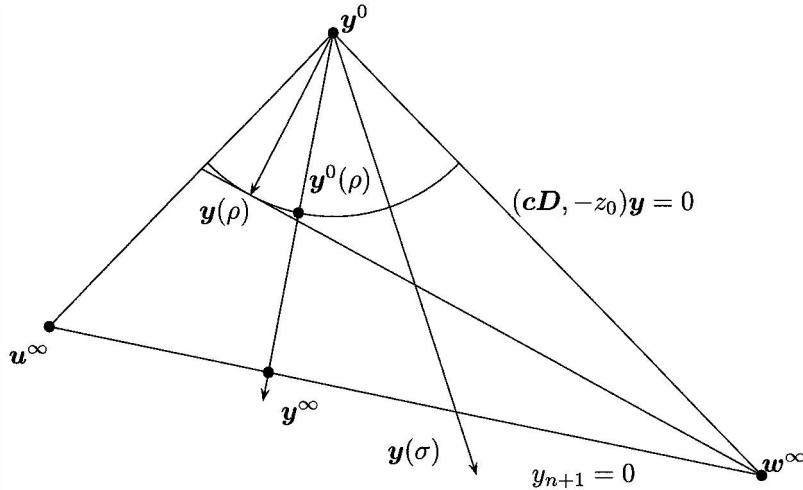
$\|u^\infty\|^2 = \|u^0\|^2 + 1/(n+1)$  and taking again a sort of limit, the point  $u^\infty$  “slides off to infinity” as the vectors  $p$  and  $d$  “become orthogonal”, but the line  $u(s)$  still exists. Like in the case of  $w^\infty$  we use  $\|u^\infty\| = \infty$  to indicate the orthogonality of  $p$  and  $d$ .

It follows that  $\|w^\infty\| = \infty = \|u^\infty\|$  if and only if  $d = 0$ . Unlike  $w^\infty$ , whose position is “fixed”,  $u^\infty$  lies on the halfline  $u(s)$  for  $s \geq 0$  if  $\gamma < 0$  while  $u^\infty$  lies on the halfline  $u(s)$  for  $s \leq 0$  if  $\gamma > 0$ .

Assuming  $\|u^\infty\| < \infty$  and  $\|w^\infty\| < \infty$  the three points  $y^0$ ,  $w^\infty$  and  $u^\infty$  determine a triangle in the plane  $L_{uw}$  to which  $y(\rho)$  belongs if  $\gamma < 0$  (see Figure 8.5). You are encouraged to supply the illustrations for the cases when  $\gamma = 0$  and  $\gamma > 0$ , respectively, yourself.

Denote the *perpendicular* from  $y^0$  on the line defined by  $u^\infty$  and  $w^\infty$ , i.e. the hypotenuse of the triangle, by  $y^\infty$  and thus

$$y^\infty = \mu u^\infty + (1-\mu)w^\infty \text{ for some } 0 < \mu < 1$$



**Fig. 8.5.** The triangle determined by  $y^0$ ,  $u^\infty$  and  $w^\infty$  if  $\gamma < 0$

since  $u$  and  $w$  form a right angle in the plane  $L_{uw}$ . From the condition of orthogonality we get the equation  $(y^0 - y^\infty)^T(u^\infty - w^\infty) = 0$ . Solving for  $\mu$  and simplifying we find

$$y^\infty = \frac{1}{\|d\|^2} \begin{pmatrix} d \\ 0 \end{pmatrix}, \quad \|y^0 - y^\infty\|^2 = (1 + \beta)/(n + 1)(n - \beta). \quad (8.35)$$

You verify that  $r^2 \leq \|y^0 - y^\infty\|^2 = \|d\|^{-2} - (n + 1)^{-1} \leq \|w\|^2$ . Equality holds in the first inequality if and only if  $d = e$ , i.e.  $\beta = 0$ , and in the second one if and only if  $\gamma = 0$  and  $\|w^\infty\| < \infty$ . So in general we have strict inequality on both sides. To see why we are interested in  $y^\infty$  consider the problem

$$\min\{y_{n+1} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}\}.$$

The solution exists for all  $\rho \geq 0$ . Denote it by  $y^0(\rho)$ . Using Remark 8.1 we calculate  $y^0(\rho) = y^0 - \rho r/\|r\|$  where  $r$  is defined in (8.27) and we get  $y^0(\rho) = y^0 + \rho(y^\infty - y^0)/\|y^\infty - y^0\|$  for all  $\rho \geq 0$ . Denote  $\rho_\infty = \|y^\infty - y^0\|$ . Hence we have e.g. from (8.35)

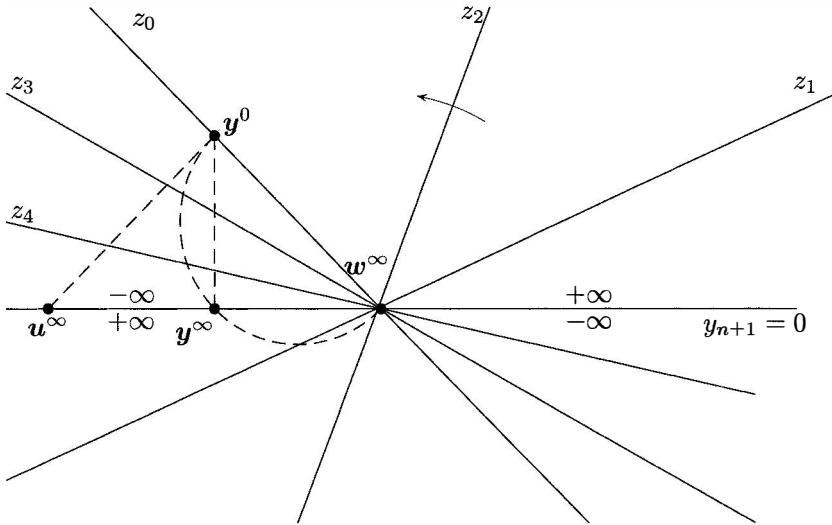
$$(n + 1)y_{n+1} \geq 1 - \rho/\rho_\infty > 0 \quad (8.36)$$

for all  $\mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}$  and  $0 \leq \rho < \rho_\infty$ . Moreover,  $y^\infty = y^0 - (n + 1)\rho_\infty r = y^0(\rho_\infty)$  and if  $\gamma < 0$  then we have  $y(\rho_\infty) = y^\infty$  as you verify using (8.32). In the case that  $\gamma = 0$  we get  $w^\infty = y^\infty$  and the triangle degenerates into a semi-open infinite rectangle.

If  $\gamma < 0$  and  $\rho > \rho_\infty$  then there exist  $\mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}$  such that  $y_{n+1} < 0$ . But “crossing the line” defined by the hypotenuse of the triangle corresponds to “passing through infinity” in  $\mathbb{R}^n$  and “coming back from infinity” which is what the *sign change* for  $y_{n+1}$  entails. Keeping in mind that we wish to apply the inverse  $T_0^{-1}$  of the projective transformation, it makes no sense to permit “solutions” to  $(FLP_\rho)$  having  $y_{n+1} < 0$ . We are thus led to consider the following restriction

$$(FLP_\rho^+) \quad \min \left\{ \frac{(cD, 0)\mathbf{y}}{y_{n+1}} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}, y_{n+1} > 0 \right\}$$

of  $(FLP_\rho)$  which is exactly  $(FLP_\rho)$  for all radii  $0 \leq \rho < \rho_\infty$  and thus in particular, for all radii  $0 \leq \rho < r$ .



**Fig. 8.6.** Lines in the plane  $L_{uw}$  if  $\gamma < 0$

#### 8.2.4 Extending the Exact Solution and Proofs

**Remark 8.6** (i) If  $x^0$  is a nonoptimal solution to (LP), then the vector  $y(\rho)$  given by (8.32) solves the problems  $(FLP_\rho)$  and  $(FLP_\rho^+)$  for all  $0 \leq \rho < \rho_\infty = \sqrt{\|d\|^{-2} - (n+1)^{-1}} \geq r$ . Moreover, if  $\gamma = p^T d \geq 0$  then the statement remains correct for all  $0 \leq \rho < \|w\|$ . If  $\gamma < 0$ , then a finite optimal solution to  $(FLP_\rho)$  and  $(FLP_\rho^+)$  does not exist for  $\rho_\infty \leq \rho < \|w\|$ .

(ii) If for some  $\rho > 0$  a finite optimal solution to  $(FLP_\rho)$  with an objective function value less than  $z_0 = cx^0$  exists, then the vector  $y(\rho)$  given by (8.32) solves the problems  $(FLP_\rho)$  and  $(FLP_\rho^+)$  and its corresponding objective function value  $z(\rho)$  satisfies  $z(\rho) < z_0$ . Moreover, for every  $\sigma \in (0, \rho)$  both problems have the same finite optimal solution with objective function value  $z(\sigma) > z(\rho)$ ,  $z(\sigma) < z_0$ .

From the analysis in this section of the text it follows that

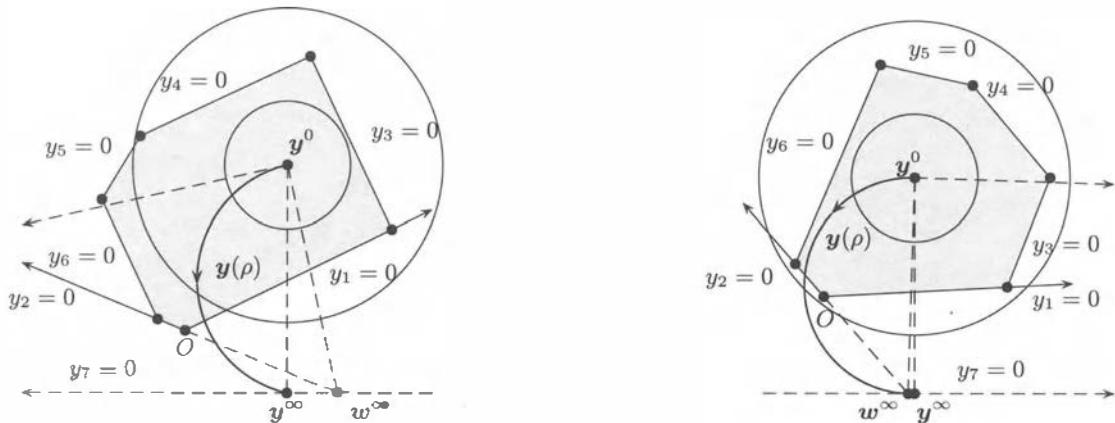
$$\begin{aligned} z(\rho) &= \frac{(cD, 0)y(\rho)}{y_{n+1}(\rho)} = z_0 - \frac{(n+1)\|u\|\rho}{\sqrt{1 - \frac{\rho^2}{\|w\|^2} + \frac{(n+1)\gamma\rho}{(1+\beta)\|u\|}}} \\ &= z_0 - (n+1)\rho \frac{(1+\beta)\|u\|\sqrt{1-\rho^2/\|w\|^2} - (n+1)\gamma\rho}{(1+\beta)(n-\beta)(n+1)\rho^2}. \end{aligned} \quad (8.37)$$

and moreover that the objective function can also be written as

$$z(\rho) = z_0 - (n+1)\rho\|q(\rho)\|. \quad (8.38)$$

The plane  $L_{uw}$  is divided into four parts by the two hyperplanes  $y_{n+1} = 0$  and  $(cD, -z_0)y = 0$ . The point  $w^\infty$  is the “origin” and every hyperplane  $(cD, -z)y = 0$  defines a line in  $L_{uw}$  that contains  $w^\infty$ .

Decreasing  $z$  we get a “bundle” of lines that turn counter-clockwise around the point  $w^\infty$ . In Figure 8.6 we illustrate the situation for  $\gamma < 0$ . It follows from the analysis of this section that approaching the line given by  $y_{n+1} = 0$  “from above” corresponds to  $z$  tending to  $-\infty$ , while



**Fig. 8.7.** Projective images of Figure 8.3 in the plane  $L_{uw}$  of  $\mathcal{P}^6$

"on the other side" of this line  $z$  is arbitrarily large. The values  $z_i$  shown in Figure 8.6 satisfy  $+\infty > z_1 > z_2 > z_0 > z_3 > z_4 > -\infty$ . Because  $(cD, -z)w^\infty = 0$  for all  $z$ , the objective function value  $z(\rho)$  of (8.37) for  $\rho = \|w\|$  does not exist if  $\gamma < 0$ :  $z(\|w\|)$  can be any real number.

The point  $u^\infty$  plays a role similar to that of  $w^\infty$  in our development except that  $u^\infty$  changes its position depending on the sign of  $\gamma = p^T d$ . Since  $(cD, -z)w^\infty = 0$  for all  $z \in \mathbb{R}$  one can ask oneself: what is the family of hyperplanes of the form  $(a^T, -z)y = 0$  where  $a \in \mathbb{R}^n$  is "fixed" and  $z \in \mathbb{R}$  arbitrary that are tangent to the  $n$ -dimensional ball  $B_n^{n+1}$  and that meet all in the point  $u^\infty$ ? See Exercise 8.6.

### 8.2.5 Examples of Projective Images

In Figure 8.7 we show projective images of the polytope of the problem of Exercise 8.2 (ii) for two different points  $x^0$  that are used in the transformation  $T_0$ : in the left part we use the point  $x^0$  of Figure 8.3 with coordinates  $x_1 = 30, x_2 = 40$ , in the right part the point with coordinates  $x_1 = 40, x_2 = 25$  for the transformation, i.e. the one indicated by ■ in Figure 8.3.

We have carried out all calculations in  $\mathbb{R}^7$  or  $\mathcal{P}^6$ , of course and the pictures are *not* an "artist's rendering" of some projective transformation. See the text for a detailed discussion of the implications of the analysis of the previous section for these examples.

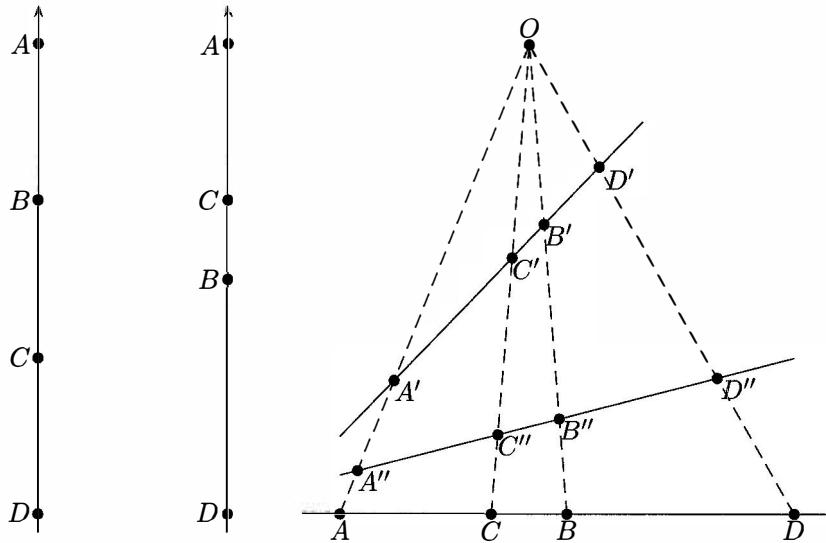
## 8.3 The Cross Ratio

Πᾶν μέτρον ἀπλοτόν! <sup>4</sup>  
Kleoboulos of Lindos (c. 550 B.C.)

To measure the progress of the objective function value along the curve  $y(\rho)$  or  $x(R)$  as defined by (8.32) and (8.17) we make initially the assumption that (LP) has a finite optimum. Denote by  $z_*$  the optimum objective function value.

The intersection of the hyperplane  $(cD, -z_*)y = 0$  with the plane  $L_{uw}$  is a line that contains  $w^\infty \in \mathcal{P}_n$ . In the  $n$ -dimensional projective space  $\mathcal{P}_n$  any two lines that belong to a plane either are

<sup>4</sup>Everything in good measure!



**Fig. 8.8.** The cross ratio of four points on a line

identical or have a nonempty intersection. For any  $\rho \in [0, \rho_\infty)$  the line

$$\mathbf{y}(\rho, \tau) = \mathbf{y}^0 + (\tau/\rho)(\mathbf{y}(\rho) - \mathbf{y}^0) \quad (8.39)$$

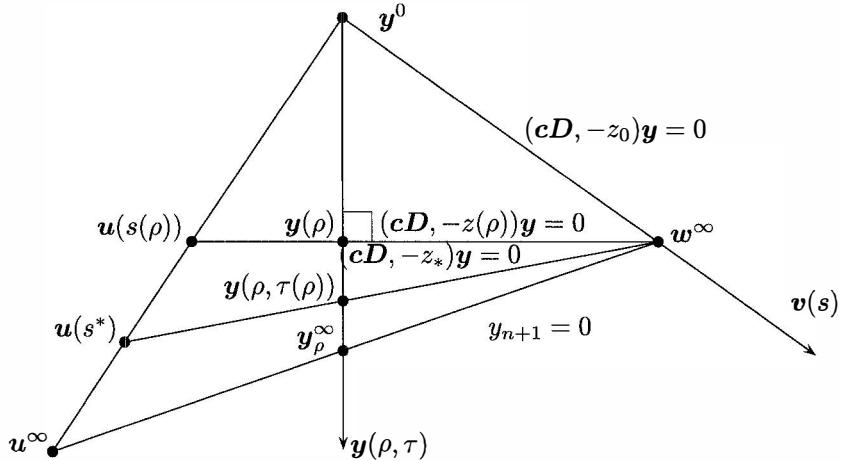
for  $\tau \in \mathbb{R}$  is different from the line  $L_{uw} \cap \{\mathbf{y} \in \mathbb{R}^{n+1} : (\mathbf{c}D, -z_s)\mathbf{y} = 0\}$  because  $\mathbf{y}^0$  belongs to  $\mathbf{y}(\rho, \tau)$ , but not to the second line since  $x^0$  is, by assumption, nonoptimal. Hence there exists  $\tau(\rho) \in \mathbb{R}$  where the two lines intersect. We now have three points  $\mathbf{y}^0$ ,  $\mathbf{y}(\rho)$ , and  $\mathbf{y}(\rho, \tau(\rho))$ , see Figure 8.9, and to measure “distance relations” in  $\mathcal{P}_n$  in a meaningful way we need an additional point.

The intersection of the hyperplane  $\{\mathbf{y} \in \mathbb{R}^{n+1} : y_{n+1} = 0\}$  with  $L_{uw}$  is the line that contains  $w^\infty$ ,  $y^\infty$  and  $u^\infty$ . This line is also different from the line  $\mathbf{y}(\rho, \tau)$  since  $y_{n+1}^0 = 1/(n+1) > 0$ . Denote by  $\mathbf{y}_\rho^\infty$  the point of intersection of  $\mathbf{y}(\rho, \tau)$  and the hyperplane  $y_{n+1} = 0$ .  $\mathbf{y}_\rho^\infty$  is, of course, in general different from the point  $\mathbf{y}^\infty$ ; see Figure 8.9 for an illustration.

We can use now the *cross ratio* of the resulting four points to measure the progress and, as we shall see, this yields the usual *relative error measure*. In elementary geometry the cross ratio of four points  $A, B, C, D$  that lie on a line is the double ratio

$$Dv(A, B; C, D) = \frac{AC}{BC} : \frac{AD}{BD},$$

where  $AC, BC, AD, BD$  are the “lengths” of the respective line segments with their respective signs according to the relative positioning of the four points; see Figure 8.8. The abbreviation  $Dv$  stands for the German *Doppelverhältnis* which translates to cross ratio and as our notation suggests, the first two points  $A$  and  $B$  are set in relation to the third and the fourth as follows: first we relate  $A$  and  $B$  to  $C$  by a ratio, we then relate  $A$  and  $B$  to  $D$  in the same fashion and finally, we form the double ratio. “Relative positioning” of the points on the line means that we choose an orientation for the line and give the length of, say, the line segment  $AC$  a negative sign if  $C$  occurs “before”  $A$  on the line. Thus the order of the points matters in the definition of  $Dv(A, B; C, D)$  which can be positive or negative.



**Fig. 8.9.** Cross ratios for the problem  $(FLP_\rho)$  if  $\gamma < 0$

It is a fundamental property of the cross ratio that it is *invariant* under central projections. In the right-hand side part of Figure 8.8 we show a projection with center  $O$  and by the invariance of the cross ratio we have  $Dv(A, B; C, D) = Dv(A', B'; C', D') = Dv(A'', B''; C'', D'')$ . This invariance has led to a concise concept of “distance” and thus to a *metric* in the projective space  $\mathcal{P}_n$  of  $n$  dimensions.

Given any four points  $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4$  of  $\mathcal{P}_n$  with  $\mathbf{y}^1 \neq \mathbf{y}^4$  and  $\mathbf{y}^2 \neq \mathbf{y}^3$  that lie on a line of  $\mathcal{P}_n$  and any two projective hyperplanes  $d_3\mathbf{y} = 0$  and  $d_4\mathbf{y} = 0$ , say, such that  $d_3\mathbf{y}^3 = d_4\mathbf{y}^4 = 0$  and  $d_3\mathbf{y}^2 \neq 0 \neq d_4\mathbf{y}^1$ , the cross ratio of the four points  $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4$  is

$$Dv(\mathbf{y}^1, \mathbf{y}^2; \mathbf{y}^3, \mathbf{y}^4) = \frac{(d_3\mathbf{y}^1)(d_4\mathbf{y}^2)}{(d_3\mathbf{y}^2)(d_4\mathbf{y}^1)}.$$

“Projective hyperplanes” means that like  $\mathbf{y}$  the  $d_i$  are *nonzero*  $(n+1)$ -tuples for  $i = 3, 4$ . Like in the elementary case, the cross ratio depends on the order in which we write the points, i.e.  $Dv(\mathbf{y}^2, \mathbf{y}^1; \mathbf{y}^3, \mathbf{y}^4) \neq Dv(\mathbf{y}^1, \mathbf{y}^2; \mathbf{y}^3, \mathbf{y}^4)$ , and the “relative positioning of the four points on the line” translates into signs of the quantities  $d_i\mathbf{y}^k$  used in the definition of  $Dv$ .

One verifies that our four points  $\mathbf{y}(\rho), \mathbf{y}^0, \mathbf{y}(\rho, \tau(\rho)), \mathbf{y}_\rho^\infty$ , see Figure 8.9, together with the hyperplanes  $(cD, -z_*)\mathbf{y} = 0$  and  $y_{n+1} = 0$ , respectively, satisfy the assumptions of the definition of  $Dv$ . Calculating the cross ratio in the stated order we find

$$Dv(\mathbf{y}(\rho), \mathbf{y}^0; \mathbf{y}(\rho, \tau(\rho)), \mathbf{y}_\rho^\infty) = \frac{(cD, -z_* + z(\rho) - z(\rho))\mathbf{y}(\rho)}{(cD, -z_* + z_0 - z_0)\mathbf{y}^0} \cdot \frac{(n+1)^{-1}}{y_{n+1}(\rho)} = \frac{z(\rho) - z_*}{z_0 - z_*},$$

which is indeed the relative error. To use the cross ratio in our estimation, we need a second way of calculating it. We do this first for the general case of points  $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4$  of  $\mathcal{P}_n$  used in the definition. Since  $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4$  lie on a line we find  $\mu_1, \mu_2 \in \mathbb{R}$  and  $\lambda_1, \lambda_2 \in \mathbb{R}$  such that  $\mathbf{y}^3 = \mu_1\mathbf{y}^1 + \mu_2\mathbf{y}^2$  and  $\mathbf{y}^4 = \lambda_1\mathbf{y}^1 + \lambda_2\mathbf{y}^2$ . Consequently, from  $d_3\mathbf{y}^3 = 0$  we get  $\mu_1 d_3\mathbf{y}^1 + \mu_2 d_3\mathbf{y}^2 = 0$  and likewise,  $\lambda_1 d_4\mathbf{y}^1 + \lambda_2 d_4\mathbf{y}^2 = 0$ . Now  $\mathbf{y}^1 \neq \mathbf{y}^4, \mathbf{y}^2 \neq \mathbf{y}^3$  imply  $\lambda_2 \neq 0$  and  $\mu_1 \neq 0$  and thus  $d_3\mathbf{y}^1/d_3\mathbf{y}^2 = -\mu_2/\mu_1$  and  $d_4\mathbf{y}^2/d_4\mathbf{y}^1 = -\lambda_1/\lambda_2$ . Hence

$$Dv(\mathbf{y}^1, \mathbf{y}^2; \mathbf{y}^3, \mathbf{y}^4) = \lambda_1 \mu_2 / \lambda_2 \mu_1.$$

To apply this formula for the cross ratio to our situation denote by  $\tau_\rho^\infty$  the value of  $\tau$  such that  $\mathbf{y}(\rho, \tau) = \mathbf{y}_\rho^\infty$ . From the definitions of  $\tau(\rho)$  and of  $\tau_\rho^\infty$  it now follows that

$$Dv(\mathbf{y}(\rho), \mathbf{y}^0; \mathbf{y}(\rho, \tau(\rho)), \mathbf{y}_\rho^\infty) = \frac{z(\rho) - z_*}{z_0 - z_*} = \frac{1 - \rho/\tau(\rho)}{1 - \rho/\tau_\rho^\infty}. \quad (8.40)$$

The question is whether or not one can estimate the last expression appropriately. In Figure 8.9 we illustrate the case where  $\gamma = \mathbf{p}^T \mathbf{d} < 0$ . To get a different way of estimating the progress of the objective function value of (LP) we can also use e.g. four points on the line  $\mathbf{u}(s)$ . In Figure 8.9 we indicate four such points and we leave it as an exercise to verify that  $Dv(\mathbf{u}(s(\rho)), \mathbf{y}^0; \mathbf{u}(s^*), \mathbf{u}^\infty) = (z(\rho) - z_*)/(z_0 - z_*)$ .

Remember from Chapter 8.2 that  $(c\mathbf{D}, -z)\mathbf{w}^\infty = 0$  for all  $z \in \mathbb{R}$ . The line  $\mathbf{y}(\rho, \tau)$  intersects the hyperplane  $(c\mathbf{D}, -z)\mathbf{y} = 0$  somewhere in the plane  $L_{uw}$  and the point of intersection lies on the line segment between  $\mathbf{y}^0$  and  $\mathbf{y}_\rho^\infty$  for all  $z_0 \geq z > -\infty$  if  $\gamma < 0$ . Indeed, taking the limit  $z \rightarrow -\infty$  the hyperplane  $(c\mathbf{D}, -z)\mathbf{y} = 0$  becomes the hyperplane  $y_{n+1} = 0$  since  $(c\mathbf{D}, -z)\mathbf{y} = 0 = ((1/z)c\mathbf{D}, -1)\mathbf{y}$  for all  $z \neq 0$  and one reasons likewise when  $\gamma \geq 0$ ; see also Figure 8.6. In case that (LP) has an unbounded optimum the points  $\mathbf{y}(\rho, \tau(\rho))$  and  $\mathbf{y}_\rho^\infty$  thus coincide, but the cross ratio (8.40) is still defined as we have not assumed that  $\mathbf{y}^3 \neq \mathbf{y}^4$  in the definition of it. So we can always work with some finite lower bound – *fictive or real* – on the optimum objective function value of (LP) when we use the cross ratio.

## 8.4 Reflection on a Circle and Sandwiching

Γνῶθι σ' αὐτόν! <sup>5</sup>  
Thales of Miletos (643-548 B.C.)

The analysis of the problem  $(FLP_\rho)$  settled the existence and uniqueness of its solution and tells us how to move from a given solution  $\mathbf{x}^0$  to a new solution  $\mathbf{x}^1$ , say, such that  $z_1 = c\mathbf{x}^1 < c\mathbf{x}^0 = z_0$  if  $\mathbf{x}^0$  is nonoptimal. The cross ratio tells us how to measure the progress that we make towards the solution of (LP) by solving the problem  $(FLP_\rho)$  for some  $\rho \geq 0$ . In what follows, we construct, starting from suitable *initial* upper and lower bounds, a sequence of lower bounds  $v^1, v^2, \dots$  concurrently with the upper bounds  $z_1, z_2, \dots$  that give a faster polynomial step complexity of a very different realization of the basic algorithmic idea.

A technique – which dates to antiquity – to perform geometrical constructions is the *inversion of a point in a circle*, or the reflection on a circle for short, and it goes as follows.

Given a circle with center  $O$  and radius  $\rho$ , say, take any point  $P \neq O$  in the circle and construct a point  $P'$  on the line determined by  $O$  and  $P$  by requiring that  $OP \cdot OP' = \rho^2$  where  $OP$  and  $OP'$  are the Euclidean lengths of the corresponding line segments. To every point inside the circle there corresponds exactly one point outside of it and the reverse holds as well. The mapping is thus bi-unique and to complete it, let us think of the center as being mapped into some (unique) “point at infinity”. By the prescription

$$\mathbf{y}^{\text{inv}} = \mathbf{y}^0 + \frac{\rho^2}{\|\mathbf{y} - \mathbf{y}^0\|^2} (\mathbf{y} - \mathbf{y}^0)$$

---

<sup>5</sup>Know thyself!

we define for every point  $\mathbf{y} \neq \mathbf{y}^0$  inside (outside) of  $B_\rho^{n+1}$  its “inverse” point  $\mathbf{y}^{\text{inv}}$  outside (inside) of  $B_\rho^{n+1}$ . It follows that for  $\rho = 1/\sqrt{n+1}$  the set  $B_r^{n+1}$  is mapped into the set

$$\left\{ \mathbf{y} \in \mathbb{R}^{n+1} : \sum_{j=1}^{n+1} y_j = 1, \sum_{j=1}^{n+1} (y_j - \frac{1}{n+1})^2 \geq \frac{n}{n+1} \right\}$$

and vice versa. Remember that  $r = 1/\sqrt{n(n+1)}$  is the largest radius of a ball that can be inscribed into  $S^{n+1}$ , whereas  $\sqrt{n/(n+1)}$  is the smallest radius of a ball that can be circumscribed to  $S^{n+1}$ ; see Exercise 8.1 and the left part of Figure 8.10 below where  $\rho = (n+1)^{-1/2}$ ,  $R = \sqrt{n/(n+1)}$  and reflection on the circle with radius  $\rho$  is illustrated for  $n = 2$ .

This “inversion” or reflection interchanges the subset of  $\mathbb{R}^{n+1}$  where positivity of the solution  $\mathbf{y}(\rho)$  to  $(\text{FLP}_\rho)$  can be guaranteed with an “outside” where the positivity of  $\mathbf{y}(\rho)$  is certain to be lost on some component of  $\mathbf{y}(\rho)$ .

For every radius  $\rho$  with  $0 < \rho < r$  we get by reflection on the boundary of the ball with radius  $(n+1)^{-1/2}$  a “twin” problem  $(\text{FLP}_\sigma)$  to the problem  $(\text{FLP}_\rho)$  where  $\sigma = 1/(n+1)\rho > \sqrt{n/(n+1)}$ .  $(\text{FLP}_\sigma)$  can be thought of as a *relaxation* of the problem  $(\text{FLP})$  and thus of the original  $(\text{LP})$  – provided that the solution to  $(\text{FLP}_\sigma)$  exists. But if  $\rho$  is not “too small”, then  $\sigma$  is not “too big” and – as Remark 8.6 shows – a solution to  $(\text{FLP}_\sigma)$  may very well exist in this case.

**Remark 8.7** Let  $\mathbf{x}^0 \in \mathcal{X}$  satisfying  $\mathbf{x}^0 > 0$  and  $z_0 = \mathbf{c}\mathbf{x}^0$  be a nonoptimal solution to  $(\text{LP})$  such that for some  $\sigma > \sqrt{n/(n+1)}$  the optimal solution  $\mathbf{y}(\sigma)$  to the problem  $(\text{FLP}_\sigma)$  exists and denote its objective function value by  $z(\sigma)$ . Then  $z(\sigma) < z_0$  and there exists an “interior point” solution to the dual of the linear program  $(\text{LP}^*)$ , i.e., to the program  $\max\{\mu_0 : \boldsymbol{\mu}^T \mathbf{A} \leq \mathbf{c}, -\boldsymbol{\mu}^T \mathbf{b} + \mu_0 \leq 0\}$ , with an objective function value equal to  $z(\sigma)$ . Thus  $z(\sigma)$  is a lower bound on the optimal objective function value of  $(\text{LP})$ .

Assuming that  $(\text{FLP}_\sigma)$  is solvable, it follows that by inverting  $\mathbf{y}(\sigma)$  on the ball with radius  $1/\sqrt{n+1}$  we get a point in  $T_0(\mathcal{X}) \cap B_\rho^{n+1}$ , i.e., we have for  $\rho = 1/(n+1)\sigma$

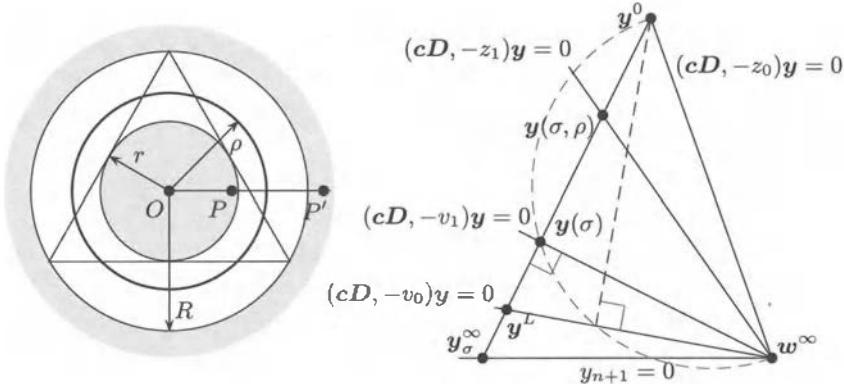
$$\mathbf{y}(\sigma, \rho) = \mathbf{y}^0 + (\rho/\sigma)\mathbf{g}(\sigma) \in T_0(\mathcal{X}) \cap B_\rho^{n+1} \quad \text{and} \quad (8.41)$$

$$\mathbf{y}^-(\sigma, \rho) = \mathbf{y}^0 - (\rho/\sigma)\mathbf{g}(\sigma) \in T_0(\mathcal{X}) \cap B_\rho^{n+1}, \quad (8.42)$$

which is used in the proof of Remark 8.7. The next remark shows that the inversion on the ball with radius  $1/\sqrt{n+1}$  gives a sharper estimation of the relative error than the one we got for the basic algorithm. It spells out most of the assumptions on the radii  $\rho$  and  $\sigma$  that are needed for an iterative application.

**Remark 8.8** Let  $\mathbf{x}^0 \in \mathcal{X}$ ,  $\mathbf{x}^0 > 0$  and  $z_0 = \mathbf{c}\mathbf{x}^0$ , be such that for some  $\omega \geq r$  the problem  $(\text{FLP}_\omega)$  has an optimal solution with objective function value  $z(\omega) = v_0 < z_0$ . Then for any  $\rho$  and  $\sigma$  satisfying  $0 < \rho < r < \sigma < \omega$  and  $\rho\sigma(n+1) = 1$  the following statements are correct: (i) The optimal objective function value  $v_1 = z(\sigma)$  of  $(\text{FLP}_\sigma)$  is a lower bound for  $(\text{LP})$ ,  $v_1 > v_0$  and thus  $v_0$  is a lower bound for  $(\text{LP})$  as well. (ii)  $\mathbf{x}^1 = T_0^{-1}(\mathbf{y}(\sigma, \rho))$  satisfies  $\mathbf{x}^1 > 0$ ,  $\mathbf{x}^1 \in \mathcal{X}$  and  $z_1 = \mathbf{c}\mathbf{x}^1 < z_0$  where  $\mathbf{y}(\sigma, \rho)$  is given by (8.41). (iii) If in addition to the above  $(n+1)\rho \leq 1$  and  $\sigma^2 \leq \omega$ , then setting  $\alpha = (n+1)\rho$  it follows that

$$\frac{z_1 - v_1}{z_0 - v_0} \leq \left(1 + \frac{\alpha(1-\alpha)}{\sqrt{n+1}}\right)^{-1}. \quad (8.46)$$



**Fig. 8.10.** Reflection on a circle and cross ratios for sandwiching if  $\gamma < 0$

The proof of Remark 8.8 utilizes *inter alia* the fact that  $z_1$  can be evaluated as follows:

$$z_1 = \frac{(cD, -z(\sigma) + z(\sigma))y(\sigma, \rho)}{y_{n+1}(\sigma, \rho)} = z(\sigma) + \frac{1 - \rho/\sigma}{(n+1)y_{n+1}(\sigma, \rho)}(z_0 - z(\sigma)). \quad (8.47)$$

#### 8.4.1 The Iterative Step

To apply Remark 8.8 iteratively we have to prove e.g. that the initially selected radii  $\rho$  and  $\sigma$  continue to satisfy the various assumptions of Remark 8.8 for some suitable radius  $\omega$  after we have moved to a new point. If radii  $\rho$  and  $\sigma$  that do not depend upon the “current” solution to (LP) exist, then we get new upper and lower bounds such that (8.46) remains correct.

Like in Remark 8.8 let  $x^1$  be the point obtained from (8.41) under the projective transformation,  $z_1 = cx^1$  its objective function value,  $v_1 = z(\sigma) < z_1$  be the current lower bound and  $D_1 = \text{diag}(x_1^1, \dots, x_n^1)$  the diagonal matrix of the “next” transformation  $T_1$  that maps  $x^1$  into the center of  $S^{n+1}$ . Denote by  $q^1(v_1)$  the orthogonal projection of  $(cD_1, -v_1)^T$  on the subspace (8.3) with  $D$  replaced by  $D_1$ .

**Claim 1** If  $x^1$  is a nonoptimal solution to (LP), then  $q^1(v_1) \neq 0$ .

If  $x^1$  is nonoptimal solution for (LP), then by Claim 1 the line  $\hat{y}(\tau) = y^0 - \tau q^1(v_1)/\|q^1(v_1)\|$  is well defined and intersects the hyperplane  $(cD_1, -v_1)y = 0$  for  $\tau = \omega_1$ , say, where

$$\omega_1 = (z_1 - v_1)/(n+1)\|q^1(v_1)\| > 0 \quad (8.48)$$

since  $z_1 - v_1 > 0$ . Thus we have a radius  $\omega_1$  corresponding to the radius  $\omega$  of Remark 8.8. Denote by  $(\text{FLP}_{\omega_1}^1)$  the program that we get under  $T_1$  using  $D_1$  rather than  $D$ . Like in Chapter 8.2 we get a two-dimensional plane  $L_{\hat{u}\hat{w}}$ , where  $\hat{u}$  corresponds to  $u$  and  $\hat{w}$  to  $w$  of Chapter 8.2. The line  $\hat{y}(\tau)$  satisfies  $\hat{y}(\tau) \in L_{\hat{u}\hat{w}}$  for all  $\tau \geq 0$  and corresponds to the “broken” line of Figure 8.10.

**Claim 2** If  $q^1(v_1) \neq 0$ , then  $(\text{FLP}_{\omega_1}^1)$  has a finite optimal solution.

It follows from Claim 2 that the problem  $(\text{FLP}_{\rho}^1)$  has an optimal solution for  $\rho = \omega_1$  with an objective function value of  $z^1(\omega_1)$ , say, and unless  $x^1$  is an optimal solution to (LP) then necessarily that  $z^1(\omega_1) < z_1$  as required for Remark 8.8. Note that  $z^1(\omega_1) = v_1$  if  $\hat{\gamma} \leq 0$ , but it is possible

that  $z^1(\omega_1) > v_1$  if  $\hat{\gamma} > 0$ . This latter possibility does, however, not change the validity of the argument used to prove Remark 8.8 since we conclude that  $z^1(\omega_1)$  is a lower bound for (LP) if the assumptions of Remark 8.8 are met. The relative error estimation remains correct in this case, too.

So if  $q^1(v_1) \neq 0$ , then the nonlinear program  $(FLP_{\omega_1}^1)$  that we get at the new point  $x^1 \in \mathcal{X}$  has a finite optimal solution. To prove that, for a suitable initial choice, the “original” radii  $\rho$  and  $\sigma$  of Remark 8.8 can be applied again we need to estimate the length of the vector  $q^1(v_1)$ .

**Claim 3** If  $0 < (n+1)\rho < 1/\sqrt{2}$ ,  $\sigma\rho(n+1) = 1$  and  $\sigma^2 \leq \omega$ , then  $\|q^1(v_1)\| < (z_1 - v_1)(n+1)\rho^2$  for all  $0 < (n+1)\rho < \varepsilon_n$ , where  $\varepsilon_n = \sqrt{(n+1)(1 - \sqrt{n/(n+1)})}$ . Moreover,  $\varepsilon_n \geq \frac{1}{\sqrt{2}}$  for all  $n \geq 1$ ,  $\lim_{n \rightarrow \infty} \varepsilon_n = \frac{1}{\sqrt{2}}$  and  $\sqrt{2} < \sigma < \sigma^2 \leq \omega_1$ .

Since  $\varepsilon_n < 1$  for all  $n \geq 1$ , this estimation of  $\|q^1(v_1)\|$  does not apply to all  $\rho$  satisfying  $0 < \rho(n+1) < 1$ , but it applies to all  $\rho$  satisfying  $0 < \rho(n+1) < 1/\sqrt{2} = 0.707\dots$ . Consequently, if we can find an *initial* radius  $\rho$  such that  $0 < (n+1)\rho < 1/\sqrt{2}$  then Remark 8.8 applies *mutatis mutandis* to the new point  $x^1$ , the radius  $\omega_1$ , the same  $\rho$  and  $\sigma$  as used before and all  $n \geq 1$ . The iterative application of Remark 8.8 is thus correct for any  $\rho$  in the stated bounds.

## 8.5 A Projective Algorithm

In fine initium.<sup>6</sup>  
Latin proverb

Given a nonoptimal interior point  $x^0 \in \mathcal{X}$  with objective function value  $z_0$  and an initial lower bound  $v_0$  on the optimal value of the objective function value of (LP) we are now ready to formulate a projective algorithm with input parameters  $\alpha$  for the step-size,  $p$  for the desired precision in terms of the relative error and the descriptive data for (LP).

**Projective Algorithm** ( $\alpha, p, m, n, A, c, x^0, z_0, v_0$ )

**Step 0:** Set  $D_0 := \text{diag}(x_1^0, \dots, x_n^0)$ ,  $z := z_0$  and  $k := 0$ .

**Step 1:** Compute  $G := AD_k^2A^T$ ,  $G^{-1}$  and  $P := I_n - D_k A^T G^{-1} A D_k$ .

**Step 2:** Compute  $p := PD_k c^T$ ,  $d := Pe$ ,  $\gamma := p^T d$ ,  $\beta := n - \|d\|^2$ ,  $\Lambda := (1 + \beta)\|p\|^2 + \gamma^2$ ,

$$K := (n+1)(\|p\|^2\|d\|^2 - \gamma^2), \quad v := z - \frac{\sqrt{(1+\beta)(\Lambda\alpha^2 - K)} - (n+1)\gamma}{(n+1)^{-1}(1+\beta)\alpha^2 - n + \beta} \quad \text{and}$$

$$t := (n+1)(1+\beta)[\gamma(n+1) + (z-v)(1+2\beta-n)]^{-1}.$$

**Step 3:** Set  $x^{k+1} := x^k - tD_k \left( p - \frac{z-v-\gamma}{1+\beta} d \right)$  and  $D_{k+1} := \text{diag}(x_1^{k+1}, \dots, x_n^{k+1})$ .

**Step 4:** if  $\frac{cx^{k+1} - v}{z_0 - v_0} < 2^{-p}$  stop “ $x^{k+1}$  is a  $p$ -optimal solution to (LP)”.

Set  $z := cx^{k+1}$ ; replace  $k+1$  by  $k$ ; go to Step 1.

To prove convergence of the projective algorithm we need, of course, an *initial* interior point  $x^0 \in \mathcal{X}$  and a lower bound  $v_0$  such that the various assumptions of Remark 8.8 are satisfied. Denote by  $q(v_0)$  the orthoprojection of  $(cD, -v_0)^T$  on the subspace (8.3) where

$$D = \text{diag}(x_1^0, \dots, x_n^0)$$

---

<sup>6</sup>In the end there is a beginning.

is given by  $x^0$ . Then like in Chapter 8.4, see (8.48), we conclude that  $q(v_0) \neq 0$ , set

$$\omega_0 = (z_0 - v_0)/(n + 1)\|q(v_0)\|$$

and prove that  $(FLP_{\omega_0})$  has a finite optimal solution. For the iterative application of Remark 8.8 a step-size  $\alpha = (n + 1)\rho$ , such that  $\sigma^2 < \omega_0$  where  $\sigma = 1/\rho(n + 1) = 1/\alpha$ , is required and thus we need

$$\alpha > \alpha_0 = \sqrt{(n + 1)\|q(v_0)\|/(z_0 - v_0)} . \quad (8.50)$$

From the analysis of Chapter 8.4 we have an upper bound of  $1/\sqrt{2}$  on the step-size  $\alpha$ . Thus for any pair of  $z_0$  and  $v_0$  such that  $\alpha_0\sqrt{2} < 1$  we get a nonempty interval for the step-size that gets us started. Since the projection  $q(v_0)$  depends upon  $x^0 \in \mathcal{X}$ , of course, not every interior point works.

**Remark 8.9** (Correctness and finiteness) *For any step length  $\alpha$  satisfying  $\alpha_0 < \alpha < 1/\sqrt{2}$  the projective algorithm iterates at most  $\mathcal{O}(p\sqrt{n+1})$  times, where  $\alpha_0$  is defined in (8.50) with respect to a suitable interior point  $x^0 \in \mathcal{X}$  and initial upper and lower bounds  $z_0$  and  $v_0$  for (LP), respectively.*

Let us now briefly discuss how to start the projective algorithm for a general linear program (LP) for which an interior point  $x^0 \in \mathcal{X}$  is known. We need to ensure that  $\alpha_0\sqrt{2} < 1$ . Consider  $q(v_0)$  and write  $q(v_0) = q - v_0 r$  where  $q$  is defined in (8.6) and  $r$  in (8.27). If  $r = 0$ , then  $\|q(v_0)\| = \|q\|$  is independent of the numerical value of  $v_0$  and thus by choosing any finite  $v_0 > -\infty$  that is “small enough” we can make  $\alpha_0 \geq 0$  as small as we wish.

By (8.27)  $r = 0$  if and only if  $d = 0$  in which case the plane  $L_{uw}$  of Chapter 8.2 degenerates into a line and  $\|w^\infty\| = \infty$  since  $p$  and  $d$  are linearly dependent. Thus any interior point  $x^0 \in \mathcal{X}$  with the property that the orthoprojection (8.4) of  $e$  on the subspace (8.5) equals zero works.

To see how we can always “force” this to happen initially let  $K > 0$  be any integer such that every basic feasible solution to (LP) satisfies  $\sum_{j=1}^n x_j < K$  with strict inequality and denote by  $x_{n+1}$  the corresponding slack variable. It is shown in the text that  $x^0 = \kappa e$ ,  $x_{n+1}^0 = x_{n+2}^0 = \kappa$  is a feasible starting point where  $\kappa = K/(n + 2)$ . Clearing fractions we get the linear program

$$(LP') \quad \begin{aligned} \min \quad & \mathbf{c}x + 0x_{n+1} + Mx_{n+2} \\ \text{subject to} \quad & KAx + \mathbf{0}x_{n+1} + \hat{\mathbf{b}}x_{n+2} = K\mathbf{b} \\ & \sum_{j=1}^n x_j + x_{n+1} + x_{n+2} = K \\ & x, x_{n+1}, x_{n+2} \geq \mathbf{0}, \end{aligned}$$

in  $n + 2$  variables where  $\hat{\mathbf{b}} = (n + 2)\mathbf{b} - KAe$ .  $(LP')$  has integer data and its digital size remains polynomial in the size  $L$  of the original linear program (LP).

## 8.6 Centers, Barriers, Newton Steps

Δός μοι ποῦ στῶ καὶ τὴν γῆν κινήσω! <sup>7</sup>  
Archimedes of Syracuse (c. 287-212 B.C.)

We assume throughout this section that the feasible region  $\mathcal{X} \subseteq \mathbb{R}_+^n$  is a polytope with a nonempty relative interior. Let  $\mathbf{x}^1, \dots, \mathbf{x}^p$  be the vertices of  $\mathcal{X}$ . The **barycenter** or the *center of gravity* of  $\mathcal{X}$  is the point

$$\mathbf{x}^G = \frac{1}{p} \sum_{i=1}^p \mathbf{x}^i, \quad (8.51)$$

which is obtained as the convex combination of all vertices with equal weights and thus  $\mathbf{x}^G \in \text{relint } \mathcal{X}$ .

A subset  $S \subseteq \mathbb{R}^n$  is **centrally symmetric**, if there exists  $\mathbf{x}^0 \in S$  such that for any  $\mathbf{y} \in \mathbb{R}^n$  with  $\mathbf{x}^0 + \mathbf{y} \in S$  we also have  $\mathbf{x}^0 - \mathbf{y} \in S$ . Ellipsoids and balls in  $\mathbb{R}^n$  are examples of such sets.

For compact convex sets  $S \subseteq \mathbb{R}^n$  of full dimension that are not centrally symmetric the notion of a “centroid” is used to define a “center” of  $S$ .  $S$  has a positive volume  $\text{vol}(S)$  and the **centroid**  $\mathbf{x}^C$  of  $S$  is defined componentwise by

$$x_j^C = \frac{1}{\text{vol}(S)} \int \cdots \int_S x_j dx_1 \cdots dx_n \quad \text{for } 1 \leq j \leq n. \quad (8.52)$$

Using integral calculus one proves that  $\mathbf{x}^C = \mathbf{x}^0$  if  $S$  is centrally symmetric with respect to  $\mathbf{x}^0$ . For if  $a\mathbf{x} = a_0$  with  $a \neq 0$  is any hyperplane containing  $\mathbf{x}^0$ , i.e.  $a\mathbf{x}^0 = a_0$ , then the sets  $S_1 = S \cap \{\mathbf{x} \in \mathbb{R}^n : a\mathbf{x} \leq a_0\}$ ,  $S_2 = \{\mathbf{x} \in \mathbb{R}^n : a\mathbf{x} \geq a_0\}$  satisfy  $\dim S_i = n$ ,  $\text{vol}(S_i) = \frac{1}{2}\text{vol}(S)$  for  $i = 1, 2$  and

$$x_j^C = \frac{1}{\text{vol}(S)} \int \cdots \int_{S_1 \cup S_2} x_j dx_1 \cdots dx_n = \frac{2x_j^0}{\text{vol}(S)} \int \cdots \int_{S_1} dx_1 \cdots dx_n = x_j^0$$

for  $1 \leq j \leq n$  because  $\dim S = n$  and  $\mathbf{x} \in S_1$  if and only if  $2\mathbf{x}^0 - \mathbf{x} \in S_2$  by the central symmetry of  $S$  with respect to  $\mathbf{x}^0$ . If  $S$  is an *arbitrary* compact convex set of full dimension in  $\mathbb{R}^n$ , then a hyperplane passing through the centroid  $\mathbf{x}^C$  of  $S$  does in general not divide  $S$  into two parts of *equal* volume. However, defining  $S_1$  and  $S_2$  as above, the volumina of  $S_i$  and  $S$  satisfy **Mityagin's inequality**

$$\text{vol}(S_i) \geq \left( \frac{n}{n+1} \right)^n \text{vol}(S) \quad \text{for } i = 1, 2. \quad (8.53)$$

This inequality implies that every hyperplane that passes through the *centroid* divides the full-dimensional compact and convex set  $S \subseteq \mathbb{R}^n$  into two parts such that the ratio of the volume of *either* part to the volume of  $S$  is *at least*  $e^{-1} \approx 0.368$  and *at most*  $1 - e^{-1} \approx 0.632$  where  $e$  is Euler's number. The latter follows from (8.53) because  $\text{vol}(S) = \text{vol}(S_1) + \text{vol}(S_2)$  and thus for  $i = 1, 2$  and all  $n \geq 1$

$$\text{vol}(S_i) \leq \left( 1 - \left( \frac{n}{n+1} \right)^n \right) \text{vol}(S) \leq (1 - e^{-1}) \text{vol}(S).$$

Returning to polytopes, let  $\mathcal{X}^\leq = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  be the feasible set of a linear program in canonical form. The assumption of the *existence* of an interior point  $\mathbf{x}^0 \in \mathcal{X}$  with  $\mathbf{x}^0 > \mathbf{0}$  is

---

<sup>7</sup>Give me a place to stand and I will unhinge the earth!

equivalent to requiring that  $\dim \mathcal{X}^{\leq} = n$ . In this case the calculation of the centroid  $\mathbf{x}^C$  is (tedious, but) straightforward. To apply this notion to flat polytopes  $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$  having an interior point  $\mathbf{x}^0 \in \mathcal{X}$  with  $\mathbf{x}^0 > 0$ , remember that we assume that  $r(\mathbf{A}) = m$  where  $\mathbf{A}$  is an  $m \times n$  matrix. Let  $\mathbf{B}$  be any basis of  $\mathbf{A}$ , let  $(\mathbf{B}, \mathbf{R})$  be the corresponding partitioning of  $\mathbf{A}$  and consider

$$\mathcal{X}' = \{\mathbf{y} \in \mathbb{R}^{n-m} : \mathbf{B}^{-1}\mathbf{R}\mathbf{y} \leq \mathbf{B}^{-1}\mathbf{b}, \mathbf{y} \geq 0\}.$$

The assumption that there exists  $\mathbf{x}^0 \in \mathcal{X}$ ,  $\mathbf{x}^0 > 0$  then implies that  $\dim \mathcal{X}' = n - m$ . So we can calculate the centroid of  $\mathcal{X}'$  and the centroid of  $\mathcal{X}$  in a (tedious, but) straightforward way. Barycenters and centroids of polytopes do, of course, not always coincide.

In the case of the simplex  $S^{n+1}$  the notions of a barycenter and of a centroid coincide and they do as well – at least in certain cases – with the following concept of centrality which takes some of the arbitrariness out of the definition by way of an “objective function.”

Let  $\text{bar}(\mathbf{x})$  be any continuous function that maps the polytope  $\mathcal{X}$  into  $\mathbb{R}$  and that satisfies:

- $\text{bar}(\mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2) \leq \mu \text{bar}(\mathbf{x}^1) + (1 - \mu) \text{bar}(\mathbf{x}^2)$  for all  $0 \leq \mu \leq 1$  and  $\mathbf{x}^1, \mathbf{x}^2 \in \text{relint} \mathcal{X}$  with strict inequality if  $\mathbf{x}^1 \neq \mathbf{x}^2$  and  $0 < \mu < 1$ .
- $\text{bar}(\mathbf{x}) = +\infty$  for all  $\mathbf{x} \in \mathcal{X} - \text{relint} \mathcal{X}$ .

Such functions are called **barrier functions** with respect to  $\mathcal{X}$ . Since every barrier function is continuous and strictly convex,

$$\min\{\text{bar}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$$

exists and the minimum is attained at a *unique* point  $\mathbf{x}^{\text{bar}} \in \mathcal{X}$ . Since  $\text{bar}(\mathbf{x}) = +\infty$  for all  $\mathbf{x} \in \mathcal{X} - \text{relint} \mathcal{X}$  and  $\mathbf{x}^0 \in \mathcal{X}$ ,  $\mathbf{x}^0 > 0$  exists it follows that  $\mathbf{x}^{\text{bar}} \in \text{relint} \mathcal{X}$ .  $\mathbf{x}^{\text{bar}}$  is called the **center** of  $\mathcal{X}$  with respect to the barrier function  $\text{bar}(\mathbf{x})$ .

Different barrier functions exist for  $\mathcal{X}$  and thus different “centers” for  $\mathcal{X}$  result. Since  $\mathcal{X} \subseteq \mathbb{R}_+^n$  the reciprocal of the *geometric mean* gives the *geometric barrier function* for  $\mathcal{X}$ , i.e.,

$$g\text{bar}(\mathbf{x}) = \left( \prod_{j=1}^n x_j \right)^{-1/n}. \quad (8.54)$$

Taking the logarithm and ignoring constants we get the *logarithmic barrier function* for  $\mathcal{X}$ , i.e.,

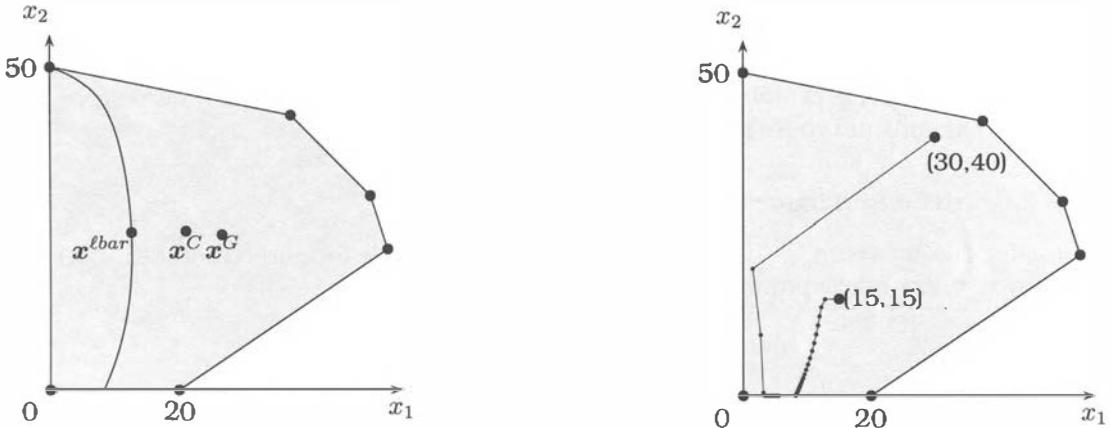
$$\ell\text{bar}(\mathbf{x}) = - \sum_{j=1}^n \log x_j. \quad (8.55)$$

Both barrier functions define, of course, the same “center” of  $\mathcal{X}$ , which is sometimes called the *analytic center*. We may as well call it the “geometric” or the “logarithmic” center of  $\mathcal{X}$ .

### 8.6.1 A Method of Centers

Consider the following algorithmic idea from the early 1960's which utilizes this notion of a center of a polytope  $\mathcal{X}$  with a nonempty relative interior. We are minimizing  $\mathbf{c}\mathbf{x}$  over  $\mathcal{X}$  and let  $z \in \mathbb{R}$  be any real number. We assume that  $\mathbf{x}^0 \in \mathcal{X}$  with  $\mathbf{x}^0 > 0$  exists and define

$$\mathcal{X}_z = \mathcal{X} \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{c}\mathbf{x} \leq z\}. \quad (8.56)$$



**Fig. 8.11.** Three “centers”, the log-central path and paths to optimality

The set  $\mathcal{X}_z$  is either empty or it has no  $x^1 \in \mathcal{X}_z$  with  $x^1 > 0$  and  $cx^1 < z$  or such a point exists. Choosing  $z$  initially large enough we can always avoid the first possibility. In the second case,  $\mathcal{X}_z$  is a face of  $\mathcal{X}$  and all points of  $\mathcal{X}_z$  are optimal for  $cx$ . Otherwise  $z$  is not the optimum objective function value of  $cx$  over  $\mathcal{X}$  and  $x^1 \in \mathcal{X}_z$  with  $x^1 > 0$  and  $cx^1 < z$  exists. Thus we can “iterate” by decreasing the value of  $z$ .

Given any barrier function  $bar(\mathbf{x})$  for  $\mathcal{X}$  let us denote its restriction to  $\mathcal{X}_z$  by  $bar_z(\mathbf{x})$  as e.g. given by

$$\ell bar_z(\mathbf{x}) = - \sum_{j=1}^n \log x_j - \log(z - cx)$$

in the case of the logarithmic barrier function. Denote by  $x^{bar}(z)$  the center of  $\mathcal{X}_z$  with respect to  $bar_z(\mathbf{x})$ . It follows that  $cx^{bar}(z) < z$ . Thus starting e.g. initially at  $x^{bar} \in \mathcal{X}$  we can construct a sequence of points  $x^{bar}(z_k)$  and a corresponding sequence of  $z_k$  with  $z_0 = cx^{bar} > z_1 > z_2 > \dots$ . Since  $\mathcal{X}$  and thus the  $\mathcal{X}_{z_k}$  are polytopes, the sequence  $\{z_k\}$  for  $k = 0, 1, 2, \dots$  is bounded from below and consequently, it has a point of accumulation  $z_\infty$ , say. It follows that either  $z_\infty$  is the optimum objective function of  $cx$  over  $\mathcal{X}$  or that there exists  $z_k$  in the infinite sequence with  $z_k < z_\infty$ . Thus an iterative application may “stall” temporarily, but it gets itself out of the “trap” eventually, i.e. there exists a subsequence  $\{z_{k_i}\}$ , say, of  $\{z_k\}$  that converges to the optimum objective function value of  $cx$  over  $\mathcal{X}$ .

In this (impractical) method we try to accomplish two objectives simultaneously: we want to minimize  $cx$  over the polytope  $\mathcal{X}$  and at the same time we want to stay in the relative interior of  $\mathcal{X}$ . Combining these two objectives into a single objective function, we are led to consider the family of problems

$$(P_\mu^{bar}) \quad \min\{cx + \mu bar(\mathbf{x}) : Ax = b, \mathbf{x} \geq 0\},$$

where  $\mu > 0$  is a parameter or some relative “weight”. Since  $cx$  is linear, the function  $cx + \mu bar(\mathbf{x})$  is strictly convex on  $\mathcal{X}$  for every  $\mu > 0$ . Since  $\mathcal{X}$  is bounded and  $\mathbf{x}^* \in \mathcal{X}$ ,  $\mathbf{x}^* > 0$  exists, the minimum exists and the minimizer of  $(P_\mu^{bar})$  is unique and positive. We can thus ignore the nonnegativity constraints of  $(P_\mu^{bar})$ . Assuming continuous differentiability of  $bar(\mathbf{x})$  we determine the minimizer

from the first order conditions for an extremum of the corresponding Lagrangean function for any fixed  $\mu > 0$ . Varying the parameter  $\mu$  we obtain a family of solutions  $x^{bar}(\mu)$ . For  $\mu \rightarrow +\infty$  the solution  $x^{bar}(\mu)$  converges towards the center  $x^{bar} \in \mathcal{X}$ , while for  $\mu \rightarrow 0$  it converges towards  $x^* \in \mathcal{X}$ , an optimal solution to (LP).

### 8.6.2 The Logarithmic Barrier Function

Let us consider the logarithmic barrier function  $\ellbar(\mathbf{x})$ , which is more tractable than  $gbar(\mathbf{x})$ , and denote by  $(P_\mu)$  the corresponding family of problems

$$(P_\mu) \quad \min\{\mathbf{c}\mathbf{x} + \mu \ellbar(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

where  $\mu > 0$  is an arbitrary parameter. Since there exists a unique *positive* minimizer for  $(P_\mu)$  we can ignore the nonnegativity constraints. We form the Lagrangean function

$$L(\mathbf{x}, \mathbf{u}, \mu) = \mathbf{c}\mathbf{x} + \mu \ellbar(\mathbf{x}) + \mathbf{u}^T(\mathbf{b} - \mathbf{A}\mathbf{x}),$$

where  $\mathbf{u} \in \mathbb{R}^m$  are the Lagrangean multipliers. The minimizer must satisfy the first-order conditions for an extremum of  $L(\mathbf{x}, \mathbf{u}, \mu)$ . The first order conditions for  $L(\mathbf{x}, \mathbf{u}, \mu)$  yield the nonlinear system of equations

$$c_j - \mu x_j^{-1} - \sum_{i=1}^m u_i a_j^i = 0 \text{ for } 1 \leq j \leq n, \quad \sum_{j=1}^n a_j^i x_j - b_i = 0 \text{ for } 1 \leq i \leq m,$$

for which we seek the unique solution  $(\mathbf{x}(\mu), \mathbf{u}(\mu))$  such that  $\mathbf{x}(\mu) > \mathbf{0}$ .

The parameter  $\mu > 0$  is assumed to be fixed, but all that we want is the unique solution for the “limiting” case where  $\mu \rightarrow 0$ . Define  $r_j = \mu x_j^{-1}$  for  $1 \leq j \leq n$  and  $\mathbf{r}^T = (r_1, \dots, r_n)$ . Then the first order conditions for an extremum of  $L(\mathbf{x}, \mathbf{u}, \mu)$  become in matrix form

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0} \tag{8.57}$$

$$\mathbf{A}^T \mathbf{u} + \mathbf{r} - \mathbf{c}^T = \mathbf{0} \tag{8.58}$$

$$\mathbf{x} * \mathbf{r} - \mu \mathbf{e} = \mathbf{0}, \tag{8.59}$$

where  $\mathbf{x} * \mathbf{r} = (x_1 y_1, \dots, x_n y_n)^T$  is the **Hadamard product** and  $\mathbf{e}^T = (1, \dots, 1) \in \mathbb{R}^n$ . If  $(\mathbf{x}(\mu), \mathbf{u}(\mu))$  is a feasible solution to the first order conditions with  $\mathbf{x}(\mu) > \mathbf{0}$ , then  $\mathbf{r} = \mathbf{r}(\mu) > \mathbf{0}$  and thus  $(\mathbf{u}(\mu), \mathbf{r}(\mu))$  is an *interior* feasible solution to the linear program

$$(dLP) \quad \max\{\mathbf{b}^T \mathbf{u} : \mathbf{A}^T \mathbf{u} + \mathbf{r} = \mathbf{c}^T, \mathbf{r} \geq \mathbf{0}\},$$

which is the dual to the linear program (LP) of the introduction to this chapter. From (8.59) we find  $\mathbf{r}^T \mathbf{x} = n\mu$ . Thus from (8.57) and (8.58) we have

$$\mathbf{r}^T \mathbf{x} = n\mu = \mathbf{c}\mathbf{x} - \mathbf{u}^T \mathbf{b}, \tag{8.60}$$

which is the *duality gap* for the primal-dual pair  $(\mathbf{x}(\mu), \mathbf{u}(\mu), \mathbf{r}(\mu))$ . Consequently, any primal-dual pair  $(\mathbf{x}(0), \mathbf{u}(0), \mathbf{r}(0))$  with  $\mathbf{x}(0) \geq \mathbf{0}$  and  $\mathbf{r}(0) \geq \mathbf{0}$ , i.e. any feasible solution to (8.57), (8.58), (8.59) for  $\mu = 0$ , is a pair of optimal solutions to (LP) and (dLP).

For  $0 \leq \mu < \infty$  the loci of  $\mathbf{x}(\mu)$  form a path connecting the log-center  $\mathbf{x}^{\ell bar}$  of  $\mathcal{X}$  to some point in the optimal face of  $\mathcal{X}$ . Likewise, the loci  $\mathbf{x}(\nu)$  of the corresponding maximization problem  $\max\{\mathbf{c}\mathbf{x} - \nu\ell bar(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$  for  $0 \leq \nu < +\infty$  form a path connecting  $\mathbf{x}^{\ell bar}$  to an optimal face. The path connecting the optimal face of  $\mathcal{X}$  with respect to maximization to the one for minimization is the **log-central path** or simply, the *central path* of  $\mathcal{X}$ , which – by construction – passes through the log-center  $\mathbf{x}^{\ell bar}$  of  $\mathcal{X}$ ; see the left part of Figure 8.11 where the barycenter  $\mathbf{x}^G$ , the centroid  $\mathbf{x}^C$  and the log-central path for the polytope of Exercise 8.2 (ii) are displayed.

To find an approximatively optimal solution to (LP) we must solve the system of nonlinear equations (8.57), (8.58), (8.59) for  $\mu \approx 0$ . This is done e.g. by a multivariate version of Newton's method for finding the root of a (nonlinear) equation. Let  $\mathbf{F}(\mathbf{z})$  be any continuously differentiable function mapping  $\mathbb{R}^t$  into  $\mathbb{R}^q$ . We wish to find  $\mathbf{z}^0 \in \mathbb{R}^t$  such that  $\mathbf{F}(\mathbf{z}^0) = \mathbf{0}$  or componentwise, such that  $F_i(\mathbf{z}^0) = 0$  for  $1 \leq i \leq q$ . By the multivariate mean-value theorem of differential calculus

$$F_i(\mathbf{z} + \Delta\mathbf{z}) = F_i(\mathbf{z}) + \nabla F_i(\mathbf{z} + \theta_i \Delta\mathbf{z}) \Delta\mathbf{z}$$

for some  $0 \leq \theta_i \leq 1$ , where  $\nabla F_i = \left( \frac{\partial F_i}{\partial z_j} \right)_{j=1,\dots,t}$  is the vector of the first derivatives of  $F_i$  and  $1 \leq i \leq q$ .  $\Delta\mathbf{z} = (\Delta z_1, \dots, \Delta z_t)^T$  is a vector of “change” for the components of  $\mathbf{z}$ , e.g.  $\Delta\mathbf{z} = \mathbf{z}' - \mathbf{z}$  for some  $\mathbf{z}' \neq \mathbf{z} \in \mathbb{R}^t$ . Given a “trial” solution  $\mathbf{z} \in \mathbb{R}^t$  for the root  $\mathbf{z}^0$  of  $\mathbf{F}$  the “new” trial solution is  $\mathbf{z} + \Delta\mathbf{z}$ . We set  $\mathbf{F}(\mathbf{z} + \Delta\mathbf{z}) = \mathbf{0}$  and ignore the dependence of  $\nabla F_i$  on  $\Delta\mathbf{z}$  by setting all  $\theta_i = 0$ . Denoting by  $\nabla \mathbf{F} = \left( \frac{\partial F_i}{\partial z_j} \right)_{i=1,\dots,q, j=1,\dots,t}$  the  $q \times t$  matrix of the first derivatives we get the system of linear equations

$$\nabla \mathbf{F}(\mathbf{z}) \Delta\mathbf{z} = -\mathbf{F}(\mathbf{z}) \quad (8.61)$$

in the variables  $\Delta\mathbf{z}$ , where  $\nabla \mathbf{F}(\mathbf{z})$  and  $\mathbf{F}(\mathbf{z})$  are evaluated at the current iterate  $\mathbf{z}$ . Every solution  $\Delta\mathbf{z}$  to this system gives a *Newton direction* or a *Newton step*. The “new” iterate is  $\mathbf{z} + \Delta\mathbf{z}$  or more generally  $\mathbf{z} + \alpha \Delta\mathbf{z}$  where  $\alpha \geq 0$  is the *step length*. If started “close” to a root  $\mathbf{z}^0$  of  $\mathbf{F}$ , then the resulting iterative scheme converges rather fast to the root  $\mathbf{z}^0$ , but in general it does not converge to  $\mathbf{z}^0$ .

For the nonlinear system (8.57), (8.58), (8.59) the mapping  $\mathbf{F}(\mathbf{z})$  is

$$\mathbf{F}(\mathbf{z}) = \begin{pmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{A}^T \mathbf{u} + \mathbf{r} - \mathbf{c}^T \\ \mathbf{x} * \mathbf{r} - \mu \mathbf{e} \end{pmatrix},$$

where  $\mathbf{z} = (\mathbf{x}, \mathbf{u}, \mathbf{r})$  is the vector of variables. Let  $\mathbf{x} > \mathbf{0}$ ,  $\mathbf{r} > \mathbf{0}$  and  $\mathbf{u}$  arbitrary be any fixed trial solution to our problem where  $\mu > 0$  is arbitrary, but fixed as well. Forming  $\nabla \mathbf{F}$  and evaluating  $\nabla \mathbf{F}$  at the point  $(\mathbf{x}, \mathbf{u}, \mathbf{r})$  we get from (8.61) the linear equations in the variables  $\Delta\mathbf{z} = (\Delta\mathbf{x}, \Delta\mathbf{u}, \Delta\mathbf{r})$

$$\begin{pmatrix} \mathbf{A} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}^T & \mathbf{I}_n \\ \mathbf{R} & \mathbf{O} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \\ \Delta\mathbf{r} \end{pmatrix} = \begin{pmatrix} \mathbf{b} - \mathbf{A}\mathbf{x} \\ \mathbf{c}^T - \mathbf{A}^T \mathbf{u} - \mathbf{r} \\ \mu \mathbf{e} - \mathbf{x} * \mathbf{r} \end{pmatrix}, \quad (8.62)$$

where  $\mathbf{D} = \text{diag}(x_1, \dots, x_n)$  and  $\mathbf{R} = \text{diag}(r_1, \dots, r_n)$ . Since  $\mathbf{x} > \mathbf{0}$  and  $\mathbf{r} > \mathbf{0}$  we have  $r(\mathbf{R}) = r(\mathbf{D}) = n$ . Since  $r(\mathbf{A}) = m$  by our blanket assumption, the  $(2n+m) \times (2n+m)$  matrix of the linear system (8.62) is nonsingular and hence the solution is unique.

Given  $x > 0$ ,  $r > 0$ , a vector  $u \in \mathbb{R}^m$  and  $\mu > 0$  denote

$$f = b - Ax, \quad g = c^T - A^T u - r, \quad h = \mu e - x * r \text{ and } B = AR^{-1}DA^T.$$

Since  $r(A) = m$  the inverse  $B^{-1}$  exists and we can solve (8.62). Since  $r > 0$  and  $x > 0$  the matrix  $T = (R^{-1}D)^{1/2}$  is well defined. Let

$$S = I_n - TA^T B^{-1} AT \text{ where } T = (R^{-1}D)^{1/2} \text{ and } B = AT^2 A^T \quad (8.63)$$

be the orthogonal projection operator on the subspace  $\{x \in \mathbb{R}^n : ATx = 0\}$ ; see Remark 8.1. After some algebraic manipulations and simplifications we find the following solution  $\Delta x$ ,  $\Delta u$  and  $\Delta r$  to (8.62):

$$\Delta x = -TSTc^T + \mu TSTD^{-1}e + T^2A^T B^{-1}f \quad (8.64)$$

$$\Delta u = B^{-1}b - \mu B^{-1}AR^{-1}e + B^{-1}AT^2g \quad (8.65)$$

$$\Delta r = -A^T B^{-1}b + \mu A^T B^{-1}AR^{-1}e + T^{-1}STg \quad (8.66)$$

The first term  $\Delta x^s$ , say, of  $\Delta x$  is a *steepest descent direction* in the “transformed” space. The second term  $\Delta x^c$ , say, of  $\Delta x$  is the *centering direction*. The third term reduces the infeasibility in the system of equations to zero if  $\alpha = 1$  and is called the *feasibility direction*. A similar interpretation can be given to the three terms of  $(\Delta u, \Delta r)$  for the dual linear program (dLP); see the text.

We can now state an *iterative scheme* that is designed to find a solution  $(x, u, r)$  with  $x \geq 0$  and  $r \geq 0$  to (8.57), (8.58), (8.59) for  $\mu \approx 0$ . We start with any triplet  $(x, u, r)$  satisfying  $x > 0$  and  $r > 0$  and some  $\mu > 0$ , e.g.  $\mu = 0.1(r^T x/n)$ . We calculate the orthoprojection (8.63) to find the direction vectors (8.64), (8.65), (8.66). Then we update

$$x^{new} = x + \frac{1}{\alpha_p} \Delta x, \quad u^{new} = u + \frac{1}{\alpha_d} \Delta u, \quad r^{new} = r + \frac{1}{\alpha_d} \Delta r,$$

where  $\alpha_p$  and  $\alpha_d$  are step lengths that are chosen to maintain the positivity of  $x^{new}$  and  $r^{new}$ . We reduce  $\mu$  by setting  $\mu^{new} = 0.1(r^{new})^T x^{new}/n$  if  $cx^{new} > b^T u^{new}$  and set  $\mu^{new} = 2(r^{new})^T x^{new}/n$  otherwise. This is motivated by relation (8.60) and in the second case, designed to permit the correction of a possible error. We thus have a new triplet  $(x^{new}, u^{new}, r^{new})$  and  $\mu^{new} > 0$  and we can iterate until primal and dual feasibility are attained and the duality gap is smaller than some tolerance, e.g. smaller than  $10^{-6}$ . To ensure positivity of  $x^{new}$  and  $r^{new}$  one chooses  $\alpha_p$  and  $\alpha_d$  e.g. as follows:

$$\alpha_p = \max\left\{1, -\frac{\Delta x_1}{0.95x_1}, \dots, -\frac{\Delta x_n}{0.95x_n}\right\}, \quad \alpha_d = \max\left\{1, -\frac{\Delta r_1}{0.95r_1}, \dots, -\frac{\Delta r_n}{0.95r_n}\right\}$$

In the right part of Figure 8.11 we display the path to optimality when the algorithm is run with the data of Exercise 8.2 (ii) where  $x_1^0 = 30$ ,  $x_2^0 = 40$ ,  $r^0 = c^T - A^T u^0$  and  $u_j^0 = -0.1$  for  $1 \leq j \leq 4$ .

### 8.6.3 A Newtonian Algorithm

Call a triplet  $(x, u, r) \in \mathbb{R}^{2n+m}$  a *feasible triplet* if  $x \in \mathbb{R}^n$  is a feasible solution to (LP) with  $x > 0$  and  $(u, r) \in \mathbb{R}^{m+n}$  a feasible solution to (dLP) with  $r > 0$ , respectively. From (8.58) and (8.59) we know

that every feasible triplet satisfies  $\mathbf{r}^T \mathbf{x} = \mathbf{c} \mathbf{x} - \mathbf{b}^T \mathbf{u}$ . To satisfy (8.59) as well we need  $\mu = \mathbf{r}^T \mathbf{x}/n$  since  $e^T(\mathbf{x} * \mathbf{r}) = \mathbf{r}^T \mathbf{x}$ . Consequently, a feasible triplet  $(\mathbf{x}, \mathbf{u}, \mathbf{r})$  belongs to the log-central path if and only if  $\|\mathbf{x} * \mathbf{r} - \mu e\| = 0$  for this value of  $\mu$ .

Call a feasible triplet  $(\mathbf{x}, \mathbf{u}, \mathbf{r})$  “close” to the log-central path if for  $\mu = \mathbf{r}^T \mathbf{x}/n$  we have  $\|\mathbf{x} * \mathbf{r} - \mu e\| \leq \Theta \mu$  for some “small”  $\Theta \geq 0$ , where  $\mathbf{x} * \mathbf{r}$  is the Hadamard-product of  $\mathbf{x}$  and  $\mathbf{r}$ . We note the following inequality:

$$\|\mathbf{s} * \mathbf{t}\| \leq \frac{1}{2} \|\mathbf{s} + \mathbf{t}\|^2 \quad \text{for all } \mathbf{s}, \mathbf{t} \in \mathbb{R}^n \text{ with } \mathbf{s}^T \mathbf{t} \geq 0. \quad (8.67)$$

**Remark 8.10** Let  $(\mathbf{x}, \mathbf{u}, \mathbf{r}) \in \mathbb{R}^{2n+m}$  be a feasible triplet satisfying

$$\|\mathbf{x} * \mathbf{r} - \mu e\| \leq \Theta \mu, \quad \mathbf{r}^T \mathbf{x} = n\mu \quad (8.68)$$

where  $\Theta$  is a real number that satisfies

$$0 \leq \Theta \leq \frac{1}{2}, \quad \Theta^2 + \delta^2 \leq 2\Theta(1 - \Theta)(1 - \frac{\delta}{\sqrt{n}}) \quad (8.69)$$

for some  $\delta$  with  $0 < \delta < \sqrt{n}$ . Let  $\hat{\mu} = \mu(1 - \delta/\sqrt{n})$  and  $(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{r}}) \in \mathbb{R}^{2n+m}$  be defined by  $\hat{\mathbf{x}} = \mathbf{x} + \Delta \mathbf{x}$ ,  $\hat{\mathbf{u}} = \mathbf{u} + \Delta \mathbf{u}$ ,  $\hat{\mathbf{r}} = \mathbf{r} + \Delta \mathbf{r}$ , where  $(\Delta \mathbf{x}, \Delta \mathbf{u}, \Delta \mathbf{r})$  is a solution to (8.62) with  $\mu$  replaced by  $\hat{\mu}$ . Then

- (i)  $\mathbf{c} \hat{\mathbf{x}} - \mathbf{b}^T \hat{\mathbf{u}} = \hat{\mathbf{r}}^T \hat{\mathbf{x}} = n\hat{\mu}$ ,
- (ii)  $\|\hat{\mathbf{x}} * \hat{\mathbf{r}} - \hat{\mu} e\| \leq \Theta \hat{\mu}$ , and
- (iii)  $(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{r}})$  is a feasible triplet.

This leads to the following Newtonian algorithm, which takes the data of linear program (LP), a “reduction” parameter  $\delta$ , a number  $p$  for the desired precision as well as a feasible triplet  $(\mathbf{x}^0, \mathbf{u}^0, \mathbf{r}^0)$  as inputs.

#### Newtonian Algorithm ( $\delta, p, m, n, \mathbf{A}, \mathbf{x}^0, \mathbf{u}^0, \mathbf{r}^0$ )

Step 0: Set  $k := 0$ ,  $\mathbf{R}_0 := \text{diag}(r_1^0, \dots, r_n^0)$ ,  $\mathbf{D}_0 := \text{diag}(x_1^0, \dots, x_n^0)$ , and  $\mu_0 = (\mathbf{r}^0)^T \mathbf{x}^0/n$ .

Step 1: **if**  $\mu^k/\mu^0 \leq 2^{-p}$  **stop** “ $(\mathbf{x}^k, \mathbf{u}^k, \mathbf{r}^k)$  is a  $p$ -optimal triplet.”

Step 2: Set  $\mu_{k+1} = \mu_k(1 - \delta/\sqrt{n})$  and solve  $\mathbf{R}_k \Delta \mathbf{x} + \mathbf{D}_k \Delta \mathbf{r} = \mu_{k+1} \mathbf{e} - \mathbf{r}^k * \mathbf{x}^k$ ,  $\mathbf{A} \Delta \mathbf{x} = \mathbf{0}$ ,  $\mathbf{A}^T \Delta \mathbf{u} + \Delta \mathbf{r} = \mathbf{0}$  for  $(\Delta \mathbf{x}, \Delta \mathbf{u}, \Delta \mathbf{r})$ .

Step 3: Set  $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}$ ,  $\mathbf{u}^{k+1} = \mathbf{u} + \Delta \mathbf{u}$ ,  $\mathbf{r}^{k+1} = \mathbf{r} + \Delta \mathbf{r}$ ,  $\mathbf{D}_{k+1} = \text{diag}(x_1^{k+1}, \dots, x_n^{k+1})$ ,  $\mathbf{R}_{k+1} = \text{diag}(r_1^{k+1}, \dots, r_n^{k+1})$ ; replace  $k + 1$  by  $k$ ; **go to** Step 1.

For the solution of the linear equations in Step 2 of the algorithm we use (8.64), (8.65), (8.66) with  $\mu$  replaced by  $\hat{\mu}$ . They simplify because  $\mathbf{f} = \mathbf{0}$  and  $\mathbf{g} = \mathbf{0}$  in the case of feasible triplets  $(\mathbf{x}, \mathbf{u}, \mathbf{r})$ .

**Remark 8.11** (Correctness and finiteness) For every  $\delta \in \mathbb{R}$  with  $0 < \delta < \sqrt{n}$  and  $\Theta \in \mathbb{R}$  satisfying (8.69) the Newtonian algorithm iterates at most  $\mathcal{O}(p\sqrt{n})$  times where  $p \geq 1$  is any integer and where  $(\mathbf{x}^0, \mathbf{u}^0, \mathbf{r}^0)$  is any feasible triplet satisfying  $\|\mathbf{r}^0 * \mathbf{x}^0 - \mu_0 \mathbf{e}\| \leq \Theta \mu_0$  for  $\mu_0 = (\mathbf{r}^0)^T \mathbf{x}^0/n$ .

Like in the case of the projective algorithm we get  $\mathcal{O}(\sqrt{n}L)$  convergence for a linear program of digital size  $L$ . We start the Newtonian algorithm by essentially the same trick that we have used to start the projective algorithm. Consider the linear program (LP') of Chapter 8.5 in  $n+2$  variables and  $m+1$  equations. Then  $x_j^0 = \kappa$  for  $1 \leq j \leq n+2$  is a feasible interior point  $\mathbf{x}^0$  to (LP'). To obtain a suitable dual solution  $(\mathbf{u}^0, \mathbf{r}^0)$  where  $\mathbf{u}^0 \in \mathbb{R}^{m+1}$  we set  $u_j^0 = 0$  for  $1 \leq j \leq m$  and choose  $u_{m+1}^0$  as follows: let  $r_j^0 = c'_j - u_{m+1}^0$  for  $1 \leq j \leq n+2$  where  $c'_j = c_j$  for  $1 \leq j \leq n$ ,  $c'_{n+1} = 0$ ,  $c'_{n+2} = M$ . By making  $u_{m+1}^0$  a small enough negative number we get  $\mathbf{r}^0 > 0$ . To satisfy the “closeness” criterion (8.68) we calculate  $\mu_0 = (\mathbf{r}^0)^T \mathbf{x}^0 / (n+2) = \kappa(C - u_{m+1}^0)$  where  $C = \sum_{j=1}^{n+2} c'_j / (n+2)$ . So

$$\|\mathbf{r}^0 * \mathbf{x}^0 - \mu^0 \mathbf{e}\| = \kappa \sqrt{\sum_{j=1}^{n+2} (c'_j - C)^2} = \kappa C^*.$$

Choosing e.g.  $\delta = \Theta = 0.35$  and  $u_{m+1}^0$  such that  $u_{m+1}^0 \leq \min\{-1, C - C^*/\Theta\}$  the various assumptions that we have made are all satisfied for (LP'). Thus we can start the Newtonian algorithm.

In the right part of Figure 8.11 we display the sequence of iterates of the Newtonian algorithm for the data of Exercise 8.2 (ii) when started with  $x_1^0 = x_2^0 = 15$ ,  $\mathbf{r}^0 = \mathbf{c}^T - \mathbf{A}^T \mathbf{u}^0$ ,  $u_1^0 = -0.1$ ,  $u_2^0 = -0.4$ ,  $u_3^0 = -0.2$ ,  $u_4^0 = -0.3$  and  $\delta = 0.40$ .

## 8.7 Exercises

---

### Exercise 8.1

- (i) Prove that  $B_\rho^{n+1} \subseteq S^{n+1}$  if and only if  $0 \leq \rho \leq r = 1/\sqrt{n(n+1)}$ .
  - (ii) Prove that  $B_\rho^{n+1} \supseteq S^{n+1}$  if and only if  $\rho \geq \sqrt{n/(n+1)}$ . (Hint: Use the Lagrangean multiplier technique to show that  $y_i = -n\rho r + 1/(n+1)$ ,  $y_j = \rho r + 1/(n+1)$  for all  $j \neq i$  solves the minimization problem  $\min\{y_i : \mathbf{y} \in B_\rho^{n+1}\}$  and that  $y_i = n\rho r + 1/(n+1)$ ,  $y_j = -\rho r + 1/(n+1)$  for all  $j \neq i$  solves the corresponding maximization problem.)
- 

(i) A point  $\mathbf{y} \in B_\rho^{n+1}$  is in  $S^{n+1}$  if and only if  $\mathbf{y} \geq \mathbf{0}$ . So we calculate the minimum value a single element of the vector  $\mathbf{y}$  can take. Since the feasible set of the problem

$$\min \left\{ y_i : \sum_{j=1}^{n+1} y_j = 1, \sum_{j=1}^{n+1} \left( y_j - \frac{1}{n+1} \right)^2 \leq \rho^2 \right\}$$

is compact, i.e., closed and bounded, there exists an optimal solution to it and we can use the Lagrangean multiplier technique to calculate it. Since  $\min\{y_i : \sum_{j=1}^{n+1} y_j = 1\} = -\infty$ , the

inequality in this minimization problem is binding and the Lagrangean multiplier technique applies. Forming the Lagrangean we have

$$\mathcal{L} = y_i - \mu \left( 1 - \sum_{j=1}^{n+1} y_j \right) - \lambda \left( \rho^2 - \sum_{j=1}^{n+1} \left( y_j - \frac{1}{n+1} \right)^2 \right).$$

From the first order conditions we have

$$1 + \mu + 2\lambda \left( y_i - \frac{1}{n+1} \right) = 0 \quad (1)$$

$$\mu + 2\lambda \left( y_j - \frac{1}{n+1} \right) = 0 \quad \text{for } 1 \leq j \leq n+1, j \neq i. \quad (2)$$

Adding up (1) and (2) for  $1 \leq j \leq n+1, j \neq i$  we get  $1 + (n+1)\mu + 2\lambda(\sum_{j=1}^{n+1} y_j - \frac{n+1}{n+1}) = 0$  and thus

$$\mu = -\frac{1}{n+1}. \quad (3)$$

Substituting  $\mu$  in (1) and (2) we get

$$2\lambda \left( y_i - \frac{1}{n+1} \right) = -\frac{n}{n+1} \quad (4)$$

$$2\lambda \left( y_j - \frac{1}{n+1} \right) = -\frac{1}{n+1}. \quad (5)$$

Squaring these two equations and adding them up gives

$$4\lambda\rho^2 = \frac{n}{n+1} = \frac{n^2}{n(n+1)} = n^2 r^2,$$

and thus we get  $2\lambda = nr$  for the minimization problem. Substituting in (4) and (5) we have that

$$y_i = -nr\rho + \frac{1}{n+1}, \quad y_j = r\rho + \frac{1}{n+1} \quad \text{for } 1 \leq j \leq n+1, j \neq i.$$

Now if the minimum component of  $\mathbf{y}$  is nonnegative then  $\mathbf{y} \geq \mathbf{0}$  and thus  $B_\rho^{n+1} \subseteq S^{n+1}$  if and only if  $-nr\rho + \frac{1}{n+1} \geq 0$ , i.e., if and only if  $\rho \leq r$ .

**(ii)** The point  $\mathbf{y} \in S^{n+1}$  is in  $B_\rho^{n+1}$  if and only if  $\sum_{j=1}^{n+1} (y_j - \frac{1}{n+1})^2 \leq \rho^2$ . Developing the square in the left-hand side of the inequality we get

$$\sum_{j=1}^{n+1} y_j^2 - \frac{2}{n+1} + \frac{1}{n+1} \leq \rho^2$$

because  $\sum_{j=1}^{n+1} y_j = 1$ . Now since  $\max\{\sum_{j=1}^{n+1} y_j^2 : \mathbf{y} \in S^{n+1}\} = 1$  we have  $S^{n+1} \subseteq B_\rho^{n+1}$  if and only if

$$1 - \frac{2}{n+1} + \frac{1}{n+1} = \frac{n}{n+1} \leq \rho^2,$$

i.e. if and only if  $\rho \geq \sqrt{\frac{n}{n+1}}$ .

---

## Exercise 8.2

- (i) Write a computer program for the basic algorithm using any subroutine for inverting a matrix. Use as test problem e.g. anyone of the following class:  $\min\{\sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_j x_j = a_0, x_j \geq 0\}$  where  $a_0 > 0$ ,  $a_j > 0$ ,  $c_j \geq 0$  for all  $1 \leq j \leq n$  and  $c_k = 0$  for some  $k$ . To initialize you may use either  $x_j^0 = a_0/na_j$  or  $x_j = a_0/\sum_{k=1}^n a_k$  for  $1 \leq j \leq n$ .
- (ii) Use your program to solve the problem  $\min\{x_2 : x_1 + 5x_2 \leq 250, x_1 + x_2 \leq 80, 3x_1 + x_2 \leq 180, 2x_1 - 3x_2 \leq 40, x_1 \geq 0, x_2 \geq 0\}$ , after bringing it into standard form, with  $x_1^0 = 30, x_2^0 = 40$  as a starting point.
- 

- (i) The following is an implementation of the basic algorithm in MATLAB.

```
%%%%%%%
%% This is the implementation of the Basic Projective Algorithm.
%%
%% NAME      : basprojal
%% PURPOSE: Solve the LP: min {cx: A x = b, x >=0}
%% INPUT    : The matrix A, the vectors c and b and a
%%             starting interior point x.
%% OUTPUT   : tabular format as follows
%%             iteration x(1) x(2) ...
%%             ....
%%             Optimal value
%%%%%%%
ptol=10^-5;
maxit = 100;
[m n] = size(A);
for i=1:m,
    sum=0;
    for j=1:n-m,
        sum=sum+A(i,j)*x(j);
    end;
    x(i+n-m)=b(i)-sum;
end;
```

```

if (any(x) <= 0)
    fprintf('Error. The starting point is NOT an interior point');
    stop;
end;

k = 1;
alpha = 1.0;
z = c*x';
z0= z;
while ( z0/z > ptol);
    x0 = x;
    z0 = c*x0';
    D = diag(x0);
    G = A*D*D*A';
    Ginv = inv(G);
    P = eye(n) - D*A'*Ginv*A*D;
    p = P*D*c';
    d = P*ones(n,1);
    gamma= p'*d;
    beta = n - norm(d,2)^2;
    normpsq= norm(p,2)^2;
    nmqu=sqrt(normpsq + ((z0-gamma)^2/(1+beta)) - (z0^2/(n+1)));
    nn = (z0-gamma)/(1+beta);
    dir= (D*(p - nn*d))';
    rho = alpha;
    num=(1+beta)*(n+1)*rho;
    den=(1+beta)*sqrt(n*(n+1))*nmqu+rho*(gamma*(n+1)-(n-beta)*z0);
    trho=num/den;
    x = x0 - trho*dir;
    fprintf('%3d ',k);
    fprintf('%10.5f ',x);
    fprintf('\n');
    if (k == maxit)
        fprintf('Maximum Iterations (%d) Exceeded - Possible Cycling\n',maxit)
        return;
    end;
    k = k + 1;
end;
fprintf('Optimal value: %10.5f\n',c*x');

```

To test the program we use the following LP

$$\min\{10x_1 + 5x_2 : 3x_1 + 4x_2 + x_3 = 10, x_1, x_2, x_3 \geq 0\}$$

with  $(\frac{10}{3}, \frac{10}{8}, \frac{10}{3})$  as starting point. The data are provided in a file with the name `bpadat.m`:

```

A=[3 4 1];
b=[10];

```

```
c=[10 5 0];
x=[10/9 10/12 10/3];
```

The output is shown in the left side of the following figure. On the right side we show the output when the other suggested starting point, i.e.,  $(\frac{10}{8}, \frac{10}{8}, \frac{10}{8})$  is used.

```
>>clear
>> bpadat
>> basprojal
>> clear
>> bpadat
>> basprojal
    1    0.13837    0.94557    5.80260    4    0.24749    1.67965    2.53894
    2    0.13482    0.09822    9.20265    5    0.37424    0.63223    6.34836
    3    0.00647    0.06605    9.71639    6    0.09480    0.27662    8.60912
    4    0.00529    0.00091    9.98049    7    0.04217    0.04527    9.69242
    5    0.00001    0.00083    9.99664    8    0.00414    0.02358    9.89325
    6    0.00001    0.00000    9.99996    9    0.00276    0.00102    9.98763
    7    0.00000    0.00000    10.00000    10   0.00003    0.00083    9.99656
Optimal value: 0.00000                         Optimal value: 0.00000
>>
```

**(ii)** The data file `bpadat.m` for the problem with the suggested starting point is:

```
A=[1 5 1 0 0 0; 1 1 0 1 0 0; 2 -3 0 0 1 0; 3 1 0 0 0 1];
b=[250 80 40 180];
c=[0 1 0 0 0 0];
x=[30 40 0 0 0 0];
```

The output from running the program with this data file is shown next.

```
>> clear
>> bpadat
>> basprojal
    1    23.76556    36.00377    46.21561    20.23067    100.48018    72.69956
    2    21.88479    25.67405    99.74493    32.44115    73.25257    88.67156
    3    19.72851    12.31375    168.70274    47.95774    37.48424    108.50073
    4    13.98604    3.33545    219.33673    62.67851    22.03426    134.70644
    5    11.34214    0.16120    237.85187    68.49666    17.79932    145.81239
    6    11.20626    0.00003    238.79361    68.79371    17.58755    146.38118
    7    11.20624    0.00000    238.79376    68.79376    17.58751    146.38127
Optimal value: 0.00000
>>
```

### Exercise 8.3

(i) Show that  $E(\mathbf{x}^0, R) \subset E(\mathbf{x}^0, R')$  for all  $0 \leq R < R'$ .

(ii) Show that  $\mathbf{x} = (1 \pm R/\sqrt{2+R^2}) \mathbf{x}_C \pm (Rx_i^0/\sqrt{2+R^2}) \mathbf{u}_i$  solves the problem  $\max\{x_i : \mathbf{x} \in E(\mathbf{x}^0, R)\}$  if both plus signs are used and the corresponding minimization problem in the opposite case where  $\mathbf{u}_i \in \mathbb{R}^n$  is the  $i^{\text{th}}$  unit vector. Moreover, for  $R \rightarrow +\infty$  the minimizing point exists and is given by  $\mathbf{x} = \mathbf{x}^0 - x_i^0 \mathbf{u}_i$  where  $1 \leq i \leq n$ . (Hint: It suffices to show that  $B_\rho^{n+1} \subset B_{\rho'}^{n+1}$  for  $\rho < \rho'$ .)

(iii) Show that  $\det(\mathbf{I}_n + (1+R^2)\mathbf{e}\mathbf{e}^T) = 1 + n + nR^2$  and that the volume of  $E(\mathbf{x}^0, R)$  is given by

$$\text{vol}(E(\mathbf{x}^0, R)) = g^n(\mathbf{x}^0) R^n \pi^{n/2} \sqrt{1+n+nR^2}/\Gamma(1+n/2)$$

where  $g(\mathbf{x}^0)$  is the geometric mean of  $\mathbf{x}^0$ .

(iv) Let  $\lambda_1$  be the smallest eigenvalue of  $H^{-1} = D(\mathbf{I}_n + (1+R^2)\mathbf{e}\mathbf{e}^T)D$ . Show that the length of the smallest principal axis  $R\sqrt{\lambda_1}$  of  $E(\mathbf{x}^0, R)$  satisfies  $R\sqrt{\lambda_1} \leq R(1+n+nR^2)^{1/2n}g(\mathbf{x}^0)$ .

(v) Show that for the data of Exercise 8.2 (ii) the ellipsoids  $E(\mathbf{x}^0, R)$  in the space of the variables  $x_1$  and  $x_2$  are given by  $(0.7614 + 0.6029R^2)x_1^2 + (3.79575 + 3.125925R^2)x_2^2 + (1.7955 + 1.3797R^2)x_1x_2 - (117.504 + 87.822R^2)x_1 - (357.525 + 285.795R^2)x_2 \leq -(8913.06 + 6803.73R^2)$ .

---

**(i)** Since  $B_\rho^{n+1} \subset B_{\rho'}^{n+1}$  implies  $T_0^{-1}(B_\rho^{n+1}) \subset T_0^{-1}(B_{\rho'}^{n+1})$ , i.e.  $E(\mathbf{x}^0, R) \subset E(\mathbf{x}^0, R')$ , it suffices to show that  $B_\rho^{n+1} \subset B_{\rho'}^{n+1}$  for all  $0 \leq R < R'$ . But this follows from the fact that (8.15), i.e.,

$$R = \rho \sqrt{\frac{n+1}{n(r^2 - \rho^2)}}$$

preserves monotonicity, that is,  $R < R'$  implies  $\rho < \rho'$  because  $\rho = R((n+1)(1+n+nR^2))^{-1/2}$  and  $d\rho/dR > 0$  for all  $R \geq 0$ .

**(ii)** Like in Exercise 8.1(i) we argue that the inequality in the maximization problem is binding and thus the Lagrangean multiplier technique applies. The Lagrangean of the problem is

$$\mathcal{L} = x_i - \frac{\lambda}{2}((\mathbf{x} - \mathbf{x}_C)^T \mathbf{H}(\mathbf{x} - \mathbf{x}_C) - R^2)$$

and thus the first order conditions give the equations

$$\mathbf{u}_i = \lambda \mathbf{H}(\mathbf{x} - \mathbf{x}_C) \tag{1}$$

$$(\mathbf{x} - \mathbf{x}_C)^T \mathbf{H}(\mathbf{x} - \mathbf{x}_C) = R^2. \tag{2}$$

Multiplying (1) by  $(\mathbf{x} - \mathbf{x}_C)^T$  from the left and using (2) we get  $\lambda = (\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_i / R^2$ . Solving (1) for  $(\mathbf{x} - \mathbf{x}_C)$  and substituting  $\lambda$  we get

$$(\mathbf{x} - \mathbf{x}_C)(\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_i = R^2(\mathbf{D}^2 \mathbf{u}_i + (1+R^2)\mathbf{D}\mathbf{e}\mathbf{e}^T \mathbf{D}\mathbf{u}_i), \tag{3}$$

where we have used equation (8.16), i.e.,  $\mathbf{H}^{-1} = \mathbf{D}(\mathbf{I}_n + (1+R^2)\mathbf{e}\mathbf{e}^T)\mathbf{D}$ . Since  $\mathbf{D}\mathbf{e} = \mathbf{x}^0$ , from (3) we have

$$(\mathbf{x} - \mathbf{x}_C)(\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_i = R^2(\mathbf{D}^2 \mathbf{u}_i + (1+R^2)\mathbf{x}^0 \mathbf{x}_i^0).$$

The  $k$ -th element  $\mathbf{u}_k^T(\mathbf{x} - \mathbf{x}_C)$  of  $(\mathbf{x} - \mathbf{x}_C)$  is

$$\mathbf{u}_k^T(\mathbf{x} - \mathbf{x}_C)(\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_i = R^2(\mathbf{u}_k^T \mathbf{D}^2 \mathbf{u}_i + (1 + R^2)x_k^0 x_i^0) . \quad (4)$$

For  $k = i$  (4) gives  $((\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_i)^2 = R^2(2 + R^2)(x_i^0)^2$  and thus

$$\begin{aligned} x_i &= x_i^C \pm Rx_i^0\sqrt{2+R^2} = x_i^C \pm \frac{R}{\sqrt{2+R^2}}x_i^C \mp \frac{R}{\sqrt{2+R^2}}x_i^C \pm Rx_i^0\sqrt{2+R^2} \\ &= \left(1 \pm \frac{R}{\sqrt{2+R^2}}\right)x_i^C \pm \frac{Rx_i^0}{\sqrt{2+R^2}}, \end{aligned} \quad (5)$$

where  $x_i^C$  is the  $i$ -th component of  $\mathbf{x}_C$ . Similarly for  $k = j \neq i$  (4) gives  $(\mathbf{x} - \mathbf{x}_C)^T \mathbf{u}_j = \pm \frac{R}{\sqrt{2+R^2}}x_j^C$  and thus

$$x_j = \left(1 \pm \frac{R}{\sqrt{2+R^2}}\right)x_j^C . \quad (6)$$

From (5) and (6) we have

$$\mathbf{x} = \left(1 \pm \frac{R}{\sqrt{2+R^2}}\right)\mathbf{x}_C \pm \frac{Rx_i^0}{\sqrt{2+R^2}}\mathbf{u}_i . \quad (7)$$

Now since  $x^0 > 0$  the plus signs in (7) give the maximum and the minus signs give the minimum value of  $x_i$ .

Since  $\mathbf{x}_C = (1 + R^2)\mathbf{x}^0$ , from (7) we have that the minimizing point is

$$\mathbf{x} = \left(1 - \frac{R}{\sqrt{2+R^2}}\right)(1 + R^2)\mathbf{x}^0 - \frac{Rx_i^0}{\sqrt{2+R^2}}\mathbf{u}_i .$$

We calculate

$$\lim_{R \rightarrow +\infty} \frac{R}{\sqrt{2+R^2}} = \lim_{R \rightarrow +\infty} \frac{1}{\sqrt{\frac{2}{R^2} + 1}} = 1$$

and

$$\begin{aligned} \lim_{R \rightarrow +\infty} \left(1 - \frac{R}{\sqrt{2+R^2}}\right)(1 + R^2) &= \lim_{R \rightarrow +\infty} \frac{1 - \frac{R}{\sqrt{2+R^2}}}{\frac{1}{\sqrt{1+R^2}}} = \lim_{R \rightarrow +\infty} \frac{-\sqrt{2+R^2} + \frac{R}{\sqrt{2+R^2}}}{-\frac{2R}{\sqrt{(1+R^2)^2(2+R^2)}}} \\ &= \lim_{R \rightarrow +\infty} \frac{(1+R^2)^2}{R(2+R^2)\sqrt{2+R^2}} = \lim_{R \rightarrow +\infty} \frac{\left(\frac{1}{R^2} + 1\right)^2}{\left(\frac{2}{R^2} + 1\right)\sqrt{\frac{2}{R^2} + 1}} = 1 . \end{aligned}$$

and therefore  $\mathbf{x} \rightarrow \mathbf{x}^0 - x_i^0 \mathbf{u}_i$  as  $R \rightarrow +\infty$ .

**(iii)** Let  $M_k = I_k + \alpha e_k e_k^T$ . We prove that  $\det M_k = 1 + k\alpha$  by induction on  $k$ . For  $k = 1$  and  $k = 2$  the assertion is trivially verified to be true. So suppose that  $\det M_k = 1 + k\alpha$  and note that

$$M_{k+1} = \begin{pmatrix} M_k & \alpha e_k \\ \alpha e_k^T & 1 + \alpha \end{pmatrix} .$$

Using the Schur complement of  $1 + \alpha$  in  $\mathbf{M}_{k+1}$  – see Chapter 2, page 29 in the book – we write  $\mathbf{M}_{k+1}$  as follows

$$\mathbf{M}_{k+1} = \begin{pmatrix} \mathbf{I}_k & \alpha \mathbf{e}_k \\ \mathbf{0}^T & 1 + \alpha \end{pmatrix} \begin{pmatrix} \mathbf{M}_k - \frac{\alpha^2}{1+\alpha} \mathbf{e}_k \mathbf{e}_k^T & \mathbf{0} \\ \frac{\alpha}{1+\alpha} \mathbf{e}_k^T & 1 \end{pmatrix}$$

and thus  $\det \mathbf{M}_{k+1} = (1 + \alpha) \det(\mathbf{M}_k - \frac{\alpha^2}{1+\alpha} \mathbf{e}_k \mathbf{e}_k^T)$ . We calculate

$$\det\left(\mathbf{M}_k - \frac{\alpha^2}{1+\alpha} \mathbf{e}_k \mathbf{e}_k^T\right) = \det\left(\mathbf{I}_k + \alpha \mathbf{e}_k \mathbf{e}_k^T - \frac{\alpha^2}{1+\alpha} \mathbf{e}_k \mathbf{e}_k^T\right) = \det\left(\mathbf{I}_k + \frac{\alpha}{1+\alpha} \mathbf{e}_k \mathbf{e}_k^T\right) = 1 + k \frac{\alpha}{1+\alpha}$$

where the last equality holds because of the induction hypothesis. So

$$\det \mathbf{M}_{k+1} = (1 + \alpha)(1 + k \frac{\alpha}{1+\alpha}) = 1 + (k+1)\alpha$$

and the induction proof is complete. For  $\alpha = 1 + R^2$  we get that

$$\det(\mathbf{I}_n + (1 + R^2) \mathbf{e} \mathbf{e}^T) = 1 + n + nR^2. \quad (8)$$

The volume of an ellipsoid is given by (7.23) as

$$vol(E) = \frac{r^n |\det \mathbf{Q}|^{\frac{1}{2}} \pi^{\frac{1}{2}}}{\Gamma(1 + \frac{n}{2})}$$

where  $E = E(\mathbf{x}_C, r) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \mathbf{x}_C)^T \mathbf{Q}^{-1} (\mathbf{x} - \mathbf{x}_C) \leq r^2\}$ . So for  $E(\mathbf{x}^0, R)$  we have

$$vol(E(\mathbf{x}^0, R)) = \frac{R^n |\det \mathbf{H}^{-1}|^{\frac{1}{2}} \pi^{\frac{1}{2}}}{\Gamma(1 + \frac{n}{2})}.$$

From (8.16)  $\mathbf{H}^{-1} = \mathbf{D}(\mathbf{I}_n + (1 + R^2) \mathbf{e} \mathbf{e}^T) \mathbf{D}$  and thus using (8) we have

$$\det \mathbf{H}^{-1} = (\det \mathbf{D})^2 \det(\mathbf{I}_n + (1 + R^2) \mathbf{e} \mathbf{e}^T) = \left(\prod_{i=1}^n x_i^0\right)^2 (1 + n + nR^2) = g^{2n}(\mathbf{x}^0) (1 + n + nR^2)$$

where  $g(\mathbf{x}^0) = (\prod_{i=1}^n x_i^0)^{\frac{1}{n}}$  is the geometric mean of  $\mathbf{x}^0$ . It follows that

$$vol(E(\mathbf{x}^0, R)) = \frac{g^n(\mathbf{x}^0) (1 + n + nR^2) R^n \pi^{\frac{1}{2}}}{\Gamma(1 + \frac{n}{2})}.$$

See Exercise 9.6(iii) for an explicit formula of the term  $\Gamma(1 + n/2)$ .

**(iv)** Since  $\mathbf{H}^{-1}$  is a positive definite matrix we have that  $\det \mathbf{H}^{-1} = \prod_{i=1}^n \lambda_i$  where  $\lambda_i > 0$  are the eigenvalues of  $\mathbf{H}^{-1}$ ; see Chapter 7.7 of the book. If  $\lambda_1$  is the smallest eigenvalue then using the formula for  $\det \mathbf{H}^{-1}$  derived in part (iii) we get

$$\lambda_1^n \leq \prod_{i=1}^n \lambda_i = g^{2n}(\mathbf{x}^0) (1 + n + nR^2)$$

and thus

$$\sqrt{\lambda_1} \leq g(\mathbf{x}^0)(1 + n + nR^2)^{\frac{1}{2n}}.$$

(v) To verify the formula, we use the following program written in Mathematica to eliminate variables  $x_3, \dots, x_6$  from

$$(\mathbf{x} - \mathbf{x}_C)^T \mathbf{H} (\mathbf{x} - \mathbf{x}_C) \leq R^2$$

using the equations  $x_3 = 250 - x_1 - 5x_2$ ,  $x_4 = 80 - x_1 - x_2$ ,  $x_5 = 180 - 3x_1 - x_2$ ,  $x_6 = 40 - 2x_1 + 3x_2$ .

```

Amat={{1,5},{1,1},{3,1},{2,-3}}
bvec={250,80,180,40}
n=6
xz={30,40}
slack=bvec-Amat.xz
xzero=Join[xz,slack]
Dmat=DiagonalMatrix[xzero]
Dinv=Inverse[Dmat]
Imat=IdentityMatrix[n]
eetr=Table[1,{i,n},{j,n}]
a=1+R^2
b=a/(1+a*n)
xc=a.xzero
H=Dinv.(Imat-b.eetr).Dinv
xv=Array[x,2]
xn=bvec-Amat.xv
x[{x1_,x2_,x3_,x4_,x5_,x6_}]:=Table[{x1,x2,x3,x4,x5,x6}]-xc
f[{x1_,x2_,x3_,x4_,x5_,x6_}]:=x[{x1,x2,x3,x4,x5,x6}].H.x[{x1,x2,x3,x4,x5,x6}]-R^2
Simplify[f[{x1,x2,250-x1-5 x2,80 -x1-x2,180-3x1-x2,40-2x1 + 3x2}]<= 0]
```

The program produces the following output

```
(356522400 + 272149200*R^2 - 4700160*x1 - 3512880*R^2*x1 + 30456*x1^2 +
24116*R^2*x1^2 - 14301000*x2 - 11431800*R^2*x2 + 71820*x1*x2 +
55188*R^2*x1*x2 + 151830*x2^2 + 125037*R^2*x2^2)/(360000*(7 + 6*R^2)) <= 0
```

which can be easily seen to be the same with the formula in the statement of the exercise; just ignore the (positive) denominator and divide throughout by 40,000.

### Exercise 8.4

(i) Show that the second derivative of  $z(R)$  is

$$\frac{d^2z}{dR^2} = \frac{((1+\beta)W - \gamma R)^2}{W^3(1+\beta+\beta R^2)^3} \left\{ 2\gamma W + R \left[ \frac{(\beta\|\mathbf{p}\|^2 + \gamma^2)(1+\beta+\beta R^2)}{1+\beta} + 2\beta W^2 \right] \right\}.$$

(ii) Show that  $z(R)$  is a convex function of  $R \in [0, \infty)$  if  $\gamma \geq 0$ ,  $z(R)$  is concave for all  $R \in [0, R_0)$  and convex for all  $R \in [R_0, \infty)$  if  $\gamma < 0$  and  $\beta \neq 0$  where  $R_0$  satisfies

$$R_0^2 = \frac{4\gamma^2 - 3(1+\beta)(\beta\|\mathbf{p}\|^2 + \gamma^2) + \sqrt{(1+\beta)(\beta\|\mathbf{p}\|^2 + \gamma^2)(9(1+\beta)(\beta\|\mathbf{p}\|^2 + \gamma^2) - 8\gamma^2)}}{6\beta(\beta\|\mathbf{p}\|^2 + \gamma^2)},$$

and  $z(R)$  is concave for all  $R \in [0, \infty)$  if  $\gamma < 0$  and  $\beta = 0$ .

(iii) Let  $z_\infty = \lim_{R \rightarrow \infty} z(R)$ . Show that  $z_\infty = z_0 - (\sqrt{(1+\beta)(\beta\|\mathbf{p}\|^2 + \gamma^2)} - \gamma)/\beta$  if  $\beta \neq 0$ , that  $z_\infty = z_0 - (\|\mathbf{p}\|^2 + \gamma^2)/2\gamma$  if  $\gamma > 0$  and  $\beta = 0$  and that  $z_\infty = -\infty$  if  $\gamma \leq 0$  and  $\beta = 0$ .

**(i)** From formula (8.18) we have that  $\frac{z_0 - z(R)}{R} = \frac{(1+\beta)W - \gamma R}{1+\beta+\beta R^2}$  and thus from (8.20) we get  $\frac{dz}{dR} = \frac{(z_0 - z(R))^2}{WR^2}$ . Also squaring (8.19) we get  $W^2 = \frac{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2}{1+\beta}$  and thus  $\frac{dW}{dR} = \frac{R(\beta\|\mathbf{p}\|^2 + \gamma^2)}{(1+\beta)W}$ . Consequently,

$$\begin{aligned} \frac{d^2z}{dR^2} &= \frac{2(z_0 - z(R))\frac{dz}{dR}WR^2 + (z_0 - z(R))^2(\frac{dW}{dR}R^2 + 2RW)}{W^2R^4} \\ &= \frac{(z_0 - z(R))^2}{W^2R^4} \left[ -2(z_0 - z(R)) + R(R\frac{dW}{dR} + 2W) \right] \\ &= \frac{(z_0 - z(R))^2}{W^3R^3} \left[ -2\frac{((1+\beta)W - \gamma R)W}{1+\beta+\beta R^2} + \frac{R^2(\beta\|\mathbf{p}\|^2 + \gamma^2)}{1+\beta} + 2W^2 \right] \\ &= \frac{((1+\beta)W - \gamma R)^2}{W^3(1+\beta+\beta R^2)^3} \left\{ 2\gamma W + R \left[ \frac{(\beta\|\mathbf{p}\|^2 + \gamma^2)(1+\beta+\beta R^2)}{1+\beta} + 2\beta W^2 \right] \right\}. \end{aligned}$$

**(ii)** First remember that by definition  $\beta \geq 0$ . If  $\gamma \geq 0$ , then from part (i) we have  $\frac{d^2z}{dR^2} > 0$  and thus  $z(R)$  is convex for all  $R \in [0, \infty)$ . If  $\gamma < 0$ , then  $\frac{d^2z}{dR^2}|_{R=0} = \frac{2\gamma}{1+\beta} < 0$ . From part (i) it follows that the sign of  $\frac{d^2z}{dR^2}$  depends upon the sign of the expression

$$\Delta(R) = 2\gamma W + R \frac{(\beta\|\mathbf{p}\|^2 + \gamma^2)(1+\beta+\beta R^2)}{1+\beta} + 2\beta RW^2.$$

Since  $\frac{\Delta(R)}{R^3} \rightarrow \frac{3\beta(\beta\|\mathbf{p}\|^2 + \gamma^2)}{1+\beta}$  for  $R \rightarrow +\infty$  it follows that  $\frac{d^2z}{dR^2}$  changes its sign in the interval  $[0, +\infty)$  if  $\beta > 0$ . If  $\beta = 0$  then  $\Delta(R) = 2\gamma W + \gamma^2 R = \gamma(2W + \gamma R) < 0$  for all  $0 \leq R < +\infty$  since  $\gamma < 0$  and for  $\beta = 0$  we have  $W = \sqrt{\|\mathbf{p}\|^2 + \gamma^2 + \gamma^2 R^2} > |\gamma|R$  and thus  $2W + \gamma R \geq (2|\gamma| + \gamma)R > 0$ . Consequently,

$z(R)$  is concave for all  $R \in [0, \infty)$  if  $\gamma < 0$  and  $\beta = 0$ . If  $\beta > 0$  and  $\gamma < 0$ , then there exist  $R \in [0, \infty)$  such that  $\Delta(R) = 0$ . From  $\Delta(R) = 0$  we have

$$R[(\beta\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2) + 2\beta(1 + \beta)W^2] = -2\gamma(1 + \beta)W$$

and squaring both sides we get using (8.19) that

$$\begin{aligned} & R^2[(\beta\|\mathbf{p}\|^2 + \gamma^2)^2(1 + \beta + \beta R^2)^2 + 4\beta(\beta\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2)((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2)) \\ & \quad + 4\beta^2((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2)^2] \\ & = 4\gamma^2(1 + \beta)((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2). \end{aligned}$$

Rearranging the right-hand side of this equation we obtain

$$4\gamma^2(((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2) + \gamma^2 R^2).$$

Subtracting  $4\gamma^4 R^2$  from both sides of the equation and using

$$\begin{aligned} & 4\beta^2((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2)^2 - 4\gamma^4 \\ & = 4(\beta\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2)[\beta((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2) - \gamma^2], \end{aligned}$$

we find by clearing  $1 + \beta + \beta R^2 > 0$  that  $\Delta(R) = 0$  if and only if

$$R^2(\beta\|\mathbf{p}\|^2 + \gamma^2)[9(\beta\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2) - 12\gamma^2] = 4\gamma^2((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2),$$

where we have repeatedly used the identity

$$\beta[(1 + \beta)\|\mathbf{p}\|^2 + \gamma^2 + (\beta\|\mathbf{p}\|^2 + \gamma^2)R^2] = (\beta\|\mathbf{p}\|^2 + \gamma^2)(1 + \beta + \beta R^2) - \gamma^2.$$

We thus obtain a quadratic equation in  $R^2$  which has at most one positive root  $R_0^2$ . Solving this equation we find the value of  $R_0^2$  given in the exercise. Rearranging the expression given there for  $R_0^2$  we have

$$R_0^2 = \frac{\gamma^2 - 3\beta((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2) + \sqrt{(9\beta((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2) + \gamma^2)(\beta((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2) + \gamma^2)}}{6\beta(\beta\|\mathbf{p}\|^2 + \gamma^2)}.$$

Since the square root is greater than  $3\beta((1 + \beta)\|\mathbf{p}\|^2 + \gamma^2)$  we thus have  $R_0^2 > 0$  and hence there exists exactly one positive  $R_0 \in [0, \infty)$  where  $\Delta(R)$  changes its sign. Consequently,  $z(R)$  is concave for all  $R \in [0, R_0]$  and convex for all  $R \in [R_0, \infty)$  if  $\gamma < 0$  and  $\beta > 0$ .

**(iii)** If  $\beta > 0$ , then we write

$$z(R) = z_0 - \frac{(1 + \beta)W/R - \gamma}{(1 + \beta)/R^2 + \beta}.$$

Now  $\lim_{R \rightarrow +\infty} W/R = \sqrt{(\beta\|\mathbf{p}\|^2 + \gamma^2)/(1 + \beta)}$  and thus the assertion follows. If  $\beta = 0$  then from (8.19) we get  $W = \sqrt{\|\mathbf{p}\|^2 + \gamma^2 + \gamma^2 R^2}$  and thus  $z(R) = z_0 - R(W - \gamma R) = z_0 - \frac{\|\mathbf{p}\|^2 + \gamma^2}{W/R + \gamma}$ . It follows that  $z_\infty = z_0 - (\|\mathbf{p}\|^2 + \gamma^2)/2\gamma$  if  $\gamma > 0$  since  $\sqrt{\gamma^2} = \gamma$  in this case. If  $\gamma \leq 0$  then  $z_\infty = -\infty$  since  $\|\mathbf{p}\|^2 > 0$  by

the nonoptimality of  $\mathbf{x}^0$  and  $W/R + \gamma = \sqrt{(\|\mathbf{p}\|^2 + \gamma^2)/R^2 + \gamma^2} + \gamma > 0$  for all  $R$ , but  $\lim_{R \rightarrow +\infty} W/R + \gamma = 0$ .

---

### Exercise 8.5

Assume that  $\mathbf{p}$  and  $\mathbf{d}$  are linearly independent and  $s \geq 0$ .

- (i) Show that  $\mathbf{v}(s)$  solves the problem  $\min\{\mathbf{y}_{n+1} : (\mathbf{cD}, -z_0)\mathbf{y} = \mathbf{0}, \mathbf{y} \in T_0(\mathcal{X}) \cap B_s^{n+1}\}$ .
  - (ii) Show that  $\mathbf{u}(s)$  solves the problem  $\min\{(\mathbf{cD}, -z_0)\mathbf{y} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_s^{n+1}\}$ .
- 

**(i)** We will apply Remark 8.1 to the following optimization problem

$$\min\{(\mathbf{0}, 1)\mathbf{y} : (\mathbf{AD}, -\mathbf{b})\mathbf{y} = \mathbf{0}, (\mathbf{cD}, -z_0)\mathbf{y} = 0, \mathbf{y} \in B_s^{n+1}\}.$$

To this end, we have to calculate the orthogonal projection of  $(\mathbf{0}, 1)$  on the subspace

$$\{\mathbf{y} \in \mathbb{R}^{n+1} : (\mathbf{AD}, -\mathbf{b})\mathbf{y} = \mathbf{0}, (\mathbf{cD}, -z_0)\mathbf{y} = 0, \mathbf{f}^T \mathbf{y} = 0\} \quad (1)$$

where, as usual,  $\mathbf{f} \in \mathbb{R}^{n+1}$  is the vector with all components equal to one. Let, for notational convenience,

$$\mathbf{E} = \begin{pmatrix} \mathbf{AD} & -\mathbf{b} \\ \mathbf{e}^T & 1 \\ \mathbf{cD} & -z_0 \end{pmatrix}.$$

First we compute the inverse of the matrix

$$\widehat{\mathbf{A}} = \mathbf{EE}^T = \begin{pmatrix} \mathbf{AD}^2 \mathbf{A}^T + \mathbf{bb}^T & \mathbf{0} & \mathbf{AD}^2 \mathbf{c}^T + z_0 \mathbf{b} \\ \mathbf{0} & n+1 & 0 \\ \mathbf{cD}^2 \mathbf{A}^T + z_0 \mathbf{b}^T & \mathbf{0}^T & \mathbf{cD}^2 \mathbf{c}^T + z_0^2 \end{pmatrix}.$$

Let  $\widehat{\mathbf{C}} = (\mathbf{cD}^2 \mathbf{A}^T + z_0 \mathbf{b}^T \ \mathbf{0}^T)$ ,  $\widehat{\mathbf{e}} = (\mathbf{cD}^2 \mathbf{c}^T + z_0^2)$  and

$$\widehat{\mathbf{B}} = \begin{pmatrix} \mathbf{AD}^2 \mathbf{A}^T + \mathbf{bb}^T & \mathbf{0} \\ \mathbf{0} & n+1 \end{pmatrix}, \quad \widehat{\mathbf{d}} = \begin{pmatrix} \mathbf{AD}^2 \mathbf{c}^T + z_0 \mathbf{b} \\ 0 \end{pmatrix}.$$

The matrix  $\widehat{\mathbf{B}}$  is the same as  $\widehat{\mathbf{G}}$  of Chapter 8.1.1 (page 220 in the book) and thus

$$\widehat{\mathbf{B}}^{-1} = \begin{pmatrix} \mathbf{G}^{-1} - \frac{1}{1+\beta} \mathbf{G}^{-1} \mathbf{bb}^T \mathbf{G}^{-1} & \mathbf{0} \\ \mathbf{0}^T & \frac{1}{n+1} \end{pmatrix}.$$

Applying the *involution formula for partitioned matrices* of Chapter 2.2 (page 29 in the book) we find

$$\widehat{\mathbf{A}}^{-1} = \begin{pmatrix} \widehat{\mathbf{B}}^{-1} + \widehat{\mathbf{B}}^{-1} \widehat{\mathbf{D}} \mathbf{F} \widehat{\mathbf{C}} \widehat{\mathbf{B}}^{-1} & -\widehat{\mathbf{B}}^{-1} \widehat{\mathbf{D}} \mathbf{F}^{-1} \\ -\mathbf{F}^{-1} \widehat{\mathbf{C}} \widehat{\mathbf{B}}^{-1} & \mathbf{F}^{-1} \end{pmatrix}$$

where  $\mathbf{F} = \widehat{\mathbf{E}} - \widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1}\widehat{\mathbf{D}}$ . Developing  $\mathbf{F}$  and substituting

$$cD^2\mathbf{A}^T\mathbf{G}^{-1}\mathbf{b} = z_0 - \gamma, \quad \mathbf{b}^T\mathbf{G}^{-1}\mathbf{b} = \beta, \quad \mathbf{D}\mathbf{A}^T\mathbf{G}^{-1}\mathbf{A}\mathbf{D} = \mathbf{I} - \mathbf{P}, \quad \text{and} \quad cD\mathbf{p} = \|\mathbf{p}\|^2$$

we get  $\mathbf{F} = \frac{\gamma^2}{1+\beta} + \|\mathbf{p}\|^2$ , i.e.,  $\mathbf{F}$  is a scalar, and thus  $\mathbf{F}^{-1} = \frac{1+\beta}{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2}$ . We can factor out  $\mathbf{F}^{-1}$  from the inversion formula and calculate  $\widehat{\mathbf{B}}^{-1}\widehat{\mathbf{D}}\widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1}$ . Multiplying the matrices and substituting  $\mathbf{b}^T\mathbf{G}^{-1}\mathbf{b} = \beta$  and  $cD^2\mathbf{A}^T\mathbf{G}^{-1}\mathbf{b} = z_0 - \gamma$  we get

$$\widehat{\mathbf{B}}^{-1}\widehat{\mathbf{D}}\widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1} = \begin{pmatrix} \Theta + \frac{\gamma}{1+\beta}\Phi + \frac{\gamma}{1+\beta}\Phi^T + \left(\frac{\gamma}{1+\beta}\mathbf{G}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{G}^{-1}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where  $\Theta = \mathbf{G}^{-1}\mathbf{A}\mathbf{D}^2\mathbf{c}^T\mathbf{c}\mathbf{D}^2\mathbf{A}^T\mathbf{G}^{-1}$  and  $\Phi = \mathbf{G}^{-1}\mathbf{A}\mathbf{D}^2\mathbf{c}^T\mathbf{b}^T\mathbf{G}^{-1}$ , and therefore

$$\begin{aligned} \mathbf{F}\mathbf{B}^{-1} + \widehat{\mathbf{B}}^{-1}\widehat{\mathbf{D}}\widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1} = \\ \begin{pmatrix} \|\mathbf{p}\|^2 + \frac{\gamma^2}{1+\beta}\mathbf{G}^{-1} + \Theta + \frac{\gamma}{1+\beta}\Phi + \frac{\gamma}{1+\beta}\Phi^T - \frac{\|\mathbf{p}\|^2}{1+\beta}\mathbf{G}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{G}^{-1} & \mathbf{0} \\ \mathbf{0} & \frac{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2}{(1+\beta)(n+1)} \end{pmatrix}. \end{aligned}$$

We calculate  $-\widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1}$  and after substituting  $cD^2\mathbf{A}^T\mathbf{G}^{-1}\mathbf{b} = z_0 - \gamma$  we get

$$-\widehat{\mathbf{C}}\widehat{\mathbf{B}}^{-1} = \left(-cD^2\mathbf{A}^T\mathbf{G}^{-1} - \frac{\gamma}{1+\beta}\mathbf{b}^T\mathbf{G}^{-1} \quad \mathbf{0}\right).$$

Since  $\widehat{\mathbf{D}} = \widehat{\mathbf{C}}^T$  we calculate the inverse  $\widehat{\mathbf{A}}^{-1}$  of  $\widehat{\mathbf{A}}$  to be equal to

$$\begin{pmatrix} \|\mathbf{p}\|^2 + \frac{\gamma^2}{1+\beta}\mathbf{G}^{-1} + \Theta + \frac{\gamma}{1+\beta}\Phi + \frac{\gamma}{1+\beta}\Phi^T - \frac{\|\mathbf{p}\|^2}{1+\beta}\mathbf{G}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{G}^{-1} & \mathbf{0} & -\mathbf{G}^{-1}\mathbf{A}\mathbf{D}^2\mathbf{c}^T - \frac{\gamma}{1+\beta}\mathbf{G}^{-1}\mathbf{b} \\ \mathbf{0} & \frac{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2}{(1+\beta)(n+1)} & \mathbf{0} \\ -cD^2\mathbf{A}^T\mathbf{G}^{-1} - \frac{\gamma}{1+\beta}\mathbf{b}^T\mathbf{G}^{-1} & 0 & 1 \end{pmatrix}$$

Now we can calculate the projection operator  $Q$  on the subspace (1) as

$$Q = \mathbf{I}_{n+1} - \mathbf{E}^T\widehat{\mathbf{A}}^{-1}\mathbf{E} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} - \mathbf{F}^{-1} \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}.$$

Using  $\mathbf{b}^T\mathbf{G}^{-1}\mathbf{A}\mathbf{D} = \mathbf{e}^T - \mathbf{d}^T$ ,  $\mathbf{P}\mathbf{D}\mathbf{c}^T = \mathbf{p}$  and  $\mathbf{D}\mathbf{A}^T\mathbf{G}^{-1}\mathbf{A}\mathbf{D} = \mathbf{I} - \mathbf{P}$ , we calculate

$$E_{11} = \left(\|\mathbf{p}\| + \frac{\gamma^2}{1+\beta}\right)(\mathbf{I} - \mathbf{P}) + \mathbf{p}\mathbf{p}^T - \frac{\gamma}{1+\beta}(\mathbf{p}(\mathbf{e} - \mathbf{d})^T + (\mathbf{e} - \mathbf{d})\mathbf{p}^T) - \frac{\|\mathbf{p}\|^2}{1+\beta}(\mathbf{e} - \mathbf{d})(\mathbf{e} - \mathbf{d})^T + \frac{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2}{(1+\beta)(n+1)}\mathbf{e}\mathbf{e}^T,$$

$$E_{12} = -\frac{\|\mathbf{p}\|^2}{1+\beta}(\mathbf{e} - \mathbf{d}) - \frac{\gamma}{1+\beta}\mathbf{p} + \frac{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2}{(1+\beta)(n+1)}\mathbf{e},$$

$$E_{21} = -\frac{\|\mathbf{p}\|^2}{1+\beta}(\mathbf{e} - \mathbf{d})^T - \frac{\gamma}{1+\beta}\mathbf{p}^T + \frac{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2}{(1+\beta)(n+1)}\mathbf{e}^T,$$

$$E_{22} = \frac{\beta\|\mathbf{p}\|^2 + \gamma^2}{1+\beta} + \frac{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2}{(1+\beta)(n+1)}.$$

It follows that the projection of the vector  $(0, 1)$  on the subspace (1) is

$$\mathbf{q} = \mathbf{Q} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} -\mathbf{F}^{-1}E_{12} \\ 1 - \mathbf{F}^{-1}E_{22} \end{pmatrix}.$$

We calculate

$$\begin{aligned} -\mathbf{F}^{-1}E_{12} &= \frac{1}{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2} \left( \gamma\mathbf{p} - \|\mathbf{p}\|^2\mathbf{d} + \frac{\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2}{n+1}\mathbf{e} \right), \\ 1 - \mathbf{F}^{-1}E_{22} &= \frac{\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2}{(n+1)(\gamma^2 + (1+\beta)\|\mathbf{p}\|^2)}, \end{aligned}$$

where we have used  $n-\beta = \|\mathbf{d}\|^2$ . Therefore we have

$$\mathbf{q} = \frac{1}{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2} \left( \frac{\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2}{n+1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} + \begin{pmatrix} \gamma\mathbf{p} - \|\mathbf{p}\|^2\mathbf{d} \\ 0 \end{pmatrix} \right)$$

$$\text{and } \|\mathbf{q}\|^2 = \frac{\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2}{(n+1)(\gamma^2 + (1+\beta)\|\mathbf{p}\|^2)}.$$

From Remark 8.1 and the definition of  $\mathbf{u}$ ,  $\mathbf{v}$  and  $\mathbf{v}(s)$ , see pages 235-236 in the book, we have that the optimal solution to our optimization problem is given by

$$\begin{aligned} \mathbf{y} &= \mathbf{y}^0 - s \frac{\mathbf{q}}{\|\mathbf{q}\|} = \mathbf{y}^0 + s \frac{\mathbf{v}\sqrt{n+1}}{\sqrt{\gamma^2 + (1+\beta)\|\mathbf{p}\|^2} \sqrt{\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2}} \\ &= \mathbf{y}^0 + s \frac{\mathbf{v}}{\sqrt{\|\mathbf{p}\|^2 + \frac{\gamma^2}{1+\beta}} \sqrt{(1+\beta)(\|\mathbf{d}\|^2\|\mathbf{p}\|^2 - \gamma^2)/(n+1)}} \\ &= \mathbf{y}^0 + s \frac{\mathbf{v}}{\|\mathbf{v}\|} = \mathbf{v}(s). \end{aligned}$$

**(ii)** The optimization problem is the same as  $(ALP_\rho)$  except that the objective function vector is  $(cD, -z_0)$  rather than  $(cD, 0)$ . So the solution to the problem is given by (8.7) which with the necessary changes is

$$\mathbf{y} = \mathbf{y}^0 - s \frac{\mathbf{q}}{\|\mathbf{q}\|}$$

where  $\mathbf{q} = \mathbf{Q} \begin{pmatrix} Dc^T \\ -z_0 \end{pmatrix}$  and  $\mathbf{Q}$  is the projection operator given in Chapter 8.1.1, i.e.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \frac{1}{1+\beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{e}^T - \mathbf{d}^T \\ 1 \end{pmatrix} - \frac{1}{n+1} \mathbf{f} \mathbf{f}^T.$$

We calculate

$$\mathbf{q} = \begin{pmatrix} PDc^T \\ 0 \end{pmatrix} + \frac{1}{1+\beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{e}^T Dc^T - \mathbf{d}^T Dc^T \\ -z_0 \end{pmatrix} - \frac{\mathbf{e}^T Dc^T - z_0}{n+1} \mathbf{f}$$

and since  $e^T D c^T = c D e = c x^0 = z_0$ ,  $p = P D c^T$ , and  $d^T D c^T = c D d = \gamma$  we get

$$\mathbf{q} = \begin{pmatrix} p \\ 0 \end{pmatrix} - \frac{\gamma}{1+\beta} \begin{pmatrix} e - d \\ 1 \end{pmatrix}.$$

Thus with  $\mathbf{u} = -\mathbf{q}$  we have

$$\mathbf{y} = \mathbf{y}^0 + s \frac{\mathbf{u}}{\|\mathbf{u}\|}.$$


---

### Exercise 8.6

- (i) Let  $\Gamma = \gamma(n+1)/((1+\beta)\|\mathbf{p}\|^2 + \gamma^2)$ . Show that  $(\Gamma c D - e^T, -z)\mathbf{u}^\infty = 0$  for all  $z \in \mathbb{R}$ . Show that the optimal solution  $\tilde{\mathbf{y}}(\rho)$  and the corresponding objective function value  $\tilde{z}(\rho)$  of the problem  $\min\{(\Gamma c D - e^T, 0)\mathbf{y}/y_{n+1} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}\}$  are given by

$$\begin{aligned}\tilde{\mathbf{y}}(\rho) &= \mathbf{y}^0 + \rho \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} \frac{\mathbf{w}}{\|\mathbf{w}\|} + \frac{\rho^2}{\|\mathbf{u}^0\|^2} \mathbf{u}^0, \\ \tilde{z}(\rho) &= \Gamma z_0 - n - \frac{(n+1)\rho}{\|\mathbf{w}\| \sqrt{1 - \rho^2/\|\mathbf{u}^0\|^2} - \rho}\end{aligned}$$

for all  $0 \leq \rho < \rho_\infty$ . Give a geometric interpretation of  $\tilde{\mathbf{y}}(\rho)$  similar to the one of  $\mathbf{y}(\rho)$ , see Figures 8.4 and 8.5.

- (ii) Show that  $\mathbf{y}^{\max}(\rho) = \mathbf{y}^0 + \rho(-\sqrt{1 - \rho^2/\|\mathbf{w}\|^2} \mathbf{u}/\|\mathbf{u}\| + \rho \mathbf{w}/\|\mathbf{w}\|^2)$  with an objective function value of  $z^{\max}(\rho) = z_0 + \rho(n+1)\|\mathbf{u}\|^2/(\|\mathbf{u}\| \sqrt{1 - \rho^2/\|\mathbf{w}\|^2} - (n+1)\gamma\rho/(1+\beta))$  is the maximizer for  $(FLP_\rho)$ . Moreover,  $z^{\max}(\rho) - z(\rho) = 2\|\mathbf{u}\|\rho(n+1)\sqrt{1 - \rho^2/\|\mathbf{w}\|^2}/(1 - \rho^2/\rho_\infty^2)$  where  $z(\rho)$  is given in (8.37). (Hint: Apply Remark 8.6.)
- 

- (i) We assume throughout that  $\gamma = \mathbf{p}^T \mathbf{d} \neq 0$  since otherwise the point  $\mathbf{u}^\infty$  does not exist; see (8.34). Moreover,  $u_{n+1}^\infty = 0$  since  $\mathbf{u}^\infty$  is an improper point of  $\mathcal{P}^n$ . From the definition (8.34) of  $\mathbf{u}^\infty$  and (8.28) of  $\mathbf{u}$  we calculate

$$\begin{aligned}(\Gamma c D - e^T, -z)\mathbf{u}^\infty &= \frac{1+\beta}{\gamma(n+1)} (\Gamma c D - e^T) \mathbf{p} + \frac{1}{n+1} (\Gamma c D - e^T) \mathbf{d} \\ &= \frac{1+\beta}{\gamma(n+1)} (\Gamma \|\mathbf{p}\|^2 - \gamma) + \frac{1}{n+1} (\Gamma \gamma - n + \beta) = \frac{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2}{(1+\beta)\|\mathbf{p}\|^2 + \gamma^2} - \frac{n+1}{n+1} = 0\end{aligned}$$

for all  $z \in \mathbb{R}$  as claimed, where we have used

$$c D \mathbf{p} = \|\mathbf{p}\|^2, \quad e^T \mathbf{p} = c D \mathbf{p} = \gamma, \quad \text{and } e^T \mathbf{d} = \|\mathbf{d}\|^2 = n - \beta.$$

To prove that  $\tilde{\mathbf{y}}(\rho)$  solves the minimization problem we proceed as we do in the proof of Remark 8.6. By a direct calculation we verify that  $\tilde{\mathbf{y}}(\rho)$  is a feasible solution to the problem, i.e., that

$$(AD, -\mathbf{b})\tilde{\mathbf{y}}(\rho) = \mathbf{0}, \quad f^T \tilde{\mathbf{y}}(\rho) = 1, \quad \text{and } \|\tilde{\mathbf{y}}(\rho) - \mathbf{y}^0\|^2 = \rho^2,$$

and note that  $\tilde{\mathbf{y}}(\rho)$  is well-defined in real terms for all  $0 \leq \rho \leq \|\mathbf{u}^0\|$ . Moreover, by (8.34) and (8.35)

$$\|\mathbf{u}^0\|^2 = \frac{(1+\beta)^2(\|\mathbf{p}\|^2 + \gamma^2/(1+\beta))}{\gamma^2(n+1)^2}, \quad \rho_\infty^2 = \frac{1+\beta}{(n+1)(n-\beta)},$$

and thus  $\rho_\infty \leq \|\mathbf{u}^0\|$  which follows from the Cauchy-Schwarz inequality – see point 7.1(a) – when applied to  $\mathbf{p}$  and  $\mathbf{d}$ . To derive the expression for the objective function  $\tilde{z}(\rho)$  we write the  $(n+1)$ -st component  $\tilde{y}_{n+1}(\rho)$  of  $\tilde{\mathbf{y}}(\rho)$  as follows

$$\tilde{y}_{n+1}(\rho) = \frac{1 - \frac{\rho}{\|\mathbf{w}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} - \frac{\rho^2}{\|\mathbf{u}^0\|^2}}{n+1} = \frac{\sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} (\|\mathbf{w}\| \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} - \rho)}{(n+1)\|\mathbf{w}\|}$$

and verify that  $\tilde{y}_{n+1}(\rho) > 0$  for all  $0 \leq \rho < \rho_\infty$ . We calculate furthermore

$$\begin{aligned} (\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, 0) \tilde{\mathbf{y}}(\rho) &= (\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T) \left[ \frac{1}{n+1} \mathbf{e} + \frac{\rho}{\|\mathbf{w}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} \left( -\frac{1}{n+1} \mathbf{e} - \frac{\gamma \mathbf{p} - \|\mathbf{p}\|^2 \mathbf{d}}{\|\mathbf{p}\|^2 \|\mathbf{d}\|^2 - \gamma^2} \right) \right. \\ &\quad \left. - \frac{\rho^2}{\|\mathbf{u}^0\|^2} \frac{1+\beta}{\gamma(n+1)} (-\mathbf{p} + \frac{\gamma}{1+\beta} (\mathbf{e} - \mathbf{d})) \right] \\ &= (\Gamma z_0 - n) \tilde{y}_{n+1}(\rho) - \frac{\rho}{\|\mathbf{w}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}}. \end{aligned}$$

Consequently, we get the following expression for  $\tilde{z}(\rho)$

$$\tilde{z}(\rho) = \frac{(\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, 0) \tilde{\mathbf{y}}(\rho)}{\tilde{y}_{n+1}(\rho)} = \Gamma z_0 - n - \frac{(n+1)\rho}{\|\mathbf{w}\| \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} - \rho}.$$

Calculating the first derivative of  $\tilde{z}(\rho)$  we find

$$\frac{d\tilde{z}(\rho)}{d\rho} = -\frac{(n+1)\|\mathbf{w}\|}{\sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} \left( \|\mathbf{w}\| \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} - \rho \right)^2} < 0$$

and thus  $\tilde{z}(\rho)$  decreases monotonically for all  $0 \leq \rho < \rho_\infty$ . Using the projection operator  $\mathbf{Q}$  of Chapter 8.1.1 we calculate the orthoprojection of  $(\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, -\tilde{z}(\rho))$  on the nullspace (8.3) to be

$$\tilde{\mathbf{q}}(\rho) = \mathbf{Q} \begin{pmatrix} \Gamma \mathbf{D} \mathbf{c}^T - \mathbf{e} \\ -\tilde{z}(\rho) \end{pmatrix} = \begin{pmatrix} \Gamma \mathbf{p} - \mathbf{d} \\ 0 \end{pmatrix} + \frac{\tilde{z}_0 - \tilde{z}(\rho) - \Gamma \gamma + \|\mathbf{d}\|^2}{1+\beta} \begin{pmatrix} \mathbf{e} - \mathbf{d} \\ 1 \end{pmatrix} - \frac{\tilde{z}_0 - \tilde{z}(\rho)}{n+1} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix}$$

where  $\tilde{z}_0 = \Gamma z_0 - n$ . We calculate next from the definitions:

$$\begin{aligned}
\|\tilde{\mathbf{q}}(\rho)\|^2 &= (\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, -\tilde{z}(\rho)) \tilde{\mathbf{q}}(\rho) \\
&= \Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{(\Gamma\gamma - \|\mathbf{d}\|^2)^2}{1+\beta} - \frac{2}{1+\beta} (\tilde{z}_0 - \tilde{z}(\rho))(\Gamma\gamma - \|\mathbf{d}\|^2) + \frac{\|\mathbf{d}\|^2(\tilde{z}_0 - \tilde{z}(\rho))^2}{(1+\beta)(n+1)} \\
&= \theta^{-2}(\rho) \left\{ \left[ \Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{(\Gamma\gamma - \|\mathbf{d}\|^2)^2}{1+\beta} \right] \theta^2(\rho) \right. \\
&\quad \left. - \frac{2(n+1)\rho}{1+\beta} (\Gamma\gamma - \|\mathbf{d}\|^2) \theta(\rho) + \frac{\|\mathbf{d}\|^2}{(1+\beta)(n+1)} (n+1)^2 \rho^2 \right\} \\
&= \theta^{-2}(\rho) \left\{ \|\mathbf{w}\|^2 \left( \Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{(\Gamma\gamma - \|\mathbf{d}\|^2)^2}{1+\beta} \right) \right. \\
&\quad \left. - 2\rho \|\mathbf{w}\| \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} \left[ \Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{(\Gamma\gamma - \|\mathbf{d}\|^2)^2}{1+\beta} + \frac{n+1}{1+\beta} (\Gamma\gamma - \|\mathbf{d}\|^2) \right] \right. \\
&\quad \left. + \rho^2 \left[ \left( 1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2} \right) \left( \Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{\Gamma\gamma - \|\mathbf{d}\|^2}{1+\beta} \right) \right. \right. \\
&\quad \left. \left. + \frac{2(n+1)}{1+\beta} (\Gamma\gamma - \|\mathbf{d}\|^2) + \frac{(n+1)\|\mathbf{d}\|^2}{1+\beta} \right] \right\} \\
&= \theta^{-2}(\rho) = \left( \frac{\tilde{z}_0 - \tilde{z}(\rho)}{(n+1)\rho} \right)^2,
\end{aligned}$$

where we have set  $\theta(\rho) = \|\mathbf{w}\| \sqrt{1 - \frac{\rho^2}{\|\mathbf{u}^0\|^2}} - \rho$  and we have used repeatedly the following identities that are readily verified:

$$\Gamma^2 \|\mathbf{p}\|^2 - 2\Gamma\gamma + \|\mathbf{d}\|^2 + \frac{(\Gamma\gamma - \|\mathbf{d}\|^2)^2}{1+\beta} + \frac{n+1}{1+\beta} (\Gamma\gamma - \|\mathbf{d}\|^2) = 0, \quad \frac{n+1}{1+\beta} (\Gamma\gamma - \|\mathbf{d}\|^2) = -\frac{1}{\|\mathbf{w}\|^2}.$$

It follows that  $\tilde{z}(\rho) = \tilde{z}_0 - (n+1)\rho \|\tilde{\mathbf{q}}(\rho)\|$  and thus  $\|\tilde{\mathbf{q}}(\rho)\| \neq 0$  for  $0 \leq \rho < \rho_\infty$ , because of the monotonicity of  $\tilde{z}(\rho)$ . To prove that  $\tilde{z}(\rho)$  is an optimal solution to the problem  $\min\{(\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, 0) \mathbf{y} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}\}$  for  $0 \leq \rho < \rho_\infty$ , we proceed – with the necessary changes – exactly as done on page 242 in the proof of Remark 8.6 and thus the assertion follows. To give a geometric interpretation to the solution, we calculate  $\|\tilde{\mathbf{y}}(\rho) - \frac{1}{2}(\mathbf{y}^0 + \mathbf{u}^\infty)\| = \|\mathbf{u}^0\|/2$  and thus  $\tilde{\mathbf{y}}(\rho)$  for  $0 \leq \rho < \rho_\infty$  lies on (a segment of) the semi-circle with center  $\frac{1}{2}(\mathbf{y}^0 + \mathbf{u}^\infty)$  that passes through  $\mathbf{y}^0$  and  $\mathbf{u}^\infty$ . By the first part of this exercise  $(\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, -z) \mathbf{u}^\infty = 0$  for all  $z \in \mathbb{R}$  and thus in particular, for all  $z = \tilde{z}(\rho)$ . Thus for a geometric interpretation of the solution to our problem all we have to do in Figures 8.4 and 8.5 is to interchange  $\mathbf{u}^\infty$  and  $\mathbf{w}^\infty$  and replace  $(\mathbf{c} \mathbf{D}, -z(\rho)) \mathbf{y} = 0$  by  $(\Gamma \mathbf{c} \mathbf{D} - \mathbf{e}^T, -\tilde{z}(\rho)) \mathbf{y} = 0$ , etc. Note that the problem and its solution depend on the sign of  $\gamma = \mathbf{p}^T \mathbf{d}$ .

**(ii)** We show that the assertions are true for all  $0 \leq \rho < \rho_\infty$  and leave a more detailed analysis to the reader. By (8.36) we have  $y_{n+1} > 0$  for all  $\mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1}$  and thus the maximization problem has a finite optimum solution since the feasible set is a compact subset of  $\mathbb{R}^{n+1}$ . Consequently, for all  $0 \leq \rho < \rho_\infty$

$$\max\left\{ \frac{(\mathbf{c} \mathbf{D}, 0) \mathbf{y}}{y_{n+1}} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1} \right\} = -\min\left\{ \frac{(-\mathbf{c} \mathbf{D}, 0) \mathbf{y}}{y_{n+1}} : \mathbf{y} \in T_0(\mathcal{X}) \cap B_\rho^{n+1} \right\}$$

and we can apply Remark 8.6 directly with the changes that result from the sign change in the objective function. It follows that  $\mathbf{p}$  must be replaced by  $-\mathbf{p}$ ,  $\gamma$  by  $-\gamma$ , and  $z_0$  by  $-z_0$ . By consequence, the vector  $\mathbf{u}$  is replaced by  $-\mathbf{u}$ , whereas the sign change does not affect the vector  $\mathbf{w}$ . Consequently, we find that the optimal solution to the maximization problem is given by

$$\mathbf{y}^{\max}(\rho) = \mathbf{y}^0 - \rho \sqrt{1 - \frac{\rho^2}{\|\mathbf{w}\|^2}} \frac{\mathbf{u}}{\|\mathbf{u}\|} + \frac{\rho^2}{\|\mathbf{w}\|^2} \mathbf{w}$$

and using the above identity we find from (8.37)

$$z^{\max}(\rho) = - \left( -z_0 - \frac{(n+1)\|\mathbf{u}\|\rho}{\sqrt{1 - \frac{\rho^2}{\|\mathbf{w}\|^2} - \frac{(n+1)\gamma\rho}{(1+\beta)\|\mathbf{u}\|}}} \right)$$

as we have asserted. Note that the denominator in  $z^{\max}(\rho)$  is positive for all  $0 \leq \rho < \rho_\infty$  no matter whether  $\gamma > 0$  or  $\gamma \leq 0$ .

We calculate  $\|\mathbf{y}^{\max}(\rho) - \frac{1}{2}(\mathbf{y}^0 + \mathbf{w}^\infty)\| = \|\mathbf{w}\|/2$  and thus the loci of the solution vector  $\mathbf{y}^{\max}(\rho)$  form a (segment of the) semi-circle around  $\frac{1}{2}(\mathbf{y}^0 + \mathbf{w}^\infty)$  with radius  $\|\mathbf{w}\|/2$  (for  $0 \leq \rho < \rho_\infty$ ) as claimed and thus we get the “other half” of the semi-circle formed by the solution to  $(\text{FLP}_\rho)$ . The formula for the difference  $z^{\max}(\rho) - z(\rho)$  follows by a straightforward calculation using the definition of  $\rho_\infty$  and the simplifications that impose themselves.

### Exercise 8.7

Let  $\mathbf{x}^0 \in \mathcal{X}$ ,  $\mathbf{x}^0 > 0$  be nonoptimal and  $\mathbf{c} \in \mathbb{R}^n$  be such that  $\gamma = \mathbf{p}^T \mathbf{d} > 0$  and  $\|\mathbf{w}\| > \rho_* = \sqrt{n/(n+1)}$  where  $\mathbf{w}$  is defined in (8.30). Show that  $\mathbf{c}\mathbf{x} \geq z_0 - (1+\beta)\|\mathbf{u}\|^2/\gamma$  for all  $\mathbf{x} \in \mathcal{X}$ . (Hint: Apply the construction of the proof of Remark 8.7 e.g. with  $\sigma = (\|\mathbf{w}\| + \rho_*)/2$  and  $\rho = 1/\sigma(n+1)$ .)

Since  $\gamma = \mathbf{p}^T \mathbf{d} > 0$  it follows from Remark 8.6(i) that  $(\text{FLP}_\rho)$  has an optimal solution for all  $0 \leq \rho \leq \|\mathbf{w}\|$ . Since  $\sigma = (\|\mathbf{w}\| + \rho_*)/2 < \|\mathbf{w}\|$ , it follows by our assumption that  $\rho_* < \|\mathbf{w}\|$ , that an optimal solution  $\mathbf{y}(\sigma)$  to  $(\text{FLP}_\sigma)$  exists. Moreover, the corresponding optimal solution  $\mathbf{y}(\sigma)$  and its objective function value  $z(\sigma)$  are given by (8.32) and (8.37), respectively, with  $\rho$  replaced by  $\sigma$ . From Remark 8.7 it follows that  $z(\sigma)$  is a lower bound on the objective function of (LP) and thus  $\mathbf{c}\mathbf{x} \geq z(\sigma)$  for all  $\mathbf{x} \in \mathcal{X}$ . To prove the assertion we calculate:

$$\begin{aligned} z(\sigma) &= z\left(\frac{1}{2}(\|\mathbf{w}\| + \rho_*)\right) = z_0 - \frac{1}{2} \frac{(n+1)\|\mathbf{u}\|(\|\mathbf{w}\| + \rho_*)}{\sqrt{1 - \frac{(\|\mathbf{w}\| + \rho_*)^2}{4\|\mathbf{w}\|^2} + \frac{(n+1)\gamma(\|\mathbf{w}\| + \rho_*)}{2(1+\beta)\|\mathbf{u}\|}}} \\ &\geq z_0 - \frac{1}{2} \frac{(n+1)\|\mathbf{u}\|(\|\mathbf{w}\| + \rho_*)}{\frac{(n+1)\gamma(\|\mathbf{w}\| + \rho_*)}{2(1+\beta)\|\mathbf{u}\|}} = z_0 - \frac{(1+\beta)\|\mathbf{u}\|^2}{\gamma}, \end{aligned}$$

because the term under the square root is nonnegative. Consequently,  $\mathbf{c}\mathbf{x} \geq z_0 - (1+\beta)\|\mathbf{u}\|^2/\gamma$  for all  $\mathbf{x} \in \mathcal{X}$  as claimed.

### Exercise 8.8

Show that for fixed  $\rho \in (0, \rho_\infty)$  the pre-image  $\mathbf{x}(\rho, \tau)$  of the line  $\mathbf{y}(\rho, \tau)$  of (8.39) under the projective transformation  $T_0$  defines a direction of descent for (LP), i.e. that  $\mathbf{x}(\rho, \tau) \in \text{relint}\mathcal{X}$  for  $0 \leq \tau \leq \tau(\rho)$  where  $\tau(\rho) > 0$  and that the objective function  $c\mathbf{x}(\rho, \tau)$  decreases strictly for  $\tau \geq 0$ .

By Remark 8.6 the vector  $\mathbf{y}(\rho)$  solves  $(\text{FLP}_\rho)$  and satisfies  $y_{n+1}(\rho) > 0$  for  $0 \leq \rho < \rho_\infty$ . However,  $\mathbf{y}(\rho)$  may become negative in one or several of the components  $1, \dots, n$  for values of  $\rho < \rho_\infty$ . The same is, of course, true for  $\mathbf{y}(\rho, \tau) = \mathbf{y}^0 + \frac{\tau}{\rho}(\mathbf{y}(\rho) - \mathbf{y}^0)$ , but for each value of  $\rho \in (0, \rho_\infty)$  there exists some  $\tau(\rho) > 0$  such that  $\mathbf{y}(\rho, \tau) > 0$  since  $\mathbf{y}^0 = \frac{1}{n+1}\mathbf{f} > 0$ . It follows that the pre-image  $\mathbf{x}(\rho, \tau)$  under the projective transformation  $T_0$ , i.e.,

$$\mathbf{x}(\rho, \tau) = \frac{1}{y_{n+1}(\rho, \tau)} \mathbf{D}\mathbf{y}_N(\rho, \tau) > \mathbf{0},$$

and moreover,  $A\mathbf{x}(\rho, \tau) = \mathbf{b}$  for  $0 \leq \tau \leq \tau(\rho)$ , where  $\rho \in (0, \rho_\infty)$  is arbitrary. Consequently,  $\mathbf{x}(\rho, \tau) \in \text{relint}\mathcal{X}$  for  $0 \leq \tau \leq \tau(\rho)$ . It remains to calculate the objective function  $c\mathbf{x}(\rho, \tau)$  along the line  $\mathbf{x}(\rho, \tau)$ . After some simplifications we find that

$$c\mathbf{x}(\rho, \tau) = \frac{(\mathbf{c}\mathbf{D}, \mathbf{0})\mathbf{y}(\rho, \tau)}{y_{n+1}(\rho, \tau)} = z_0 - \frac{\tau \|\mathbf{u}\| \sqrt{1 - \rho^2/\|\mathbf{w}\|^2}}{\frac{1}{n+1} + \tau \left[ \frac{\gamma}{(1+\beta)\|\mathbf{u}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{w}\|^2}} - \frac{\rho}{(n+1)\|\mathbf{w}\|^2} \right]}.$$

Calculating the first derivative of  $c\mathbf{x}(\rho, \tau)$  with respect to  $\tau$  we get

$$\frac{dc\mathbf{x}(\rho, \tau)}{d\tau} = - \frac{\frac{\|\mathbf{u}\|}{n+1} \sqrt{1 - \rho^2/\|\mathbf{w}\|^2}}{\left( \frac{1}{n+1} + \tau \left[ \frac{\gamma}{(1+\beta)\|\mathbf{u}\|} \sqrt{1 - \frac{\rho^2}{\|\mathbf{w}\|^2}} - \frac{\rho}{(n+1)\|\mathbf{w}\|^2} \right] \right)^2} < 0,$$

i.e., for every  $\rho \in (0, \rho_\infty)$  the objective function value  $c\mathbf{x}(\rho, \tau)$  decreases monotonically for  $\tau \geq 0$ . In other words, every point  $\mathbf{y}(\rho)$  of the projective curve for  $\rho \in (0, \rho_\infty)$  defines a direction of descent for (LP).

### Exercise 8.9

Write a computer program that converts any linear program into the form required by the projective algorithm. Write a computer program for the algorithm using any “canned” subroutine for inverting a square matrix and solve the numerical examples of Exercises 5.1, 6.8 and 8.2.

The following program is written in MATLAB and can be used to convert an LP form standard form to the form required by the Projective Algorithm; see the discussion on pages 259–262 of the book. The parameters  $K$  and  $M$  are data dependent. Here they are set at values that will work for the problems of Exercises 5.1, 6.8, and 8.2.

```

%%%%%
%% This program converts an LP from standard form to the format
%% required by the Projective Algorithm.
%%
%% NAME    : convert
%% PURPOSE: Prepare LP for projective algorithm
%% INPUT   : The matrix A, and the vectors c and b.
%%           The parameters M, K are data dependent.
%%%%%

M=10000;
K=600;
[m,n]=size(A);
c=[c 0 M];
bhat=(n+2)*b'-K*A*ones(n,1);
A=[K*A zeros(m,1) bhat; ones(1,n) 1 1];
b=[K*b K];
[m,n]=size(A);
x=[(K/n)* ones(1,n)];

```

The following is the MATLAB implementation of the projective algorithm. It requires the program convert.m.

```

%%%%%
%% This is the implementation of the Projective Algorithm.
%%
%% NAME    : projal
%% PURPOSE: Solve the LP: min {cx: A x = b, x >=0}
%% INPUT   : The matrix A, and the vectors c and b. The input
%%           data are converted using the program convert.m
%%           to be in the required format.
%%
%% OUTPUT  : z : the optimal value
%%           x : the optimal solution
%%           k : the number of iterations
%%%%%

convert;
if (any(x) <= 0)
error('x is NOT an interior point');
end;

alpha = 0.5;
ptol = 10^(-12);
k = 1;
z = c*x';
z0= z;
v=-K;
```

```

while ( abs((z0-v)/(z+M)) > ptol );
x0 = x;
D = diag(x0);
G = A*D*D*A';
Ginv = inv(G);
P = eye(n) - D*A'*Ginv*A*D;
p = P*D*c';
d = P*ones(n,1);
gamma= p'*d;
beta = n - norm(d,2)^2;
normpsq= norm(p,2)^2;
kappa =(n+1)*(normpsq*(norm(d,2)^2) - (gamma^2));
z0 = c*x0';
lambda=(1+beta)*normpsq+gamma^2;
rho = alpha;
num=sqrt((1+beta)*(lambda*rho^2-kappa))-(n+1)*gamma;
den=beta-n+(1+beta)*(rho^2)/(n+1);
v=z0-num/den;
t=(n+1)*(beta+1)/(gamma*(n+1)+(z0-v)*(1+2*beta-n));
x = x0 - t*(D*(p - ((z0-v-gamma)/(1+beta))*d))';
if (imag(any(x)) ~= 0 )
error('x has complex components');
end;
fprintf('%3d ',k);
fprintf('%10.5f ',x);
fprintf('\n');
k = k + 1;
end;
fprintf('Optimal value: %10.5f\n',c*x')

```

We test the program for the problem of Exercise 5.1. The input is given in the file bpadat.m as follows

```

A=[1 1 1 1 1 0; 3 1 4 2 0 1];
c=[-2 -3 -4 -2 0 0];
b=[10 12];

```

The first and last iterations from the screen output are shown next.

```

>> clear
>> bpadat
>> projal
   1   64.69482   70.56818   61.80460   67.62221   73.43981   95.75955   98.68694   67.42389
   2   53.47868   64.84489   48.98715   58.74233   71.71628   114.26242   128.83756   59.13069
   3   42.93777   57.74544   37.95660   49.34318   68.76391   124.53185   168.01982   50.70142
   4   34.07462   49.54035   29.46349   40.40608   63.34234   122.25529   218.41915   42.49868
   5   26.83375   40.96193   22.89722   32.42821   55.20654   109.02405   278.06043   34.58787
   6   20.72534   32.68284   17.53339   25.36278   45.68306   90.27042   340.62146   27.12070
   7   15.61572   25.24214   13.12603   19.29136   36.30436   70.62721   399.33163   20.46156

```

```

8   11.50832  19.01266  9.61992  14.33496  28.06552  52.89125  449.66161  14.90576
9   8.34107  14.09220  6.93351  10.47732  21.38683  38.30755  489.93461  10.52690
10  5.97983  10.36857  4.93940  7.58385  16.27285  27.04741  520.58662  7.22147
11  4.26417  7.63527  3.49561  5.47167  12.52191  18.74221  543.06329  4.80588
12  3.04351  5.67322  2.47235  3.96176  9.87342  12.84832  559.04024  3.08718
.....  

79  0.00000  9.33333  0.66667  0.00000  0.00000  0.00000  590.00000  0.00000
80  0.00000  9.33333  0.66667  0.00000  0.00000  0.00000  590.00000  0.00000
81  0.00000  9.33331  0.66667  0.00000  0.00000  0.00000  590.00000  0.00000
Optimal value: -30.66660
>>

```

For Exercise 6.8 the data and the output are as follows

```

A=[-1 -1 -1 -1 1 0; -2 -1 -4 -2 0 1];
c=[2 3 4 2 0 0];
b=[-10 -12];

>> clear
>> bpadat
>> proj1
1   69.42450  76.69456  54.84772  69.42450  89.32738  93.94356  86.66433  59.67345
2   60.42165  74.37959  43.44866  60.42165  98.42957  121.66414  95.80148  45.43325
3   52.11869  67.49443  35.66125  52.11869  105.48386  155.25321  99.80853  32.06135
4   45.02822  59.48423  30.26283  45.02822  111.64495  189.59155  98.67214  20.28787
5   39.85124  52.96356  26.61795  39.85124  116.70207  217.73253  94.91561  11.36579
6   36.72009  48.89227  24.46881  36.72009  120.04525  235.82215  91.48621  5.84512
7   35.07718  46.72240  23.33928  35.07718  121.86067  245.53726  89.47135  2.91467
8   34.26434  45.59576  22.76877  34.26434  122.74497  250.38327  88.53150  1.44706
9   33.87916  44.94881  22.47731  33.87916  123.12184  252.73373  88.24046  0.71951
10  33.72389  44.44958  22.31676  33.72389  123.18671  253.80234  88.43845  0.35839
11  33.71759  43.84748  22.20481  33.71759  122.97497  254.13546  89.22333  0.17878
12  33.85140  42.85251  22.08166  33.85140  122.38080  253.88414  91.00873  0.08936
.....  

76  5.00000  0.00000  0.00000  5.00000  0.00000  8.00000  582.00000  0.00000
77  5.00000  0.00000  0.00000  5.00000  0.00000  8.00000  582.00000  0.00000
78  5.00000  0.00000  0.00000  5.00000  0.00000  8.00000  582.00000  0.00000
Optimal value: 20.00000
>>

```

For Exercise 8.2 the data and the output are as follows

```

A=[1 5 1 0 0 0; 1 1 0 1 0 0; 2 -3 0 0 1 0; 3 1 0 0 0 1 ];
b=[250 80 40 180];
c=[0 1 0 0 0 0];

>> clear
>> bpadat
>> proj1
1   58.27568  63.78933  84.79335  69.72507  83.97798  91.72202  89.89413  57.82245
2   43.53301  53.58143  93.07818  64.35887  91.20288  105.38709  106.71718  42.14137

```

```

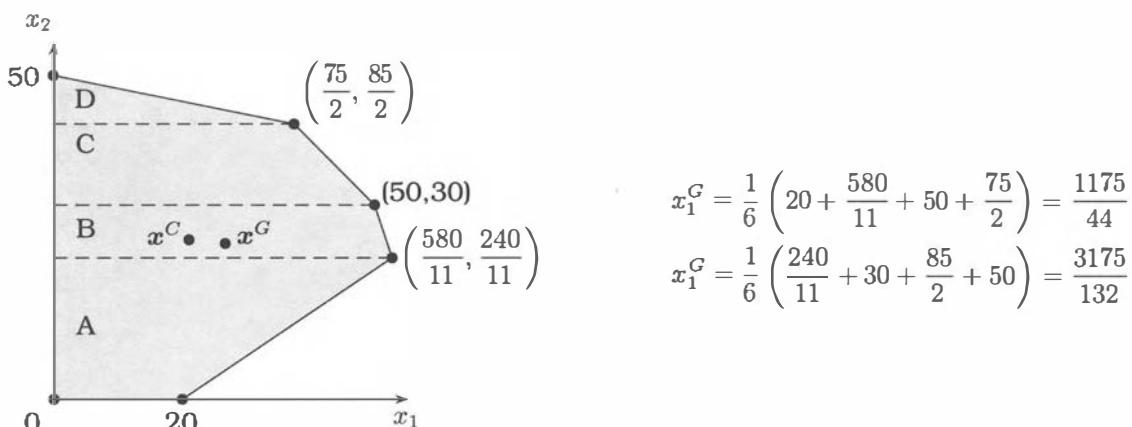
3 32.21091 44.95982 99.17441 58.81741 95.01263 113.70185 127.16359 28.95939
4 24.80541 38.09825 102.64587 52.92058 94.80138 115.66295 152.53579 18.52978
5 20.62466 33.08330 103.43521 47.10685 92.25856 113.03505 179.69008 10.76628
6 18.44056 29.91897 102.79707 42.62486 89.84560 109.53145 201.15991 5.68158
7 17.32901 28.19226 102.18098 39.99995 88.39565 107.24579 213.80055 2.85580
8 16.76731 27.31175 101.87929 38.66558 87.64349 106.07739 220.23555 1.41964
9 16.48603 26.86598 101.77192 38.01250 87.24947 105.51096 223.39737 0.70578
10 16.34731 26.63080 101.78723 37.70131 87.01035 105.24096 224.93061 0.35142
11 16.28214 26.48679 101.92627 37.56976 86.80265 105.12227 225.63494 0.17518
12 16.25784 26.36108 102.25721 37.55004 86.52095 105.09262 225.87287 0.08739
.....
59 10.68217 0.00000 239.31783 69.31783 18.63566 147.95349 114.09302 0.00000
60 10.68217 0.00000 239.31783 69.31783 18.63566 147.95349 114.09302 0.00000
61 10.68217 0.00000 239.31783 69.31783 18.63566 147.95349 114.09302 0.00000
Optimal value: 0.00000
>>

```

### Exercise 8.10

- (i) Show that the barycenter of the feasible set of the linear program of Exercise 8.2 (ii) is given by  $x_1^G = 1,175/44 \approx 26.705$ ,  $x_2^G = 3,175/132 \approx 24.053$  and that its centroid is given by  $x_1^C = 331,855/15,774 \approx 21.038$ ,  $x_2^C = 2,719,075/110,418 \approx 24.625$ .
- (ii) Show that  $x^0 = \frac{1}{2}e$  is the barycenter and centroid of the  $n$ -dimensional unit cube  $C_n$ .

- (i) The figure shows the extreme points of the polytope with their coordinates as well as the calculation of the barycenter.



For the analytic center we calculate first the volume (which in two dimensional space is the area) of the polytope, which is the sum of the areas  $A$ ,  $B$ ,  $C$ , and  $D$ , i.e.

$$\text{vol}(S) = \frac{96000}{121} + \frac{50850}{121} + \frac{4375}{8} + \frac{1125}{8} = \frac{41825}{22}.$$

Now for  $j = 1, 2$  we have

$$x_j^C = \frac{1}{\text{vol}(S)} \iint_A x_j dx_1 dx_2 + \iint_B x_j dx_1 dx_2 + \iint_C x_j dx_1 dx_2 + \iint_D x_j dx_1 dx_2 .$$

For  $j = 1$  we have

$$\begin{aligned} \iint_A x_1 dx_1 dx_2 &= \int_0^{\frac{240}{11}} dx_2 \int_0^{20+\frac{3}{2}x_2} x_1 dx_1 = \frac{20496000}{1331} \\ \iint_B x_1 dx_1 dx_2 &= \int_{\frac{240}{11}}^{30} dx_2 \int_0^{60-\frac{1}{3}x_2} x_1 dx_1 = \frac{14368500}{1331} \\ \iint_C x_1 dx_1 dx_2 &= \int_{30}^{\frac{85}{2}} dx_2 \int_0^{80-x_2} x_1 dx_1 = \frac{578125}{48} \\ \iint_D x_1 dx_1 dx_2 &= \int_{\frac{85}{2}}^{50} dx_2 \int_0^{250-5x_2} x_1 dx_1 = \frac{28125}{24} \end{aligned}$$

and thus

$$x_1^C = \frac{22}{41825} \left( \frac{20496000}{1331} + \frac{14368500}{1331} + \frac{578125}{48} + \frac{28125}{24} \right) = \frac{331855}{15774} .$$

For  $j = 2$  we calculate

$$\begin{aligned} \iint_A x_2 dx_1 dx_2 &= \int_0^{\frac{240}{11}} x_2 dx_2 \int_0^{20+\frac{3}{2}x_2} dx_1 = \frac{13248000}{1331} \\ \iint_B x_2 dx_1 dx_2 &= \int_{\frac{240}{11}}^{30} x_2 dx_2 \int_0^{60-\frac{1}{3}x_2} dx_1 = \frac{14472000}{1331} \\ \iint_C x_2 dx_1 dx_2 &= \int_{30}^{\frac{85}{2}} x_2 dx_2 \int_0^{80-x_2} dx_1 = \frac{471875}{24} \\ \iint_D x_2 dx_1 dx_2 &= \int_{\frac{85}{2}}^{50} x_2 dx_2 \int_0^{250-5x_2} dx_1 = \frac{151875}{24} \end{aligned}$$

and thus

$$x_2^C = \frac{22}{41825} \left( \frac{13248000}{1331} + \frac{14472000}{1331} + \frac{471875}{24} + \frac{151875}{24} \right) = \frac{2719075}{110418} .$$

(ii) From Chapter 7.7 (p. 212) we have  $\text{vol}(C_n) = 1$  and thus

$$x_j^C = \int_0^1 x_j \int_0^1 \cdots \int_0^1 dx_1 \cdots dx_n = \int_0^1 x_j dx_j = \frac{1}{2}$$

and therefore  $x^C = \frac{1}{2}e$ . To calculate the barycenter, we observe that  $C_n$  has  $2^n$  extreme points and that each component  $i$ ,  $1 \leq i \leq n$  has value of 1 in exactly half of them. Thus

$$x_j^G = \frac{1}{2^n} \sum_{k=1}^{2^n} x_j^k = \frac{2^{n-1}}{2^n} = \frac{1}{2}$$

and therefore  $x^G = \frac{1}{2}e$  as well.

---

### Exercise 8.11

- (i) Show that the analytic center of the simplex  $S^{n+1}$  is given by  $x^{gbar} = \frac{1}{n+1}f$  where  $f^T = (1, \dots, 1) \in \mathbb{R}^{n+1}$ . (Hint: Use the geometric/arithmetic mean inequality.)
  - (ii) Show that the analytic center of the polytope of the linear program of Exercise 8.2 (ii) is given by the unique positive maximizer  $(x_1^0, x_2^0)$  of the function  $10^6 x_1 x_2 (144 - 11.976 x_1 + 5.32 x_2 + 0.2856 x_1^2 - 0.349 x_2^2) + 10^3 x_1 x_2 (50.2 x_1 x_2 - 3.97 x_1^2 x_2 - 2.46 x_1^3 + 6.14 x_1 x_2^2 + 4.45 x_2^3) - 15 x_1^3 x_2^3 + 29 x_1^4 x_2^2 - 53 x_1^2 x_2^4 + 6 x_1^5 x_2 - 15 x_1 x_2^5$  and that  $x_1^0 \approx 12.507$ ,  $x_2^0 \approx 24.407$ .
- 

**(i)** The analytic center for the simplex  $S^{n+1} = \{x \in \mathbb{R}^n : \sum_{j=1}^{n+1} x_j = 1, x \geq 0\}$  is the point  $x$  that minimizes  $\left(\prod_{j=1}^{n+1} x_j\right)^{-\frac{1}{n+1}}$  which is the same point that maximizes  $\left(\prod_{j=1}^{n+1} x_j\right)^{\frac{1}{n+1}}$ . From the geometric/arithmetic mean inequality (see point 7.7(f) on page 212) we have that

$$\left(\prod_{j=1}^{n+1} x_j\right)^{\frac{1}{n+1}} \leq \frac{1}{n+1} \sum_{j=1}^{n+1} x_j = \frac{1}{n+1}$$

where the last equality follows since  $x \in S^{n+1}$ . Moreover, from point 7.7(f) we have that equality is achieved if and only if  $x_j = \lambda$  for all  $j = 1, \dots, n+1$ , which in our case means that the equality is achieved for  $x = \frac{1}{n+1}f$ , where  $f$  is the vector in  $\mathbb{R}^{n+1}$  with all components equal to one. Thus, the analytic center of  $S^{n+1}$  is  $\frac{1}{n+1}f$ .

**(ii)** The minimizer of the function  $gbar(x)$  is the same as the maximizer of  $\prod_{j=1}^{n+1} x_j$ . So the analytic center of the polytope is the point that solves the problem  $\max x_1 x_2 x_3 x_4 x_5 x_6$ , where  $x_j$ ,  $j = 3, \dots, 6$  are the slack variables. Substituting  $x_3 = 250 - x_1 - 5x_2$ ,  $x_4 = 80 - x_1 - x_2$ ,  $x_5 = 180 - 3x_1 - x_2$  and  $x_6 = 40 - 2x_1 - 3x_2$  we get the following problem in two variables

$$\max x_1 x_2 (250 - x_1 - 5x_2)(80 - x_1 - x_2)(180 - 3x_1 - x_2)(40 - 2x_1 - 3x_2)$$

which gives, after the multiplications and simplifications, the problem  $\max 10^6 x_1 x_2 (144 - 11.976 x_1 + 5.32 x_2 + 0.2856 x_1^2 - 0.349 x_2^2) + 10^3 x_1 x_2 (50.2 x_1 x_2 - 3.97 x_1^2 x_2 - 2.46 x_1^3 + 6.14 x_1 x_2^2 + 4.45 x_2^3) - 15 x_1^3 x_2^3 + 29 x_1^4 x_2^2 - 53 x_1^2 x_2^4 + 6 x_1^5 x_2 - 15 x_1 x_2^5$ . Using the following MATLAB code

```

F=' -1000000*x(1)*x(2)*(144-11.976*x(1)+5.32*x(2)+0.2856*x(1)*x(1)-0.349*x(2)*x(2))
-1000*x(1)*x(2)*(50.2*x(1)*x(2)-3.97*x(1)*x(1)*x(2)-2.46*x(1)^3+6.14*x(1)*x(2)^2
+4.45*x(2)^3)+15*x(1)^3*x(2)^3-29*x(1)^4*x(2)^2+53*x(1)^2*x(2)^4-6*x(1)^5*x(2)-
15*x(1)*x(2)^5';
y=fmins(F,[10;20])

```

we obtain the optimizing vector  $(12.5073, 24.4066)$ . Note that the definition of F must be given in one line for MATLAB to work correctly.

---

### Exercise 8.12

- (i) Suppose that  $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \neq \emptyset$  is bounded and that  $c \in \mathbb{R}^n$  is arbitrary. Show that  $\mathcal{U}_c$  is not bounded and that  $\text{relint } \mathcal{U}_c \neq \emptyset$  where  $\mathcal{U}_c = \{u \in \mathbb{R}^m : u^T A \leq c\}$ . (Hint: See Exercise 6.9 (ii).)
  - (ii) Compute the log-center and the log-central path for the polytope of Exercise 8.2 (ii).
- 

**(i)** By Exercise 6.9(ii) it follows that there exists  $u \in \mathbb{R}^m$  such that  $d = u^T A > 0$ . Consequently,  $(-\lambda)d \leq c$  for some  $\lambda > 0$  no matter what  $c \in \mathbb{R}^n$  and thus  $\mathcal{U}_c \neq \emptyset$ . Now let  $v \in \mathcal{U}_c$  be arbitrary. Then  $v - \lambda u \in \mathcal{U}_c$  for all  $\lambda \geq 0$ , where  $u \in \mathbb{R}^m$  is such that  $u^T A > 0$  and consequently  $\mathcal{U}_c$  is unbounded. Moreover,  $(v^T - \lambda u^T)A \leq c - \lambda u^T A < c$  for all  $\lambda > 0$  and thus  $\text{relint } \mathcal{U}_c \neq \emptyset$  as well.

**(ii)** The log-center is the analytic center we calculated in part (ii) of Exercise 8.11. To calculate the log-central path we use the following program (to be used with Mathematica for various values of the parameter m which was changed by increments of 0.001 in a do-loop).

```

A={{1,5,1,0,0,0},{1,1,0,1,0,0},{3,1,0,0,1,0},{2,-3,0,0,0,1}}
c={0,1,0,0,0,0}
b={250,80,180,40}
u={u1,u2,u3,u4}
x={x1,x2,x3,x4,x5,x6}
r={r1,r2,r3,r4,r5,r6}
m=0.001
N[Solve[{u . A + r == c, A . x == b, x1 r1 == m, x2 r2 == m,
          x3 r3 == m, x4 r4 == m, x5 r5 == m, x6 r6 == m},
          {x1,x2,x3,x4,x5,x6,u1,u2,u3,u4,r1,r2,r3,r4,r5,r6}]]
```

---

### Exercise 8.13

- (i) Write a computer program for the iterative procedure discussed in this section using any subroutine for inverting a square matrix and solve the numerical examples of Exercises 5.1, 6.8 and 8.2.

- (ii) Derive a method to find a **basic feasible solution**  $x^1$  for the linear program (LP) given a near-optimal feasible interior point  $x^0 \in \mathcal{X}$  satisfying  $cx^1 \leq cx^0$ . Generalize this construction so as to permit a practical way of "crossing over" to a simplex algorithm from any near-optimal interior point  $x^0 \in \mathcal{X}$ . (Hint: Use the proof of Theorem 1.)
- 

(i)

```

%%%%%
%% This is the implementation of the Iterative Procedure.
%%
%% NAME      : itpro
%% PURPOSE: Solve the LP: min {cx: A x = b, x >=0}
%% INPUT    : The matrix A, the vectors c and b, a
%%             starting interior point x and a vector u
%%             such that c'-A'*u > 0.
%% OUTPUT   : z : the optimal value
%%             x : the optimal solution
%%             k : the number of iterations
%%%%%

ptol=10^(-8);
maxit = 100;
[m,n]=size(A);
for i=1:m,
  sum=0;
  for j=1:n-m,
    sum=sum+A(i,j)*x(j);
  end;
  x(i+n-m)=b(i)-sum;
end;

if (any(x) <= 0)
  error('x is NOT an interior point');
end;

k = 1;
r=c'-A' *u';
R=diag(r);
D=diag(x);
mu=(1/n)*x*r;
z=c*x';
Z=b*u';
e=ones(n,1);
while (abs(Z-z) > ptol);

```

```

f=b'-A*x';
g=c' - A'*u' -r;
h=mu*e - D*R*e;
Rinv=inv(R);
B=A*inv(R)*D*A';
Binv=inv(B);
Dx= Rinv*D*A'*Binv*f + Rinv*(eye(n)-D*A'*Binv*A*Rinv)*(h-D*g);
Du= Binv*f+Binv*A*Rinv*(D*g-h);
Dr= -A'*Binv*f+(eye(n)-A'*Binv*A*Rinv*D)*g+A'*Binv*A*Rinv*h;
Dinv=inv(D);
p=(1/.95)*Dinv*Dx;
d=(1/.95)*Rinv*Dr;
pp=max(-p);
dd=max(-d);
alphap=max(1,pp);
alphad=max(1,dd);
x=x+(1/alphap)*Dx';
u=u+(1/alphad)*Du';
r=r+(1/alphad)*Dr;
z=c*x';
Z=b*u';
dualinf=norm(c'-A'*u'-r)/(norm(c)+1);
priminf=norm(A*x'-b')/(norm(b)+1);
if (c*x' > b*u') mu=0.1*r'*x'/n;
else mu=2*r'*x'/n;
end;
D=diag(x);
R=diag(r);
fprintf('%3d ',k);
fprintf('%10.5f ',x);
fprintf('\n');
k=k+1;
end;
fprintf('Optimal value: %10.5f \n',c*x')

```

For Exercise 5.1 the data in a file called `bpadat.m` and the output are as follows

```

A=[1 1 1 1 1 0; 3 1 4 2 0 1];
c=[-2 -3 -4 -2 0 0];
b=[10 12];
x=[1 1 1 1 0 0];
u=[-3 -1];

>> clear
>> bpadat
>> itpro

```

```

1   0.51420   4.69563   0.24579   1.12540   3.41899   2.52783
2   0.37613   6.81110   0.24430   0.83844   1.73004   1.40646
3   0.19677   9.14892   0.17435   0.39346   0.08650   0.77647
4   0.15200   9.36016   0.23011   0.25341   0.00433   0.75660
5   0.02245   9.36526   0.58022   0.02601   0.00607   0.19452
6   0.00197   9.33267   0.66168   0.00247   0.00120   0.00973
7   0.00020   9.33328   0.66616   0.00024   0.00012   0.00100
8   0.00002   9.33333   0.66662   0.00002   0.00001   0.00010
9   0.00000   9.33333   0.66666   0.00000   0.00000   0.00001
10  0.00000   9.33333   0.66667   0.00000   0.00000   0.00000
11  0.00000   9.33333   0.66667   0.00000   0.00000   0.00000
12  0.00000   9.33333   0.66667   0.00000   0.00000   0.00000
13  0.00000   9.33333   0.66667   0.00000   0.00000   0.00000
Optimal value: -30.66667
>>

```

For Exercise 6.8 the data and the output are as follows

```

A=[-1 -1 -1 -1 1 0; -2 -1 -4 -2 0 1];
c=[2 3 4 2 0 0];
b=[-10 -12];
x=[12 1 1 1 0 0];
u=[-0.1 -0.1];

>> clear
>> bpadat
>> itpro
1   10.88071   2.18482   1.85885   3.67470   8.59908   26.73103
2   6.23662   0.91935   0.71275   2.56124   0.42995   9.36606
3   6.43364   0.52467   0.25042   2.81277   0.02150   8.01917
4   6.31115   0.08015   0.01252   3.61056   0.01438   7.97365
5   6.15815   0.00415   0.00177   3.83764   0.00172   8.00282
6   6.03924   0.00036   0.00018   3.96039   0.00018   8.00035
7   5.93515   0.00004   0.00002   4.06481   0.00002   8.00004
8   5.84163   0.00000   0.00000   4.15837   0.00000   8.00000
9   5.75747   0.00000   0.00000   4.24253   0.00000   8.00000
10  5.68172   0.00000   0.00000   4.31828   0.00000   8.00000
11  5.61355   0.00000   0.00000   4.38645   0.00000   8.00000
12  5.55219   0.00000   0.00000   4.44781   0.00000   8.00000
Optimal value: 20.00000
>>

```

For Exercise 8.2 the data and the output are as follows

```

A=[1 5 1 0 0 0; 1 1 0 1 0 0; 2 -3 0 0 1 0; 3 1 0 0 0 1 ];
b=[250 80 40 180];
c=[0 1 0 0 0 0];
u=[-0.1 -0.1 -0.1 -0.1];

```

```

>> clear
>> bpdat
>> itpro
    1    1.50000   19.69390   150.03049   58.80610   96.08171   155.80610
    2    2.72075   9.41600   200.19925   67.86325   62.80650   162.42175
    3    3.13002   0.47080   244.51598   76.39918   35.15236   170.13914
    4    3.33234   0.02354   246.54996   76.64412   33.40594   169.97944
    5    3.67629   0.00520   246.29774   76.31852   32.66301   168.96595
    6    4.11005   0.00053   245.88729   75.88942   31.78150   167.66932
    7    4.51568   0.00005   245.48406   75.48427   30.96881   166.45292
    8    4.88307   0.00001   245.11690   75.11692   30.23388   165.35079
    9    5.21401   0.00000   244.78599   74.78599   29.57198   164.35797
   10   5.51189   0.00000   244.48811   74.48811   28.97623   163.46434
   11   5.77998   0.00000   244.22002   74.22002   28.44004   162.66007
   12   6.02126   0.00000   243.97874   73.97874   27.95748   161.93622
Optimal value: 0.00000
>>

```

**(ii)** Let us assume for convenience that  $r(\mathbf{A}) = m$ . The proof of Theorem 1 (part b) suggests the following procedure. Initially we let  $k = 0$ ,  $\mathbf{x}^k = \mathbf{x}^0$ ,  $I_k = \{j \in N : x_j^k > 0\}$ . In the iterative step we solve  $\sum_{j \in I_k} \lambda_j a_{j,j} = 0$  by Gaussian elimination to find  $\boldsymbol{\lambda}^k = (\lambda_j)_{j \in I_k} \neq \mathbf{0}$ . This is accomplished by forcing e.g. the “first”  $\lambda_j = 1$  where  $j \in I_k$ . If no such  $\boldsymbol{\lambda}^k$  exists, we stop;  $\mathbf{x}^k$  is a basic feasible solution with  $\mathbf{c}\mathbf{x}^k \leq \mathbf{c}\mathbf{x}^0$ . Otherwise, we compute  $\gamma = \sum_{j \in I_k} c_j \lambda_j$ . If  $\gamma < 0$ , we change the sign of all  $\lambda_j$  with  $j \in I_k$  so that we get WROG  $\gamma \geq 0$ . Now suppose that  $\gamma > 0$  and  $\lambda_j \leq 0$  for all  $j \in I_k$ . Then  $\mathbf{x}(\theta)$  defined by  $x_j(\theta) = x_j^k - \theta \lambda_j$  for  $j \in I_k$ ,  $x_j(\theta) = 0$  for  $j \in N - I_k$  is feasible for all  $\theta \geq 0$  and  $z(\theta) = \mathbf{c}\mathbf{x}^k - \theta \gamma \rightarrow -\infty$  for  $\theta \rightarrow +\infty$  and we stop with the message that (LP) is unbounded. Otherwise,  $\gamma \geq 0$  and there exists at least one  $j \in I_k$  with  $\lambda_j > 0$ . We let as in the least ratio test of the simplex algorithm  $\theta_0 = \min\{\frac{x_j^k}{\lambda_j} : \lambda_j > 0\}$  and define  $x_j^{k+1} = x_j^k - \theta_0 \lambda_j$  for  $j \in I_k$ ,  $x_j^{k+1} = 0$  for  $j \in N - I_k$ . It follows that  $\mathbf{x}^{k+1} \in \mathcal{X}$  and  $\mathbf{c}\mathbf{x}^{k+1} = \mathbf{c}\mathbf{x}^k - \theta_0 \gamma \leq \mathbf{c}\mathbf{x}^k$ . We set  $I_{k+1} = \{j \in N : x_j^{k+1} > 0\}$ , replace  $k$  by  $k + 1$  and repeat. Since  $|I_{k+1}| \leq |I_k| - 1$  for each  $k$ , the procedure executes at most  $n$  steps before it comes to a halt in either of the two cases. In a practical application we need to assure the accuracy of the resulting solution by using tolerances, etc., when the operations are carried out in floating point arithmetic. The full rank assumption is fulfilled by considering artificial variables when necessary. Once a basic feasible solution is obtained, one then carries out a “pricing step” of the simplex algorithm and iterates if necessary.

### Exercise 8.14

Write a computer program for the Newtonian algorithm using the formulas (8.64), (8.65), (8.66) and any subroutine for inverting a square matrix. Use your program to reproduce the path to optimality starting at  $x_1 = x_2 = 15$  for Figure 8.11 and Exercise 8.2.

```

%%%%%
%% This is the implementation of the Newtonian Algorithm.
%%
%% NAME      : newton
%% PURPOSE: Solve the LP: min {cx: A x = b, x >=0}
%% INPUT    : The matrix A, the vectors c and b and a
%%             starting interior point x and a suitable vector u
%%             (see pg. 275 for details)
%% OUTPUT   : z : the optimal value
%%             x : the optimal solution
%%             k : the number of iterations
%%%%%

ptol=5;
[m n] = size(A);
for i=1:m,
  sum=0;
  for j=1:n-m,
    sum=sum+A(i,j)*x(j);
  end;
  x(i+n-m)=b(i)-sum;
end;
if (any(x) <= 0)
fprintf('Error. Not an interior point');
  stop;
end;
k = 1;
r=c'-A' *u';
delta=0.4;
R=diag(r);
D=diag(x);
mu=(1/n)*x*r;
theta=norm(x' .* r - mu * ones(n,1))/mu;
while ( x*r > 10^(-ptol) );
  mu=mu*(1-delta/sqrt(n));
  Rinv=inv(R);
  Dinv=inv(D);
  T=(Rinv*D)^(1/2);
  B=A*T^2*A';
  Binv=inv(B);
  S=eye(n,n)-T*A'*Binv*A*T;
  Dx= -T*S*T*c' + mu *T*S*T*Dinv*ones(n,1);
  Du= Binv*b'-mu*Binv*A*Rinv*ones(n,1);
  Dr= -A'*Binv*b'+mu*A'*Binv*A*Rinv*ones(n,1);
  x=x+Dx';
  u=u+Du';
end;

```

```

r=r+Dr;
D=diag(x);
R=diag(r);
fprintf('%3d ',k);
fprintf('%10.5f ',x);
fprintf('\n');
k=k+1;
end;
fprintf('Optimal value: %10.5f \n',c*x')

```

For the data of Exercise 8.2 with starting point  $x = (15, 15)$  and  $u = (-0.1, -0.4, -0.2, -0.3)$  we get

```

>> clear
>> bpadat
>> newton
    1   12.65413   15.87074   157.99216   51.47513   62.30396   126.16687
    2   12.26339   14.55247   164.97425   53.18414   59.13065   128.65737
    3   12.06108   13.18958   171.99100   54.74934   55.44659   130.62718
    4   11.84223   11.78299   179.24281   56.37477   51.66451   132.69031
    5   11.58310   10.36974   186.56818   58.04716   47.94304   134.88097
    6   11.28724   8.99307   193.74743   59.71969   44.40472   137.14521
    7   10.96622   7.69292   200.56918   61.34086   41.14632   139.40842
    8   10.63523   6.50150   206.85726   62.86327   38.23404   141.59280
    9   10.30999   5.44002   212.48990   64.24999   35.70008   143.63000
   10   10.00388   4.51764   217.40793   65.47848   33.54515   145.47072
   11   9.72611   3.73258   221.61099   66.54131   31.74552   147.08909
   12   9.48129   3.07495   225.14396   67.44376   30.26225   148.48117
   13   9.27018   2.53015   228.07908   68.19967   29.05010   149.65932
   14   9.09089   2.08194   230.49942   68.82717   28.06402   150.64538
   15   8.94019   1.71453   232.48717   69.34528   27.26319   151.46489
.....
   91   8.19281   0.00000   241.80718   71.80719   23.61438   155.42156
   92   8.19281   0.00000   241.80718   71.80719   23.61438   155.42156
   93   8.19281   0.00000   241.80718   71.80719   23.61438   155.42156
Optimal value: 0.00000
>>

```

## 9. Ellipsoid Algorithms

Divide et impera!<sup>1</sup>  
Niccolo Machiavelli (1469-1527 A.D.)

Here we summarize the essentials of Chapter 9 of the text. We consider the linear optimization problem over a rational polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$

$$\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}.$$

In Chapter 7.5.3 we reduced the problem of polynomial solvability of this problem to the question of the existence of subroutines FINDXZ( $P, n, \phi, \Phi, \mathbf{c}, \mathbf{z}^k, \mathbf{x}, \text{FEAS}$ ) or FINDZX( $P, n, \phi, \mathbf{c}, \mathbf{z}^k, \mathbf{x}, \text{FEAS}$ ) that solve a **feasibility problem** in polynomial time. The ellipsoid algorithm settles this existence question in a theoretically satisfactory way for any rational polyhedron  $P \subseteq \mathbb{R}^n$ .

By point 7.5(d), we can replace  $P$  by a rational *polytope*  $P_\Phi$  of equal dimension without changing the optimization problem. We assume that we have a linear description  $A\mathbf{x} \leq \mathbf{b}$  of  $P_\Phi$  with rational data  $A, \mathbf{b}$  and initially, that either  $P_\Phi = \emptyset$  or  $\dim P_\Phi = n$ . The case of flat polyhedra is discussed separately. It follows from point 7.5(d) that the ball  $B(\mathbf{0}, R)$  contains  $P_\Phi$ , where  $R = \sqrt{n}2^\Phi$  and

$$\Phi = \langle \mathbf{c} \rangle + 8n\phi + 2n^2\phi + 2.$$

The center of  $B(\mathbf{0}, R)$  is  $\mathbf{x}^0 = \mathbf{0}$ . Checking  $\mathbf{x}^0 \in P_\Phi$  we either find an inequality  $\mathbf{a}^0\mathbf{x} \leq a_0$  of the linear description of  $P_\Phi$  such that  $\mathbf{a}^0\mathbf{x}^0 > a_0$  or  $\mathbf{x}^0 \in P_\Phi$  and we are done. If  $\mathbf{a}^0\mathbf{x}^0 > a_0$  then  $P_\Phi \subseteq B(\mathbf{0}, R) \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^0\mathbf{x} \leq a_0\}$ . Replacing  $a_0$  by  $\mathbf{a}^0\mathbf{x}^0$  we have that

$$P_\Phi \subseteq S_1 = B(\mathbf{0}, R) \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^0\mathbf{x} \leq \mathbf{a}^0\mathbf{x}^0\} \subseteq E_1,$$

where  $E_1$  is an ellipsoid (of minimum volume) that contains  $S_1$ . Let  $\mathbf{x}^1$ , the center of  $E_1$ , be the next “trial” solution: if  $\mathbf{x}^1 \in P_\Phi$  we are done. Otherwise, we find an inequality  $\mathbf{a}^1\mathbf{x} \leq a_1$  from the linear description of  $P_\Phi$  such that  $\mathbf{a}^1\mathbf{x}^1 > a_1$  and iterate. At the  $k^{th}$  iteration of this algorithm we have the center  $\mathbf{x}^k$  of an ellipsoid  $E_k = E_{Q_k}(\mathbf{x}^k, 1)$ , where  $Q_k = F_k F_k^T$  is a positive definite matrix defining  $E_k$ . By construction  $P_\Phi \subseteq E_k$ . Either  $\mathbf{x}^k \in P_\Phi$  – in which case we are done – or we find an inequality  $\mathbf{a}^T\mathbf{x} \leq a_0$  belonging to the linear description of  $P$  or  $P_\Phi$  such that  $\mathbf{a}^T\mathbf{x}^k > a_0$ . In this case we set

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{1}{n+1}F_k d \quad \text{where} \quad d = \frac{F_k^T \mathbf{a}}{\|F_k^T \mathbf{a}\|}, \quad (9.1)$$

$$F_{k+1} = \sqrt{\frac{n^2}{n^2 - 1}} F_k \left( I_n - \left( 1 - \sqrt{\frac{n-1}{n+1}} \right) dd^T \right). \quad (9.2)$$

We get an ellipsoid  $E_{k+1} = E_{Q_{k+1}}(\mathbf{x}^{k+1}, 1)$  with center  $\mathbf{x}^{k+1}$  and positive definite matrix  $Q_{k+1} = F_{k+1} F_{k+1}^T$  defining  $E_{k+1}$ . As shown in Chapter 9.2,  $E_{k+1} \supseteq P_\Phi$  and

$$\text{vol}(E_{k+1}) \leq e^{-1/2n} \text{vol}(E_k). \quad (9.3)$$

---

<sup>1</sup>Divide and conquer!

Iterating  $k$  times, we get  $\text{vol}(E_k) \leq V_0 e^{-k/2n}$  for  $k \geq 0$ , where  $V_0$  is the volume of  $B(\mathbf{0}, R)$ . Unless the algorithm stops with  $\mathbf{x}^k \in P_\Phi$  for some  $k$ , it suffices to iterate at most

$$k_E = \lceil 2n(\log V_0 - \log V_{P_\Phi}) \rceil$$

times to conclude that  $P_\Phi = \emptyset$ , where  $V_{P_\Phi}$  is the volume of  $P_\Phi$ . By assumption  $P$  is either empty or full-dimensional. If  $P \neq \emptyset$ , we can bound  $V_{P_\Phi}$  from below because  $P$  is a rational polyhedron.

In the left part of Figure 9.1 we show the iterative application of the ellipsoid algorithm when applied to the data of Exercise 8.2 (ii) without the objective function. We start at the point  $\mathbf{x}^0 = (60, 60)$  as the center of the initial ball with radius  $\|\mathbf{x}^0\|$  which contains all of the feasible set. We select as the “next” inequality  $\mathbf{a}^T \mathbf{x} \leq a_0$  the one for which the slack  $\mathbf{a}^T \mathbf{x}^0 - a_0$  is largest, to get  $\mathbf{x}^1$ ; etc.

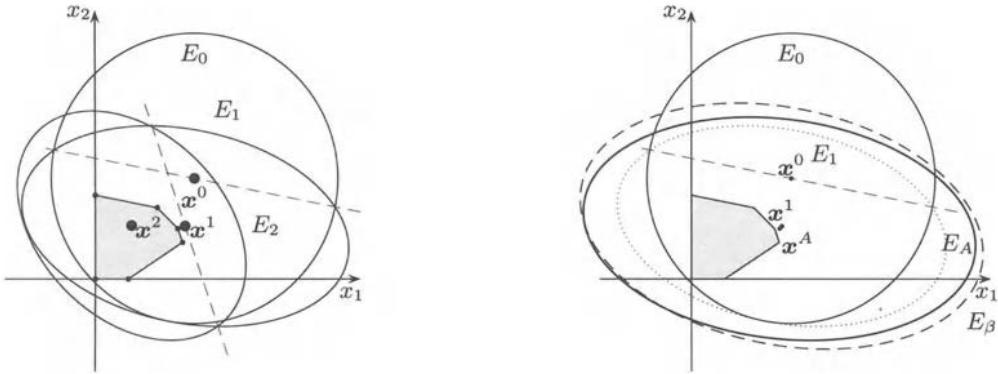
The above formulas yield polynomial *step complexity*, but not polynomial *time complexity* of the calculation. For the latter it is necessary that the digital sizes of  $\langle \mathbf{x}^k \rangle$  and  $\langle \mathbf{F}_k \rangle$  of the iterates stay bounded by a polynomial function of  $n$ ,  $\phi$  and  $\langle c \rangle$ . It must also be shown that all necessary calculations can be carried out in *approximate* arithmetic, i.e. on a computer with limited word size. In the left part of Figure 9.1 we pretended that we can compute (9.1) and (9.2) “perfectly” – even though we divide, take square roots and calculate on a computer with a word size of merely 64 bits. To be correct, we have to replace the equality signs in (9.1) and (9.2) by the  $\approx$  sign and specify the *precision* with which we need to calculate the corresponding numbers.

The geometric idea for the way to deal with the problem of approximate calculations is shown in the right part of Figure 9.1 for the ellipsoid  $E_1$ : since we cannot compute the center  $\mathbf{x}^1$  of  $E_1$  by formula (9.1) exactly, we get an approximate center  $\mathbf{x}^A$  by committing round-off errors. To approximate the matrix  $\mathbf{F}_1$  given by (9.2) we multiply the right-hand side by some factor  $\beta \geq 1$ , i.e. we scale all elements of it *up* to make them bigger. An approximate computation with round-off errors yields a matrix  $\mathbf{F}_A$  that is used in lieu of  $\mathbf{F}_1$ . This corresponds to “blowing up” the perfect arithmetic ellipsoid  $E_1$  concentrically to the ellipsoid  $E_\beta$  of Figure 9.1, i.e.  $E_\beta$  is a homothetic image of  $E_1$  with a factor  $\beta \geq 1$  of dilatation. The approximate calculation of  $\mathbf{F}_A$  is then carried out with a sufficient precision to guarantee that the ellipsoid  $E_A$  with center  $\mathbf{x}^A$  and defining matrix  $\mathbf{Q}_A = \mathbf{F}_A \mathbf{F}_A^T$  contains the ellipsoid  $E_1$  completely. In Chapter 9.2 we show that a blow-up factor of

$$\beta = 1 + 1/12n^2$$

works, where  $n$  is the number of variables of the optimization problem. Our graphical illustration in Figure 9.1 is “artistic”. We used  $\beta \approx \sqrt{1.5}$  to produce the figure and not  $\beta = 49/48$ , which is a lot smaller and works in  $\mathbb{R}^2$ .

In every iteration the ellipsoid algorithm needs only one inequality that is violated or the message that a violated inequality does not exist, i.e., at every iteration we have to solve a **separation problem** (or constraint identification problem) for the polyhedron  $P$  like the one we discussed in point 7.5(h). So far we have assumed that the number of constraints of the linear description of  $P$  does not matter and that the problem of finding a violated inequality can e.g. be done by *listing and checking* every single one of them. We call this method LIST-and-CHECK. LIST-and-CHECK does not give a polynomial algorithm for the linear optimization over rational polyhedra if the number of constraints of the linear description of  $P$  is exponential in  $n$ . But assuming the existence of *some* polynomial-time algorithm for the separation problem we get the existence of a polynomial-time algorithm for the optimization problem over rational polyhedra – namely the ellipsoid algorithm. It is a nontrivial result that the reverse statement holds as well:



**Fig. 9.1.** The ellipsoid algorithm: “perfect” and approximate arithmetic

if for some rational polyhedron  $P$  the optimization problem can be solved in polynomial time, then the separation problem for  $P$  can be solved in polynomial time as well. This equivalence of optimization and separation for rational polyhedra is of fundamental importance in itself and particularly important for the field of combinatorial optimization: it constitutes the theoretical backbone of the algorithmic approach to combinatorial optimization problems called branch-and-cut.

## 9.1 Matrix Norms, Approximate Inverses, Matrix Inequalities

We have to “truncate” numbers in the ellipsoid algorithm and thus to replace e.g. a matrix  $F$  by some matrix  $F_A$ , say, satisfying  $F_A \approx F$ . To analyze such an approximation we need a *norm*. The *Frobenius norm* of an  $m \times n$  matrix  $F$  of real numbers  $f_{ij}^i$  is

$$\|F\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (f_{ij}^i)^2}. \quad (9.4)$$

Other norms that are frequently encountered in numerical linear algebra are

$$\|F\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |f_{ij}^i|, \quad \|F\|_2 = \max\{\|Fx\| : \|x\| = 1\}, \quad \|F\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |f_{ij}^i|.$$

$\|F\|_2$  is the *spectral norm* and its value equals the square root of the largest eigenvalue of  $F^T F$ . If  $F$  is nonsingular, then  $\|F^{-1}\|_2 = \lambda^{-1}(F)$ , i.e., it is the reciprocal of the square root of the smallest eigenvalue of  $F^T F$ . There are a number of relationships between these various matrix norms, e.g.

$$\|F\|_2 \leq \|F\|_F \leq \sqrt{n} \|F\|_2.$$

The spectral norm yields probably the most elegant proofs for what is to follow, but it is also the hardest one to compute. To avoid issues of computation we will not use it. We use the Frobenius norm and simply drop the subscript  $F$  for notational convenience. In Exercise 9.2 most of the

properties of the Frobenius norm that we need are stated. In particular, for any two  $m \times n$  real matrices we have

$$\|\mathbf{F}\mathbf{R}\| \leq \|\mathbf{F}\| \|\mathbf{R}\|, \quad \|\mathbf{F}(\mathbf{I}_n - \alpha \mathbf{r} \mathbf{r}^T)\|^2 = \|\mathbf{F}\|^2 - \alpha(2 - \alpha \|\mathbf{r}\|^2) \|\mathbf{F}\mathbf{r}\|^2. \quad (9.5)$$

**Remark 9.1** Let  $\mathbf{R}$  be any  $n \times n$  matrix of reals with  $\|\mathbf{R}\| < 1$ . Then  $(\mathbf{I}_n - \mathbf{R})^{-1}$  exists and

$$\|(\mathbf{I}_n - \mathbf{R})^{-1}\| \leq \frac{1}{1 - \|\mathbf{R}\|}. \quad (9.6)$$

Let the elements of  $\mathbf{F}_A$  be obtained by truncating the elements of some nonsingular matrix  $\mathbf{F}$ , i.e.,  $\mathbf{F}_A \approx \mathbf{F}$ , and  $\mathbf{F}_A = \mathbf{F} + \mathbf{R}$ . Thus  $\mathbf{R}$  is the matrix of “errors” due to the rounding or the truncation and the Frobenius norm is the sum of the squared errors for each element of  $\mathbf{F}_A$ . We need to know when  $\mathbf{F}_A$  is nonsingular and how the errors “propagate” into the inverse of  $\mathbf{F}_A$ .

**Remark 9.2** Let  $\mathbf{F}$  be any nonsingular matrix of size  $n \times n$  and  $\mathbf{R}$  be any  $n \times n$  matrix with  $\|\mathbf{F}^{-1}\mathbf{R}\| < 1$ . Then  $(\mathbf{F} + \mathbf{R})^{-1}$  exists and satisfies the inequalities

$$\|(\mathbf{F} + \mathbf{R})^{-1}\| \leq \frac{\|\mathbf{F}^{-1}\|}{1 - \|\mathbf{F}^{-1}\mathbf{R}\|}, \quad (9.7)$$

$$\|(\mathbf{F} + \mathbf{R})^{-1} - \mathbf{F}^{-1}\| \leq \frac{\|\mathbf{R}\| \|\mathbf{F}^{-1}\|^2}{1 - \|\mathbf{F}^{-1}\mathbf{R}\|}. \quad (9.8)$$

To carry out the analysis of the ellipsoid algorithm using approximate arithmetic, we need the following two inequalities for the determinant and the norm of the inverse of a nonsingular matrix repeatedly.

**Remark 9.3** (i) Let  $\mathbf{F}$  be any  $n \times n$  matrix of reals. Then we have the inequality

$$|\det \mathbf{F}| \leq n^{-n/2} \|\mathbf{F}\|^n. \quad (9.9)$$

(ii) If  $\mathbf{F}$  is nonsingular and  $n \geq 2$ , then we have the inequality

$$\|\mathbf{F}^{-1}\| \leq n(n-1)^{-\frac{n-1}{2}} \frac{\|\mathbf{F}\|^{n-1}}{|\det \mathbf{F}|}. \quad (9.10)$$

## 9.2 Ellipsoid “Halving” in Approximate Arithmetic

To carry out the various constructions analytically, we drop the index  $k$  and denote by  $\mathbf{F}$  the nonsingular matrix defining the “current” ellipsoid  $E_0$  and by  $x^0$  its center, i.e.,

$$E_0(x^0, 1) = \{x \in \mathbb{R}^n : \|\mathbf{F}^{-1}(x - x^0)\| \leq 1\}. \quad (9.11)$$

Let  $a^T x \leq a_0$  be the linear inequality that the algorithm identifies and denote by  $x^P$  and  $\mathbf{F}_P$  the updates (9.1) and (9.2) that result if calculated in *perfect* arithmetic, i.e. with an infinite precision:

$$x^P = x^0 - \frac{1}{n+1} \mathbf{F} d \quad \text{where} \quad d = \frac{\mathbf{F}^T a}{\|\mathbf{F}^T a\|} \quad (9.12)$$

$$\mathbf{F}_P = \sqrt{\frac{n^2}{n^2 - 1}} \mathbf{F} \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n-1}{n+1}} \right) \mathbf{d} \mathbf{d}^T \right). \quad (9.13)$$

Assuming that  $\mathbf{F}_P$  is nonsingular, we get in perfect arithmetic an ellipsoid  $E_P$  with center  $\mathbf{x}^P$  which in the iterative scheme of the introduction is the “next” ellipsoid, i.e.

$$E_P(\mathbf{x}^P, 1) = \{ \mathbf{x} \in \mathbb{R}^n : \| \mathbf{F}_P^{-1}(\mathbf{x} - \mathbf{x}^P) \| \leq 1 \}. \quad (9.14)$$

For any “blow-up” factor  $\beta \geq 1$  denote by  $E_\beta$  the enlarged ellipsoid with center  $\mathbf{x}^P$ , i.e.,

$$E_\beta = E_P(\mathbf{x}^P, \beta) = \{ \mathbf{x} \in \mathbb{R}^n : \| \mathbf{F}_P^{-1}(\mathbf{x} - \mathbf{x}^P) \| \leq \beta \}. \quad (9.15)$$

Thus the enlarged ellipsoid  $E_\beta$  is defined with respect to the matrix

$$\mathbf{F}_\beta = \beta \mathbf{F}_P. \quad (9.16)$$

Because of the finite wordlength of the computer we commit round-off errors and compute approximately

$$\mathbf{x}^A \approx \mathbf{x}^P, \quad \mathbf{F}_A \approx \mathbf{F}_\beta. \quad (9.17)$$

Assume that the error in the approximate calculation satisfies

$$\| \mathbf{x}^A - \mathbf{x}^P \| \leq \delta \quad \text{and} \quad \| \mathbf{F}_A - \mathbf{F}_\beta \| \leq \delta \quad \text{where} \quad \delta \leq p(n) \frac{|\det \mathbf{F}|}{\| \mathbf{F} \|^{n-1}}, \quad p(n) = 10^{-4} n^{-2}. \quad (9.18)$$

From inequality (9.9) it follows that the error of the approximation is less than  $\| \mathbf{F} \|$  because by (9.9) e.g.  $\| \mathbf{x}^A - \mathbf{x}^P \| \leq p(n) n^{-n/2} \| \mathbf{F} \|$ . Condition (9.18) is translated in Chapter 9.3 into the number of digits of each component of  $\mathbf{x}^A$  and  $\mathbf{F}_A$  that need to be calculated *correctly* – before and after the “decimal” point in binary arithmetic. The calculations (9.17) yield an approximation  $E_A$

$$E_A(\mathbf{x}^A, 1) = \{ \mathbf{x} \in \mathbb{R}^n : \| \mathbf{F}_A^{-1}(\mathbf{x} - \mathbf{x}^A) \| \leq 1 \}, \quad (9.19)$$

to  $E_\beta$  which is an ellipsoid if  $\mathbf{F}_A$  is “close enough” to  $\mathbf{F}_\beta$  so as to guarantee the nonsingularity of  $\mathbf{F}_A$ .

The “battle-plan” of the proof is as follows:

- Establish that  $\mathbf{F}_P$  is nonsingular if  $\mathbf{F}$  is nonsingular and that (9.3) remains correct.
- Show that if  $E_0$  contains  $P$  or  $P_\Phi$  then so does the ellipsoid  $E_P$ .
- Show that if (9.18) is satisfied neither  $\mathbf{x}^A$  nor  $\mathbf{F}_A$  “grow” too much in size.
- Assuming (9.18) show that  $\mathbf{F}_A$  is nonsingular and that  $E_A \supseteq E_P$ .
- Assuming (9.18) show that  $\frac{\text{vol}(E_A)}{\text{vol}(E_0)}$  satisfies a relation like (9.3) to conclude a polynomial running time of the approximate calculations.
- Establish a lower bound on the volume  $V_{P_\Phi}$  if  $\dim P_\Phi = n$  and prove inductively that (9.18) is satisfied for all necessary iterations. This is done in the next section.

The first step is easy because

$$\det \mathbf{F}_P = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} \det \mathbf{F}. \quad (9.20)$$

The factor appearing in (9.20) satisfies

$$\left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} \leq e^{-\frac{1}{2n}} \quad \text{for all } n \geq 1. \quad (9.21)$$

Computing the volumina of  $E_0$  and  $E_P$  we get from (7.23) and (9.20)

$$\frac{\text{vol}(E_P)}{\text{vol}(E_0)} = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} \leq e^{-\frac{1}{2n}}$$

by (9.21) and (9.3) follows. By (9.20)  $\mathbf{F}_P$  is nonsingular and its inverse in terms of  $\mathbf{F}^{-1}$  is

$$\mathbf{F}_P^{-1} = \sqrt{\frac{n^2 - 1}{n^2}} \left( \mathbf{I}_n - \left(1 - \sqrt{\frac{n+1}{n-1}}\right) \mathbf{d} \mathbf{d}^T \right) \mathbf{F}^{-1}. \quad (9.22)$$

For notational convenience define  $\mathbf{Q} = \mathbf{F} \mathbf{F}^T$  and  $\mathbf{Q}_P = \mathbf{F}_P \mathbf{F}_P^T$ , i.e.  $\mathbf{Q}$  and  $\mathbf{Q}_P$  are the positive definite matrices defining  $E_0$  and  $E_P$ .

$$\mathbf{Q}_P = \frac{n^2}{n^2 - 1} \mathbf{Q} \left( \mathbf{I}_n - \frac{2}{n+1} \frac{\mathbf{a} \mathbf{a}^T \mathbf{Q}}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right), \quad \mathbf{Q}_P^{-1} = \frac{n^2 - 1}{n^2} \left( \mathbf{Q}^{-1} + \frac{2}{n-1} \frac{\mathbf{a} \mathbf{a}^T}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right). \quad (9.23)$$

**Remark 9.4** (i) For all  $\mathbf{x} \in E_0$  and  $\mathbf{a} \in \mathbb{R}^n$ ,  $\mathbf{a} \neq 0$ ,  $-\sqrt{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \leq \mathbf{a}^T (\mathbf{x} - \mathbf{x}^0) \leq \sqrt{\mathbf{a}^T \mathbf{Q} \mathbf{a}}$ .  
(ii) Let  $\mathbf{a}^T \mathbf{x} \leq a_0$  with  $\mathbf{a} \neq 0$  be any inequality such that  $\mathbf{a}^T \mathbf{x}^0 \geq a_0$  and  $\mathcal{X} \subseteq E_0$  be any subset of  $E_0$  such that  $\mathbf{a}^T \mathbf{x} \leq a_0$  for all  $\mathbf{x} \in \mathcal{X}$ . Then  $\mathcal{X} \subseteq E_P$ .

By inductive reasoning it follows that  $P_\Phi \subseteq E_k$  if  $B(\mathbf{0}, R)$  has a large enough radius to contain  $P_\Phi$  initially.

To justify the approximate calculations of  $\mathbf{x}^A$  and  $\mathbf{F}_A$ , we compute like in Exercise 9.2 (iii) from (9.16), (9.13) and (9.22)

$$\|\mathbf{F}_\beta\| \leq \beta \sqrt{\frac{n^2}{n^2 - 1}} \|\mathbf{F}\|, \quad \|\mathbf{F}_\beta^{-1}\| \leq \beta^{-1} \frac{n+1}{n} \|\mathbf{F}^{-1}\|, \quad (9.24)$$

where we have used  $\|\mathbf{d}\| = 1$ , see (9.12). In the following we use the inequality

$$2^x \geq 1 + \frac{2}{3}x + \frac{2}{9}x^2 \quad \text{for all } x \geq 0. \quad (9.25)$$

**Remark 9.5** If (9.18) is satisfied, then for  $\beta = 1 + 1/12n^2$  and all  $n \geq 2$

$$\|\mathbf{x}^A\| \leq \|\mathbf{x}^0\| + \frac{1}{n} \|\mathbf{F}\|, \quad \|\mathbf{F}_A\| \leq 2^{1/n^2} \|\mathbf{F}\|. \quad (9.26)$$

The following inequality is readily verified for all  $n \geq 2$

$$1 + 2(n+1)(n-1)^{-\frac{n-1}{2}} n^{-2} 10^{-4} \leq 1 + \frac{1}{12n^2}. \quad (9.27)$$

**Remark 9.6** If (9.18) is satisfied and  $\det \mathbf{F} \neq 0$ , then for  $\beta = 1 + 1/12n^2$  and all  $n \geq 2$  the matrix  $\mathbf{F}_A$  is nonsingular and  $E_A(\mathbf{x}^A, 1) \supseteq E_P(\mathbf{x}^P, 1)$ .

The main point in the proof that  $E_P(\mathbf{x}^P, 1) \subseteq E_A(\mathbf{x}^A, 1)$  is the estimation

$$\|\mathbf{F}_A^{-1}(\mathbf{x} - \mathbf{x}^A)\| \leq (1 + \|\mathbf{F}_A^{-1}\| \|\mathbf{F}_\beta - \mathbf{F}_A\|)(\|\mathbf{F}_\beta^{-1}(\mathbf{x} - \mathbf{x}^P)\| + \|\mathbf{F}_\beta^{-1}(\mathbf{x}^P - \mathbf{x}^A)\|). \quad (9.28)$$

By Remark 9.4 the feasible set is contained in the approximate ellipsoid if (9.18) is satisfied. The verification of (9.18) will be done by induction.

To estimate next  $\frac{\text{vol}(E_0)}{\text{vol}(E_A)}$  we calculate using (9.4), (9.13), (9.16) and (9.22)

$$\|\mathbf{F}^{-1} \mathbf{F}_\beta\| = \beta \sqrt{n \left(1 - \frac{1}{(n+1)^2}\right)}, \quad \|\mathbf{F}_\beta^{-1} \mathbf{F}\| = \frac{\sqrt{n^3 + n + 2}}{\beta n}. \quad (9.29)$$

**Remark 9.7** If  $\det \mathbf{F} \neq 0$  and (9.18) is satisfied, then for  $\beta = 1 + 1/12n^2$  and for all  $n \geq 2$

$$2^{-1/n} |\det \mathbf{F}| \leq |\det \mathbf{F}_A| \leq 2^{-1/4n} |\det \mathbf{F}|. \quad (9.30)$$

If (9.18) is satisfied at every iteration we can calculate the number of iterations required by the ellipsoid algorithm using approximate arithmetic, i.e. when (9.1) and (9.2) are replaced by (9.17). From (7.23) we compute

$$\frac{\text{vol}(E_A)}{\text{vol}(E_0)} = \frac{|\det \mathbf{F}_A|}{|\det \mathbf{F}|} \leq 2^{-1/4n} < e^{-1/6n}$$

using Remark 9.7 and (9.25). Denoting the ellipsoid at the  $k^{th}$  iteration again by  $E_k$ , we get  $\text{vol}(E_k) < V_0 e^{-k/6n}$  and consequently, after at most

$$k_A = \lceil 6n(\log V_0 - \log V_{P_\Phi}) \rceil$$

iterations the ellipsoid algorithm with approximate arithmetic stops with the message that  $P_\Phi = \emptyset$  – unless for some  $k < k_A$  the corresponding iterate  $\mathbf{x}^k$  belongs to  $P_\Phi$ . Here we have assumed that (9.18) is satisfied at every iteration and that  $P_\Phi \neq \emptyset$  implies  $\dim P_\Phi = n$  and that  $\text{vol}(P_\Phi) \geq V_{P_\Phi}$  – assumptions that we will have to remove later.

### 9.3 Polynomial-Time Algorithms for Linear Programming

Having settled the *analytical* problem of calculating the “perfect” arithmetic ellipsoid approximately by permitting round-off errors let us consider the linear program

$$(LP) \quad \max\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\},$$

where  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$  all have **integer** components only,  $m \geq 1$ ,  $n \geq 2$  and  $\mathbf{A} \neq \mathbf{0}$  to rule out trivialities. The problem (LP) has only inequalities, i.e. any equations have been eliminated or replaced by their corresponding pairs of inequalities.

From Chapter 7.5.1 we know that the integrality assumption is polynomially equivalent to assuming that the data are rational. So the integrality assumption is convenient, but not necessary. Other than this one we make no assumption. The rank of  $A$  can be anywhere between 1 and  $\min\{m, n\}$  and the feasible set

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\} \quad (9.31)$$

can be a solid, flat, pointed or blunt polyhedron of  $\mathbb{R}^n$ . Nonnegativity requirements are part of  $(A, b)$ .

To demonstrate the polynomial solvability of (LP) – polynomial in terms of  $m, n$ , the digital size  $\langle c \rangle$  of the vector  $c$  and the facet complexity  $\phi$  of  $\mathcal{X}$  – we proceed in three steps:

- We assume that  $\mathcal{X}$  is either empty or bounded and full dimensional. By running the “basic” ellipsoid algorithm we decide whether or not  $\mathcal{X} = \emptyset$  by producing a rational vector  $\mathbf{x} \in \mathcal{X}$  if  $\mathcal{X} \neq \emptyset$ .
- We remove the assumptions and show that by embedding any  $\mathcal{X}$  into a somewhat larger polyhedron  $\mathcal{X}_h$  we can answer the question in polynomial time for any rational  $\mathcal{X}$ .
- We use binary search like in Chapter 7.5.3 to prove the existence of a polynomial-time algorithm for the linear program (LP).

To carry out the first step we make the following assumption.

**Assumption A:** There exists  $R \geq 1$ , i.e. some radius, such that  $\mathcal{X} \subseteq B(\mathbf{0}, R)$ . Moreover, if  $\mathcal{X} \neq \emptyset$  then there exist  $\mathbf{x} \in \text{relint}\mathcal{X}$  and a radius  $r > 0$  such that  $B(\mathbf{x}, r) \subseteq \mathcal{X}$ .

To state the basic ellipsoid algorithm, we denote by  $x_j^P, x_j^k$  the components of  $\mathbf{x}^P, \mathbf{x}^k$  and by  ${}_\beta f_j^i, {}_k f_j^i$  the elements of  $\mathbf{F}_\beta, \mathbf{F}_k$  for  $1 \leq i, j \leq n$  and  $k \geq 0$ . The input consists of  $m, n, A$  and  $b$ , the parameters  $R, T$  and  $p$  and if  $\mathcal{X} \neq \emptyset$ ,  $\mathbf{x}$  is the output of the algorithm.

### Basic Ellipsoid Algorithm ( $m, n, R, T, p, A, b, \mathbf{x}$ )

Step 0: Set  $\mathbf{x}^0 := \mathbf{0}$ ,  $\mathbf{F}_0 := RI_n$ ,  $k := 0$ .

Step 1: **if**  $k \geq T$ , **stop** “ $\mathcal{X}$  is empty”.

**if**  $a^i x^k \leq b_i$  for all  $1 \leq i \leq m$ , **stop** “ $\mathbf{x} := \mathbf{x}^k$ ”.

Let  $(a^i, b_i)$  for some  $i \in \{1, \dots, m\}$  be such that  $a^i x^k > b_i$  and set  $a^T := a^i$ .

Step 2: Calculate approximately  $\mathbf{x}^{k+1} \approx \mathbf{x}^P$  and  $\mathbf{F}_{k+1} \approx \mathbf{F}_\beta$  where

$$\mathbf{x}^P := \mathbf{x}^k - \frac{1}{n+1} \frac{\mathbf{F}_k \mathbf{F}_k^T \mathbf{a}}{\|\mathbf{F}_k^T \mathbf{a}\|}, \quad (9.32)$$

$$\mathbf{F}_\beta := \frac{n+1/12n}{\sqrt{n^2-1}} \mathbf{F}_k \left( \mathbf{I}_n - \frac{1 - \sqrt{(n-1)/(n+1)}}{a^T \mathbf{F}_k \mathbf{F}_k^T \mathbf{a}} (\mathbf{F}_k^T \mathbf{a})(\mathbf{a}^T \mathbf{F}_k) \right), \quad (9.33)$$

such that the binary representation of each component of  $\mathbf{x}^{k+1}$  and  $\mathbf{F}_{k+1}$  satisfies  $|x_j^{k+1} - x_j^P| \leq 2^{-p}$  and  $|{}_{k+1} f_j^i - {}_k f_j^i| \leq 2^{-p}$  for  $1 \leq i, j \leq n$ . Replace  $k + 1$  by  $k$  and **go to** Step 1.

**Remark 9.8** (Correctness and finiteness) If Assumption A is true, then the basic ellipsoid algorithm finds a rational vector  $\mathbf{x} \in \mathcal{X}$  or concludes correctly that  $\mathcal{X} = \emptyset$  when it is executed with the parameters

$$T = \lceil 6n^2 \log \frac{R}{r} \rceil, \quad p = 14 + n^2 + \lceil 15n \log \frac{R}{r} \rceil. \quad (9.34)$$

The proof uses Remarks 9.5, 9.6 and 9.7 to estimate *inter alia*

$$\|\mathbf{F}_k\| \leq \sqrt{n}R2^{\frac{k}{n^2}}, \|\mathbf{x}^k\| \leq kR2^{\frac{k}{n^2}}, R^n2^{-\frac{k}{n}} \leq |\det \mathbf{F}_k| \leq R^n2^{-\frac{k}{4n}}. \quad (9.35)$$

We are now ready to drop Assumption A. Denote by  $\phi$  the facet complexity of  $\mathcal{X}$ , i.e.

$$\phi \geq \max_{1 \leq i \leq m} \{\langle \mathbf{a}^i \rangle + \langle \mathbf{b}_i \rangle\}, \quad \phi_A = \max_{1 \leq i \leq m} \langle \mathbf{a}^i \rangle. \quad (9.36)$$

It follows that  $n+1 \leq \phi_A < \phi$  since  $\mathbf{A} \neq \mathbf{O}$  and moreover, we can choose  $\phi$  such that  $\phi \leq \phi_A + \langle \mathbf{b} \rangle$ , where  $\langle \mathbf{b} \rangle$  is the digital size of  $\mathbf{b}$ . For any integer  $h \geq 1$  we denote by  $\mathbf{h}^{-1} \in \mathbb{R}^m$  the vector having  $m$  components equal to  $1/h$  and let

$$\mathcal{X}_h = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b} + \mathbf{h}^{-1}\}, \quad (9.37)$$

which corresponds to “perturbing” the feasible set  $\mathcal{X}$  of (LP).

**Remark 9.9** (i)  $\mathcal{X} \neq \emptyset$  if and only if  $\mathcal{X}_h \neq \emptyset$  for all  $h \geq p2^{p\phi_A}$  where  $p = 1 + \min\{m, n\}$ .  
(ii) If  $\mathcal{X} \neq \emptyset$ , then for all  $u \geq 2^{n\phi}$  and all integers  $h \geq 1$  the set  $\mathcal{X}_h^u$  of solutions to

$$\mathbf{Ax} \leq \mathbf{b} + \mathbf{h}^{-1}, -u - 1/h \leq x_j \leq u + 1/h, \text{ for } 1 \leq j \leq n \quad (9.38)$$

is bounded, full dimensional and  $B(\mathbf{x}, r_h) \subseteq \mathcal{X}_h^u$  for all  $\mathbf{x} \in \mathcal{X}$  where

$$r_h = h^{-1}2^{-\phi_A+n}. \quad (9.39)$$

To visualize the construction of Remark 9.9 take for instance

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 : 1 \leq x_1 \leq 1, 1 \leq x_2 \leq 1\}$$

and bring it into the form (9.37). Graphing the corresponding solution set, one sees that the introduction of the perturbation  $1/h$  in each inequality corresponds to a “tearing apart” of the solution set to obtain a full dimensional set of solutions; see also Figure 9.8 below. If  $\mathcal{X}$  is empty then there is nothing to tear apart and as the first part of Remark 9.9 shows, the emptiness of  $\mathcal{X}_h$  is preserved if the perturbation is “small enough”. Running the basic ellipsoid algorithm with  $R = 2^{n\phi}$ ,  $T = 20n^3\phi$  and  $p = 55n^2\phi$  we conclude like in point 7.6(g):

**Remark 9.10** Every  $m \times n$  system of linear inequalities  $\mathbf{Ax} \leq \mathbf{b}$  with rational data can be “solved” in time that is polynomial in the digital size of its input.

There are several ways to deal with the optimization aspect of the linear program (LP). The simplest way is to use linear programming duality and to reduce the optimization problem to the problem of finding a solution to a system of linear inequalities – like we did in Remark 6.5. The second way uses binary search and a “sliding objective” function; see the text.

**Remark 9.11** Every linear program with rational data can be “optimized” in time that is polynomial in the digital size of its input.

Neither the radius  $R$  of the ball circumscribing  $\mathcal{X}$  or  $\mathcal{X}_h^u$  nor the radius  $r$  of the ball that is inscribed into  $\mathcal{X}$  or  $\mathcal{X}_h^u$ , see (9.39), depend on the number  $m$  of linear inequalities of (LP). Consequently, none of the other two parameters  $T$  and  $p$  of the basic ellipsoid algorithm depends

on the number  $m$ . They are polynomial functions of  $\phi$ ,  $n$  and  $\langle c \rangle$  only. The dependence of the basic ellipsoid algorithm on the number  $m$  of the inequalities of (LP) enters in Step 1 when we have to find a violated inequality for the system (9.43) or prove that none exists. The same is true for the auxiliary computations. For the time being we assume that we *find* violated inequalities by the “algorithm” LIST-and-CHECK that we discussed in the introduction. If  $m$  is of the order of  $n$ , i.e.  $m = \mathcal{O}(n)$ , then the total effort to solve (LP) becomes a polynomial function of  $n$ ,  $\phi$  and  $\langle c \rangle$  only, whereas in the general case we need, of course, note the dependence on  $m$  explicitly. Before coming back to the question of how to deal with the case of possibly exponentially many constraints defining  $\mathcal{X}$  we first discuss some “practical” variants of the basic ellipsoid algorithm.

## 9.4 Deep Cuts, Sliding Objective, Large Steps, Line Search

Going back to Figure 9.1 we see that instead of cutting  $E_k$  with  $a^T x \leq a_0$  – where the right-hand side equals  $a_0$  and which is valid for all  $x \in \mathcal{X}$  – we replaced  $a_0$  by the larger quantity  $a^T x^k$ . This replacement forces the cut to pass through the *center* of the current ellipsoid and the resulting algorithm is therefore called **central cut** ellipsoid algorithm. It is not overly difficult to work out the formulas corresponding to (9.1) and (9.2) when instead of  $a^T x \leq a^T x^k$  we use the **deep cut**  $a^T x \leq a_0$ . They are given below.

A less obvious modification of the basic algorithmic idea concerns the optimization aspect of (LP). Let

$$z = \max\{cx^k : x^k \text{ feasible}\},$$

where initially  $z = -\infty$ . Then we can use the objective function as a “sliding constraint” of the form  $cx \geq z$  where the value of  $z$  increases during the course of the calculations. This gives rise to a **sliding objective** and thereby to a device that speeds the convergence of the procedure considerably.

A third modification to the basic idea goes as follows. Suppose the current iterate  $x^k$  is feasible. Then the point  $x^* = x^k + F_k F_k^T c^T / \|cF_k\|$  maximizes the linear function  $cx$  over the current ellipsoid  $E_k = \{x \in \mathbb{R}^n : \|F_k^{-1}(x - x^k)\| \leq 1\}$ ; see Remark 9.4(i). Consequently, we can determine by a *least ratio test* the largest  $\lambda \geq 0$  such that

$$x(\lambda) = x^k + \lambda(x^* - x^k) \text{ is feasible}$$

and thereby make a **large step** towards optimality by “shooting” through the interior of the feasible set. We calculate the largest  $\lambda$ , the corresponding feasible  $x(\lambda)$  and its objective function value  $z_\lambda$ , say. If  $z_\lambda > z$ , then we update the current best solution to be  $x(\lambda)$ , replace  $z$  by  $z_\lambda$  and use in one of the subsequent iterations the objective function as a cut to reduce the volume of the ellipsoid.

The fourth modification of the basic algorithmic idea is aimed at improving the chances of the algorithm to find *feasible* solutions to (LP). The algorithm generates a sequence of individual points and their probability to fall into the feasible set is rather small. Consider two consecutive centers  $x^k$  and  $x^{k+1}$  of the ellipsoids generated by the algorithm. They determine a line

$$x(\mu) = (1 - \mu)x^k + \mu x^{k+1} \text{ where } -\infty < \mu < +\infty.$$

We can decide the question of whether or not  $x(\mu)$  *meets* the feasible set by a **line search** that involves again a simple least ratio test. If the test is negative, we continue as we would do without

it. If the test comes out positive, then we get an interval  $[\mu_{\min}, \mu_{\max}]$  such that  $\mathbf{c}\mathbf{x}(\mu)$  is feasible for all  $\mu$  in the interval. Computing the objective function we find  $\mathbf{c}\mathbf{x}(\mu) = \mathbf{c}\mathbf{x}^k + \mu(\mathbf{c}\mathbf{x}^{k+1} - \mathbf{c}\mathbf{x}^k)$ . Consequently, if  $\mathbf{c}\mathbf{x}^{k+1} > \mathbf{c}\mathbf{x}^k$  then  $\bar{\mu} = \mu_{\max}$  yields the best possible solution vector while  $\bar{\mu} = \mu_{\min}$  does so in the opposite case. The rest is clear: we proceed like we did in the case of large steps.

We are now ready to state an ellipsoid algorithm for linear programs in canonical form

$$(LP_C) \quad \max\{\mathbf{c}\mathbf{x} : \tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}, \mathbf{x} \geq \mathbf{0}\},$$

where  $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}})$  is an  $m \times (n+1)$  matrix of rationals. We assume that  $\tilde{\mathbf{A}}$  contains no zero row and denote by  $(\mathbf{a}^i, b_i)$  for  $1 \leq i \leq m+n$  the rows of the matrix  $(\mathbf{A}, \mathbf{b}) = \begin{pmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{b}} \\ -I_n & \mathbf{0} \end{pmatrix}$ .

The DCS ellipsoid algorithm takes  $m, n, \mathbf{A}, \mathbf{b}, \mathbf{c}$  as inputs.  $z_L$  is a lower bound,  $z_U$  an upper bound on the optimal objective function value.  $R$  is a common upper bound on the variables of  $(LP_C)$  and  $\varepsilon$  a perturbation parameter to ensure full dimensionality of the feasible set when intersected with the sliding objective function constraint  $\mathbf{c}\mathbf{x} \geq z$ . Since we are perturbing the constraint set of  $(LP_C)$  by a parameter  $\varepsilon > 0$  we shall call solutions to the perturbed constraint set nearly feasible or  $\varepsilon$ -feasible solutions and correspondingly, we shall utilize the term  $\varepsilon$ -optimal solution to denote a nearly feasible, nearly optimal solution to  $(LP_C)$ .  $V_F$  is a positive lower bound on the volume of a full dimensional,  $\varepsilon$ -optimal set. In other words, if the current ellipsoid has a volume less than  $V_F$  we shall conclude that either  $\varepsilon$ -optimality is attained – if a feasible solution  $\bar{\mathbf{x}}$  with objective function value  $z$  was obtained – or else that the feasible set of  $(LP_C)$  is empty. As we know from Chapters 7.5 and 9.3 we can always find theoretical values for  $\varepsilon$  and  $R$  and by consequence for  $z_L, z_U$  and  $V_F$  as well that the algorithm needs to converge.

In practice, we set the perturbation parameter e.g.  $\varepsilon = 10^{-4}$  and use a rough data dependent estimate for the common upper bound  $R$  on the variables. Similarly, we use e.g.  $V_F = 10^{-2}$  to fix the stopping criterion and from  $R$  we estimate  $z_L$  and  $z_U$  e.g. as follows

$$z_L = -1 + nc^- R, \quad z_U = 1 + nc^+ R,$$

where  $c^- = \min\{c_j : 1 \leq j \leq n\}$  and  $c^+ = \max\{c_j : 1 \leq j \leq n\}$ .

“DCS” stands for deep cut, sliding objective, large steps and line search, i.e. all of the devices that we discussed above to speed the empirical rate of convergence of the underlying basic algorithmic idea. For this “practical” version of the ellipsoid algorithm we ignore the blow-up factor  $\beta \geq 1$  that is necessary to obtain the theoretical result since  $\beta - 1 = 1/12n^2$  is a horribly small positive number for reasonably sized  $n$ .

In the DCS ellipsoid algorithm we assume  $\mathbf{c} \neq 0$ . If  $\mathbf{c} = 0$  then some modifications and simplifications impose themselves the details of which we leave as an exercise for you to figure out.

#### DCS Ellipsoid Algorithm ( $m, n, z_L, z_U, \varepsilon, R, V_F, \mathbf{A}, \mathbf{b}, \mathbf{c}, \bar{\mathbf{x}}, z$ )

**Step 0:** Set  $k := 0$ ,  $x_j^0 := R/2$  for  $1 \leq j \leq n$ ,  $z := z_L$ ,  $z_0 := z_L$ ,

$$R_0 := \sqrt{n}(1 + R/2), \quad H_0 := R_0 I_n, \quad f_0 := \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}}, \quad V_0 := R_0^n \pi^{n/2} / \Gamma(1 + n/2).$$

**Step 1:** Set  $mxv := b_j + \varepsilon - \mathbf{a}^j \mathbf{x}^k$  where  $b_j - \mathbf{a}^j \mathbf{x}^k \leq b_i - \mathbf{a}^i \mathbf{x}^k$  for all  $1 \leq i \leq n+m$ .

**if**  $mxv < 0$  **go to** Step 2.

Set  $\mathbf{x}^* := \mathbf{x}^k + H_k H_k^T \mathbf{c}^T / \|\mathbf{c} H_k\|$ ,  $\lambda := \max\{\lambda : \lambda \mathbf{a}^i (\mathbf{x}^* - \mathbf{x}^k) \leq b_i + \varepsilon - \mathbf{a}^i \mathbf{x}^k, 1 \leq i \leq n+m\}$ .

**if**  $\lambda \geq 1$  **stop** “ $(LP_C)$  is unbounded.”

**if**  $c(\mathbf{x}^k + \lambda(\mathbf{x}^* - \mathbf{x}^k)) \leq z$  **go to** Step 2.

Set  $\bar{\mathbf{x}} := \mathbf{x}^k + \lambda(\mathbf{x}^* - \mathbf{x}^k)$ ,  $z := c\bar{\mathbf{x}}$ .

**Step 2:** **if**  $(c\mathbf{x}^k > z \text{ or } (m\mathbf{x}\mathbf{v} < 0 \text{ and } z_0 - z > m\mathbf{x}\mathbf{v}))$  **then**

Choose  $\Theta$  so that  $b_j + \varepsilon \leq \Theta \leq \mathbf{a}^j \mathbf{x}^k$ . Set  $\alpha_k := \frac{\mathbf{a}^j \mathbf{x}^k - \Theta}{\|\mathbf{a}^j \mathbf{H}_k\|}$ ,  $\mathbf{r}^T := \mathbf{a}^j$ .

**else**

Set  $\alpha_k := (z - c\mathbf{x}^k)/\|\mathbf{c}\mathbf{H}_k\|$ ,  $\mathbf{r}^T := -\mathbf{c}$ ,  $z_0 := z$ .

**endif**

**Step 3:** **if**  $\alpha_k < 1$  and  $V_k \geq V_F$  **go to** Step 4.

**if**  $z_L < z < z_U$  **stop** “ $\bar{\mathbf{x}}$  is an  $\varepsilon$ -optimal solution to  $(LP_C)$ .”

**stop** “ $(LP_C)$  is infeasible or unbounded.”

**Step 4:** Set

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \frac{1 + n\alpha_k}{(n+1)\|\mathbf{H}_k^T \mathbf{r}\|} \mathbf{H}_k \mathbf{H}_k^T \mathbf{r}, \quad (9.44)$$

$$\mathbf{H}_{k+1} := n\sqrt{\frac{1 - \alpha_k^2}{n^2 - 1}} \mathbf{H}_k \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}} \right) \frac{(\mathbf{H}_k^T \mathbf{r})(\mathbf{r}^T \mathbf{H}_k)}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right), \quad (9.45)$$

$$V_{k+1} := (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k) f_0 V_k. \quad (9.46)$$

Let  $I := \{\mu \in \mathbb{R} : \mu \mathbf{a}^i(\mathbf{x}^{k+1} - \mathbf{x}^k) \leq b_i + \varepsilon - \mathbf{a}^i \mathbf{x}^k \text{ for } 1 \leq i \leq n+m\}$ .

**if**  $I \neq \emptyset$  and  $c\mathbf{x}^k \neq c\mathbf{x}^{k+1}$  **then**

**if**  $c\mathbf{x}^{k+1} > c\mathbf{x}^k$  **then** set  $\bar{\mu} := \max\{\mu : \mu \in I\}$  **else**  $\bar{\mu} := \min\{\mu : \mu \in I\}$ .

**if**  $|\bar{\mu}| = \infty$  **stop** “ $(LP_C)$  is unbounded.”

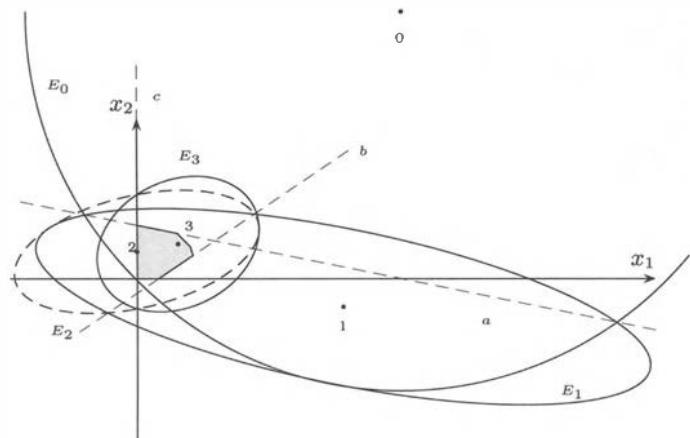
**if**  $c\mathbf{x}^k + \bar{\mu}(c\mathbf{x}^{k+1} - c\mathbf{x}^k) > z$  **then** set  $\bar{\mathbf{x}} := \mathbf{x}^k + \bar{\mu}(\mathbf{x}^{k+1} - \mathbf{x}^k)$ ,  $z := c\bar{\mathbf{x}}$ .

**endif**

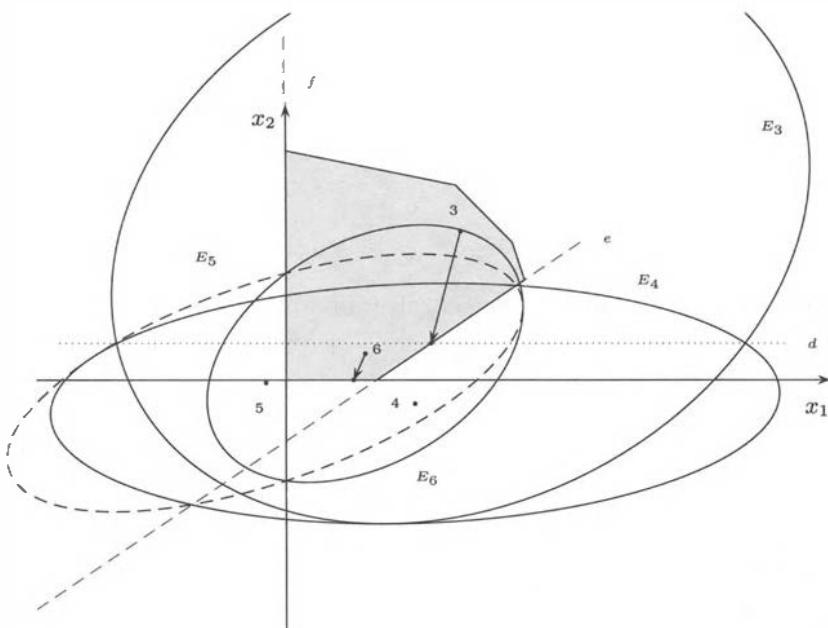
Replace  $k+1$  by  $k$  and **go to** Step 1.

### 9.4.1 Linear Programming the Ellipsoidal Way: Two Examples

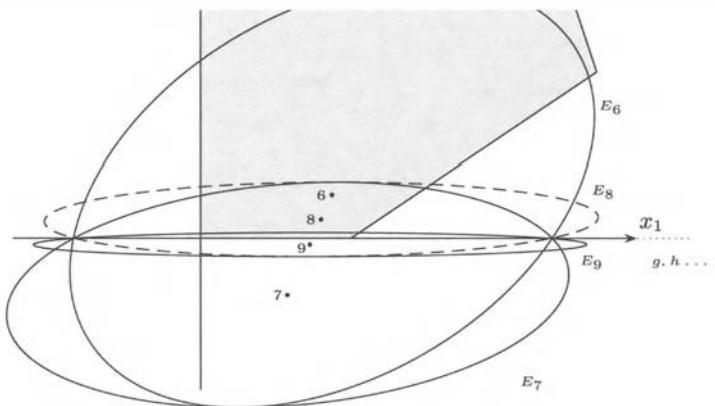
In Figures 9.2, 9.3, 9.4 we show the first nine iterations that result when we use the data of Exercise 8.2 (ii) and minimize  $x_2$  *without* the use of the line search, i.e. we assume in Step 4 that always  $I = \emptyset$ . In Figures 9.5, 9.6, 9.7 we show the corresponding 12 first iterations when we maximize  $x_2$  *with* line search. To make the corresponding pictures more readable we have depicted every *third* ellipse by a “dashed” curve, whereas all the others are drawn solidly. The first ellipse shown in Figures 9.3, 9.4 and Figures 9.6, 9.7, respectively, are the “last” ellipse of the respective preceding picture. “Dashed” lines correspond to using the original constraints to cut the ellipse in half, while “dotted” lines correspond to cuts using the sliding objective. In Figure 9.3 the arrows show the “large” steps that the algorithm takes, while there are none in Figures 9.5, 9.6, 9.7. Note that in Step 2 we use – like in Step 1 – a “most violated” constraint: a sliding objective cut is executed only if the objective cut is a most violated constraint. Note the different convergence behavior of the algorithm that results from the existence of alternative optima; see the text for a detailed discussion of the computer runs.



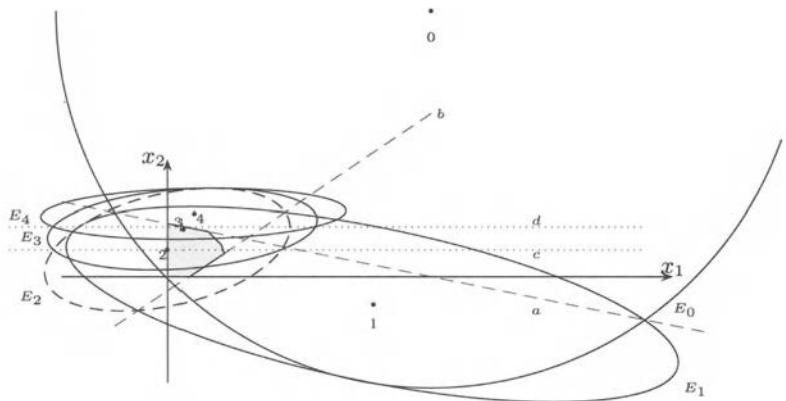
**Fig. 9.2.** Deep cuts, sliding objective, large steps (minimize  $x_2$ )



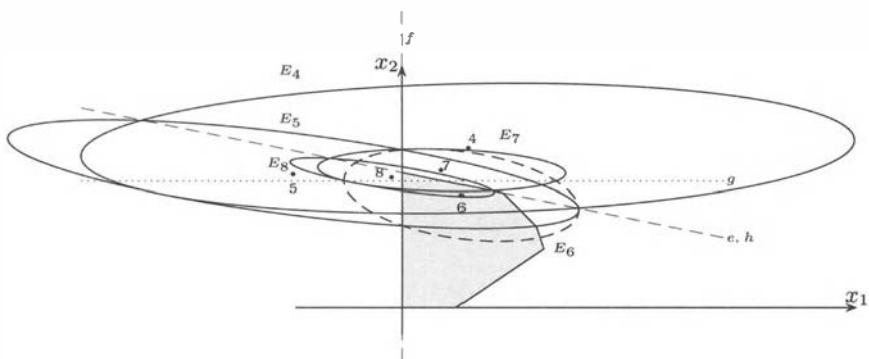
**Fig. 9.3.** Deep cuts, sliding objective, large steps for iterations  $3, \dots, 6$



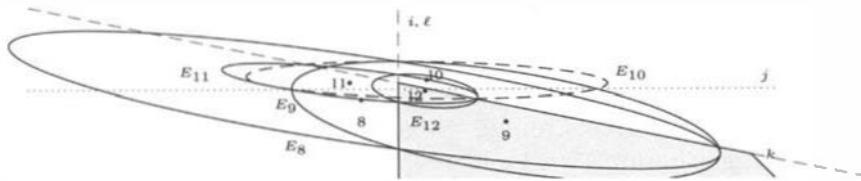
**Fig. 9.4.** Proving optimality of a face of dimension 1 in  $\mathbb{R}^2$  the ellipsoidal way



**Fig. 9.5.** Deep cuts, sliding objective, line search (maximize  $x_2$ )



**Fig. 9.6.** Deep cuts, sliding objective, line search for iterations  $4, \dots, 8$



**Fig. 9.7.** Proving optimality of a face of dimension 0 in  $\mathbb{R}^2$  the ellipsoidal way

#### 9.4.2 Correctness and Finiteness of the DCS Ellipsoid Algorithm

By assumption the parameter  $R$  is a common upper bound on the variables of  $(LP_C)$  that is large enough so that the hypercube

$$\{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_j \leq R \text{ for } 1 \leq j \leq n\}$$

contains all of the feasible set of  $(LP_C)$  if  $(LP_C)$  is bounded and enough of the unbounded portion of the feasible set to permit us to conclude unboundedness via the value of the objective function; see point 7.5(d) and the discussion of binary search in Chapter 9.3. We start the algorithm at the center  $\mathbf{x}^0 = \frac{1}{2}R\mathbf{e}$  of this hypercube and by choice of  $R_0$  in Step 0 the initial ball  $B(\mathbf{x}^0, R_0)$  does the job.

The validity of the DCS ellipsoid algorithm is established inductively like in Chapter 9.2. Using formula (9.45) for the update  $\mathbf{H}_{k+1}$  we compute its determinant in terms of the determinant  $\mathbf{H}_k$

$$\det \mathbf{H}_{k+1} = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k) \det \mathbf{H}_k. \quad (9.47)$$

To establish the containment of the feasible set in the updated ellipsoid, we form the positive definite matrix  $\mathbf{G}_{k+1} = \mathbf{H}_{k+1} \mathbf{H}_{k+1}^T$  and compute its inverse

$$\mathbf{G}_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left( \mathbf{G}_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1 - \alpha_k)} \frac{\mathbf{r}\mathbf{r}^T}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right), \quad (9.48)$$

see Exercise 9.6. Let  $E_k = E_k(\mathbf{x}^k, 1)$  be the ellipsoid that the DCS algorithm constructs at iteration  $k$

$$E_k(\mathbf{x}^k, 1) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \mathbf{x}^k)^T \mathbf{G}_k^{-1} (\mathbf{x} - \mathbf{x}^k) \leq 1\}. \quad (9.49)$$

It follows using (7.23) from (9.47) that

$$\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} = \left( \frac{1 - \alpha_k}{1 + 1/n} \right)^{\frac{n+1}{2}} \left( \frac{1 + \alpha_k}{1 - 1/n} \right)^{\frac{n-1}{2}} \leq e^{-\alpha_k - \frac{1}{2n}}$$

for all  $k \geq 0$ . Setting  $V_{k+1} = \text{vol}(E_{k+1})$  and  $f_0 = (1 - 1/n)^{-\frac{n+1}{2}} (1 + 1/n)^{-\frac{n-1}{2}}$  we get

$$V_{k+1} = V_0 f_0 \prod_{\ell=0}^k (1 - \alpha_\ell^2)^{\frac{n-1}{2}} (1 - \alpha_\ell), \quad (9.50)$$

which shows that the DCS algorithm updates the volume of the current ellipsoid correctly in formula (9.46). Moreover, it shows the “deflating” effect of the deep cuts on the volume of the ellipsoid quite clearly.

It is shown in the text that the DCS ellipsoid algorithm is correct if the bound  $R$  is “large enough” and the perturbation  $\varepsilon$  is “small enough”. From (9.50) and the formula for the ratio of the volumina it follows that the stopping criterion  $V_k < V_F$  is satisfied after at most

$$\lceil 2n \log \frac{V_0}{V_F} \rceil$$

iterations. The  $\alpha_\ell$ 's introduce a data-dependency into the stopping criterion that does, however, not change the theoretical worst-case behavior of the algorithm.

## 9.5 Optimal Separators, Most Violated Separators, Separation

”Ηξεις ἀρίξεις οὐ θνήξεις ἐν πολέμῳ.<sup>2</sup>  
Pythia, High priestess of Delphi.

Throughout the rest of this chapter we will deal mostly with rational polytopes  $P \subseteq \mathbb{R}^n$  rather than with polyhedra. From Chapter 7.5 we know that we can always do so while preserving polynomiality of the facet complexity and vertex complexity in terms of the original parameters. We are interested in the linear optimization problem  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$  where the vector  $\mathbf{c}$  is some row vector with rational coefficients and  $P$  has a linear description with possibly exponentially many linear inequalities. To approach this problem we start with a *partial* linear description of the rational polytope  $P$  having  $\mathcal{O}(n)$  constraints, which gives us a larger polytope  $P_0$  that contains  $P$ . Solving the linear program

$$\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P_0\}$$

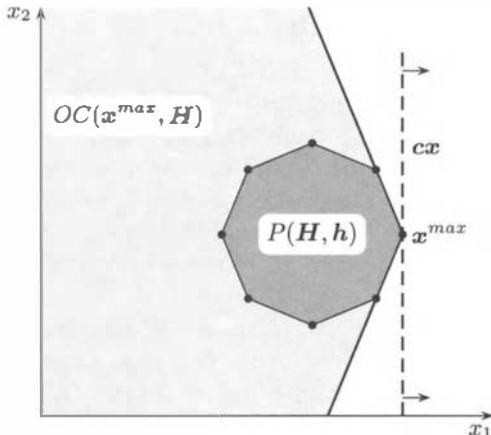
we either conclude that  $P_0$  and thus  $P$  is empty or we get an optimal solution  $\mathbf{x}^0 \in P_0$ , e.g. an optimal extreme point of  $P_0$ . Now we check the constraint set of  $P$  by *some* “separation algorithm” - other than LIST-and-CHECK - to find a violated constraint, i.e. we solve something like the separation problem of Chapter 7.5.4 *algorithmically*. If the separation problem does not produce a violated constraint, then  $\mathbf{x}^0$  is an optimal solution to the problem  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$  – see the *outer inclusion principle* of Chapter 7.5.4 which works with a “local” description of  $P$  in the neighborhood of a maximizer  $\mathbf{x}^{max}$ , say, of  $\mathbf{c}\mathbf{x}$  over  $P$  rather than the *complete* linear description of  $P$ .

To stress the point that we wish to make once again, suppose that the feasible set of our linear program is given by a *convex polygon* in  $\mathbb{R}^2$  with, say,  $10^{10^{10}}$  or more “corners” and just as many facets: all you need are at most **three** of its facet defining constraints to *prove* optimality of some corner that maximizes your *linear* function in  $\mathbb{R}^2$ , see Figure 9.8. The problem is to find the “right” ones and almost nothing else matters.

Suppose that the separation algorithm finds a constraint  $\mathbf{h}^1 \mathbf{x} \leq h_0^1$ , say, such that  $\mathbf{h}^1 \mathbf{x}^0 > h_0^1$ . Then

$$P \subseteq P_1 = P_0 \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}^1 \mathbf{x} \leq h_0^1\} \subset P_0$$

<sup>2</sup>“You will depart, you will arrive, you will not die in the war” or is it “you will depart, you will not arrive, you will die in the war”?



**Fig. 9.8.** The outer inclusion principle in  $\mathbb{R}^2$

and we can iterate. The question: does this iterative scheme converge “fast enough” to permit linear optimization over any rational polytope in polynomial time?

This is where the (basic) ellipsoid algorithm enters: neither its running time  $T$  nor the required precision  $p$ , see e.g. (9.34), depend on the number  $m$  of the constraints of the linear program. So if we can find a constraint defining  $P$  that is violated by the *current* iterate  $x^k$  or prove that no such constraint exists in time that is polynomial in  $n$ ,  $\phi$ ,  $\langle x^k \rangle$  and  $\langle c \rangle$ , then the polynomiality of the entire iterative scheme follows from the polynomiality of the ellipsoid algorithm.

Let us discuss first *what kind* of a violated constraint we wish to find *ideally* to obtain “fast” convergence of this iterative scheme. We shall forget what other authors have called “the separation problem” and first determine what it is that we really want.

Denote by  $P_k$  the polytope that we have after  $k$  iterations and by  $x^k$  the current optimizer. Denote by

$$SP = \{(\mathbf{h}, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : \mathbf{h}x \leq h_0\}\} \quad (9.51)$$

the set of all candidates for a solution of the separation problem e.g. as defined in Chapter 7.5.4. The set  $SP$  is the  $h_0$ -polar of the polytope  $P$ , see (7.13) in Chapter 7.4 where  $Y$  is void because  $P$  by assumption is a polytope. Ideally, we wish to find a constraint that “moves” the objective function “down” as fast as possible because we know that  $P \subseteq P_k$ . So we want ideally a solution to the problem

$$\min_{(\mathbf{h}, h_0) \in SP} \max_{\mathbf{x} \in \mathbb{R}^n} \{\mathbf{c}\mathbf{x} : \mathbf{x} \in P_k \cap \{x \in \mathbb{R}^n : \mathbf{h}x \leq h_0\}\}. \quad (9.52)$$

Since  $P$  is a polytope this min-max problem has a solution if  $P \neq \emptyset$  and if  $P = \emptyset$  we simply declare an arbitrary “violated” inequality to be the solution, e.g.  $\mathbf{h}x = \mathbf{c}x \leq h_0 = \mathbf{c}x^k - 100$ . If the objective function value of (9.52) is greater than or equal to  $\mathbf{c}x^k$ , then by the outer inclusion principle the current iterate  $x^k$  solves the linear optimization problem  $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$  and we are done. Otherwise, the objective function value is less than  $\mathbf{c}x^k$  and let us call any  $(\mathbf{h}, h_0) \in \mathbb{R}^{n+1}$  that solves (9.52)

an **optimal separator** for  $P$  with respect to the objective function  $cx$

that we wish to maximize over  $P$ . It follows that  $hx^k > h_0$  and we can iterate. What we like to have ideally is not necessarily what we can do in computational practice and indeed, we are not aware of any linear optimization problem for which a solution to the min-max problem (9.52) is known. We refer to this problem sometimes as the problem of “finding the right cut” because we are evidently cutting off a portion of the polytope  $P_k$  by an optimal separator  $hx \leq h_0$  and the cut is “right” because it moves the objective function value as much as possible. A general solution to (9.52) does not seem possible, but for certain classes of optimization problems an answer to this problem may be possible.

Since a solution to (9.52) seems elusive, we have to scale down our aspirations somewhat and approximate the problem of finding the right cut. The next best objective that comes to one’s mind is to ask for  $(h, h_0) \in SP$  such that the amount of violation  $hx^k - h_0$  is maximal. So we want to solve

$$\max\{hx^k - h_0 : (h, h_0) \in SP, \|h\|_\infty = 1\}, \quad (9.53)$$

where  $\|h\|_\infty = \max\{|h_j| : 1 \leq j \leq n\}$  is the  $\ell_\infty$ -norm. Because we normalized by  $\|h\|_\infty = 1$  the maximum in (9.53) exists if  $P \neq \emptyset$ . If  $P = \emptyset$ , then any inequality  $hx \leq h_0$  with  $hx^k - h_0 > 0$  is simply declared to be a “solution” to (9.53). If the objective function value in (9.53) is less than or equal to zero, we stop: the current iterate  $x^k$  maximizes  $cx$  over  $P$ . Otherwise, any solution to (9.53) is

a **most violated separator** for  $P$ .

It follows that  $hx^k > h_0$  and we can iterate. Indeed, in all of the computational work that preceded as well as that followed the advent of the ellipsoid algorithm the constraint identification (or separation) problem was approached in this and no other way. Posing the separation problem that we need to solve iteratively the way we have done it solves the separation problem of Chapter 7.5.4: the objective function value of (9.53) provides a “proof” that all constraints of  $P$  are satisfied by the current iterate  $x^k$  of the overall iterative scheme. Of course, problem (9.53) has one nonlinear constraint, but we can get around the nonlinearity easily.

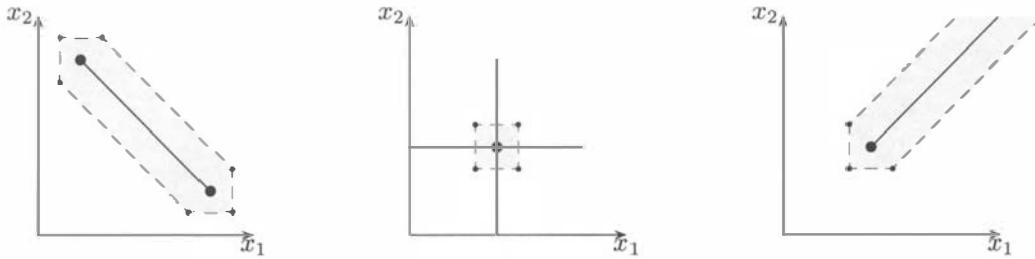
We consider in the **separation step** of the basic ellipsoid algorithm, i.e. in Step 1, only *most violated* separators. This agrees not only with computational practice, but it also alleviates certain theoretical difficulties that arise when the separation step is not treated as an optimization problem on its own.

If one merely asks for *some* separator as we do in the first part of point 7.5(j), then – in case that  $\dim P = k < n$  – it can happen that the separation subroutine always returns a hyperplane that is parallel to the affine hull of  $P$ , see e.g. Figure 9.4, and, of course, the volumina of the ellipsoids tend to zero. Thus if the volume falls below a certain value  $V_F$ , say, we can no longer conclude that  $P = \emptyset$  because – even though  $P \neq \emptyset$  – the  $n$ -dimensional volume of  $P$  is zero if  $\dim P < n$ .

## 9.6 $\varepsilon$ -Solidification of Flats, Polytopal Norms, Rounding

For any  $\varepsilon > 0$  define the  **$\varepsilon$ -solidification**  $P_\varepsilon^\infty$  of a polytope  $P \subseteq \mathbb{R}^n$  with respect to the  $\ell_\infty$ -norm by

$$P_\varepsilon^\infty = \{z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_\infty \leq \varepsilon\}, \quad (9.54)$$



**Fig. 9.9.**  $\varepsilon$ -Solidification (9.54) with  $\varepsilon = 0.5$  of three rational flats in  $\mathbb{R}^2$

where  $\|x - z\|_\infty = \max\{|x_j - z_j| : 1 \leq j \leq n\}$  is the  $\ell_\infty$ -norm.

**Remark 9.12** (i) For every  $\varepsilon > 0$  and nonempty polytope  $P \subseteq \mathbb{R}^n$  the set  $P_\varepsilon^\infty \subseteq \mathbb{R}^n$  is a full dimensional polytope.

(ii) If  $hx \leq h_0$  for all  $x \in P$ , then  $hx \leq h_0 + \varepsilon \|h\|_1$  for all  $x \in P_\varepsilon^\infty$ . If  $hx \leq h_0$  for all  $x \in P_\varepsilon^\infty$ , then  $hx \leq h_0 - \varepsilon \|h\|_1$  for all  $x \in P$ , where  $\|h\|_1 = \sum_{j=1}^n |h_j|$  is the  $\ell_1$ -norm.

(iii) Let  $Hx \leq h$  be any linear description of  $P_\varepsilon^\infty$ . Then  $Hx \leq h - \varepsilon d$  is a linear description of  $P$  where  $d$  is the vector of the  $\ell_1$ -norms of the rows of  $H$ .

Suppose  $P$  is a rational flat. Then  $P \subseteq \{x \in \mathbb{R}^n : hx = h_0\}$  for some  $(h, h_0) \in \mathbb{R}^{n+1}$ . Consequently,  $hx \leq h_0$  and  $-hx \leq -h_0$  for all  $x \in P$ . Thus

$$P_\varepsilon^\infty \subseteq \{x \in \mathbb{R}^n : hx \leq h_0 + \varepsilon \|h\|_1, -hx \leq -h_0 + \varepsilon \|h\|_1\},$$

which corresponds to “tearing apart” the equations defining the affine hull of  $P$ . Since for every  $x \in P$  the hypercube  $x + \{z \in \mathbb{R}^n : |z_j| \leq \varepsilon \text{ for } 1 \leq j \leq n\}$  is contained in  $P_\varepsilon^\infty$  and this hypercube contains  $B(x, r = \varepsilon)$  the  $n$ -dimensional volume of  $P_\varepsilon^\infty$  satisfies

$$\text{if } P \neq \emptyset \text{ then } \text{vol}(P_\varepsilon^\infty) \geq 2^n \varepsilon^n > \frac{\varepsilon^n \pi^{n/2}}{\Gamma(1 + n/2)} \text{ for all } \varepsilon > 0. \quad (9.55)$$

Since  $\dim P_\varepsilon^\infty = n$  if  $P \neq \emptyset$  the polytope  $P_\varepsilon^\infty$  has a linear description that is unique modulo multiplication by positive scalars. Thus if  $Hx \leq h$  is some linear description of  $P$  then the set  $\{x \in \mathbb{R}^n : Hx \leq h + \varepsilon d\}$  contains  $P_\varepsilon^\infty$  by Remark 9.12 (ii), where  $d$  is defined in part (iii). But the containment can be proper.

Let  $x^* \in \mathbb{R}^n$  be arbitrary and suppose that we have a **separation subroutine** for the polytope  $P$  that finds a most violated separator  $hx \leq h_0$ . If  $hx^* \leq h_0$ , then  $x^* \in P_\varepsilon^\infty$  where  $\varepsilon > 0$  is arbitrary. From Remark 9.12 it follows that for a point outside of  $P_\varepsilon^\infty$  there exists at least one of the representations of some facet of  $P$  that is violated by it; see the text for more detail.

**Remark 9.13** Let  $Hx \leq h$  and  $\phi \geq n + 1$  be such that  $\langle h^i \rangle + \langle h_i \rangle \leq \phi$  for all rows  $(h^i, h_i)$  of  $(H \ h)$ . Let  $x^* \in \mathbb{R}^n$  and  $(H_1 \ h^1), (H_2 \ h^2)$  be a partitioning of  $(H \ h)$  such that

$$h^1 - \varepsilon d^1 \leq H_1 x^* \leq h^1 + \varepsilon d^1, \quad H_2 x^* \leq h^2 - \varepsilon d^2, \quad (9.56)$$

where  $d^1, d^2$  are the vectors of the  $\ell_1$ -norms of the corresponding rows of  $H_1, H_2$ . If  $0 \leq \varepsilon \leq 2^{-5(n+1)\phi}$  then the system  $H_1 x = h^1, H_2 x \leq h^2$  is solvable.

In Figure 9.9 we have illustrated the  $\varepsilon$ -solidification of three flats in  $\mathbb{R}^2$ . Exercise 9.7 shows that solidification works for polyhedra and states various facts about  $P_\varepsilon^\infty$ . It shows also that we can replace the  $\ell_\infty$ -norm in the definition of the  $\varepsilon$ -solidification of a polyhedron  $P$  by the  $\ell_1$ -norm without changing the basic properties.

The  $\ell_1$ -norm and the  $\ell_\infty$ -norm are special **polytopal norms** on  $\mathbb{R}^n$  in the sense that their respective “unit spheres”  $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_p \leq 1\}$  where  $p \in \{1, \infty\}$  are full dimensional polytopes in  $\mathbb{R}^n$ . Moreover, the polytopes  $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_1 \leq 1\}$  and  $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_\infty \leq 1\}$  are **dual polytopes** in the sense that there is an one-to-one correspondence between the facets of either of them and the extreme points of the other. More generally, let  $\|\cdot\|_P$  be any polytopal norm on  $\mathbb{R}^n$ , i.e. the unit sphere with respect to  $\|\cdot\|_P$  is a polytope. Given  $\|\cdot\|_P$  define for  $\mathbf{y} \in \mathbb{R}^n$  the “length” of  $\mathbf{y}$  in the **dual norm**  $\|\cdot\|_P^*$  by

$$\|\mathbf{y}\|_P^* = \max\{\mathbf{y}^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\}. \quad (9.57)$$

The maximum exists because  $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_P \leq 1\}$  is a polytope, i.e. a compact convex subset of  $\mathbb{R}^n$ , and  $\mathbf{y}^T \mathbf{x}$  is continuous in  $\mathbf{x}$ . You prove that  $\|\cdot\|_1$  and  $\|\cdot\|_\infty$  is a pair of dual norms. Exercise 9.8 shows that  $\|\cdot\|_P^*$  is a polytopal norm on  $\mathbb{R}^n$  and that it satisfies **Hölder's inequality**

$$\mathbf{y}^T \mathbf{x} \leq \|\mathbf{y}\|_P^* \|\mathbf{x}\|_P \text{ for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \quad (9.58)$$

It also shows that an  $\varepsilon$ -solidification of polyhedra can be defined with respect to any polytopal norm on  $\mathbb{R}^n$ . Moreover, if we have a separation subroutine for any polyhedron  $P$  that finds a most violated separator then we can separate points from it using the corresponding  $\varepsilon$ -solidification of  $P$ .

### 9.6.1 Rational Rounding and Continued Fractions

**Rational rounding** is the process of approximating a real number  $\Theta \in \mathbb{R}$  as a ratio of two integer numbers  $p/q$  with  $q \geq 1$ . If the denominator  $q = 1$  then  $p = \lfloor \Theta \rfloor$  or  $p = \lceil \Theta \rceil$  are the only “reasonable” choices with a maximum error of  $1/2$ . We denote by

$$[\Theta] = \min\{\Theta - \lfloor \Theta \rfloor, \lceil \Theta \rceil - \Theta\} \quad (9.59)$$

the smaller of the two fractional parts of  $\Theta$  and pronounce  $[\Theta]$  as “frac of  $\Theta$ .”

**Remark 9.14** Let  $\Theta$  and  $Q > 1$  be any real numbers. Then there exists an integer  $q$  such that  $1 \leq q \leq Q$  and  $[q\Theta] \leq Q^{-1}$ .

A ratio  $p/D$  with integers  $p$  and  $D \geq 1$  is a **best approximation** to  $\Theta \in \mathbb{R}$  if

$$[D\Theta] = |D\Theta - p| \text{ and } [q\Theta] > [D\Theta] \text{ for all } 1 \leq q < D. \quad (9.60)$$

Two sequences of integers  $q_1 = 1 < q_2 < q_3 < \dots$  and  $p_1, p_2, p_3, \dots$  such that  $p_n/q_n$  is a best approximation to  $\Theta$  for all  $1 \leq q \leq D = q_n$  can be constructed inductively as follows. Initially, we let  $q_1 = 1$  and find an integer  $p_1$  such that  $|q_1\Theta - p_1| = [\Theta] \leq 1/2$ . By definition  $p_1/q_1$  is a best approximation of  $\Theta$  with  $D = 1$ . If  $p_1 = q_1\Theta$ , i.e. if  $\Theta$  is an integer, the inductive process stops. So suppose that we have constructed the  $n \geq 1$  first, best approximations to  $\Theta$  and that the process does not stop. Then we have integers  $p_n$  and  $q_n$  such that  $p_n \neq q_n\Theta$ , i.e.  $[q_n\Theta] > 0$ . From Remark 9.14 when applied with  $Q > [q_n\Theta]^{-1}$  it follows that there exist integer numbers  $q \geq 1$  such

that  $[q\Theta] < [q_n\Theta]$ . Let  $q_{n+1}$  be the *smallest* integer and  $p_{n+1}$  be a corresponding integer number such that

$$[q_{n+1}\Theta] = |q_{n+1}\Theta - p_{n+1}| \quad (9.61)$$

$$[q_{n+1}\Theta] < [q_n\Theta] \quad (9.62)$$

$$[q\Theta] \geq [q_n\Theta] \text{ for all } 1 \leq q < q_{n+1}. \quad (9.63)$$

Since by the inductive hypothesis  $p_n/q_n$  is a best approximation to  $\Theta$  with  $D = q_n$  we have from (9.62) that  $q_{n+1} > q_n$  and from (9.63) and (9.62) that the ratio  $p_{n+1}/q_{n+1}$  is a best approximation to  $\Theta$  with  $D = q_{n+1}$ . Consequently, the induction works, the numbers  $p_n$  and  $q_n$  have the stated properties and either the process continues or it stops.

Several important properties of this sequence of best approximations to  $\Theta$  can be established that lead to a polynomially bounded algorithm for finding best approximations to rational numbers when the denominator is bounded by a prescribed number. Since  $q_n < q_{n+1}$  and applying Remark 9.14 with  $Q = q_{n+1}$  it follows from (9.63) that

$$q_n[q_n\Theta] < q_{n+1}[q_n\Theta] \leq 1. \quad (9.64)$$

Moreover, the signs of  $q_n\Theta - p_n$  and  $q_{n+1}\Theta - p_{n+1}$  alternate, i.e.

$$(q_n\Theta - p_n)(q_{n+1}\Theta - p_{n+1}) \leq 0. \quad (9.65)$$

All numbers in the sequence are integers and

$$p_n q_{n+1} - p_{n+1} q_n = q_n(q_{n+1}\Theta - p_{n+1}) - q_{n+1}(q_n\Theta - p_n), \quad (9.66)$$

$$p_n q_{n+1} - p_{n+1} q_n = \pm 1, \quad (9.67)$$

$$\text{sign}(p_n q_{n+1} - p_{n+1} q_n) = -\text{sign}(q_n\Theta - p_n) \quad (9.68)$$

$$p_n q_{n+1} - p_{n+1} q_n = -(p_{n-1} q_n - p_n q_{n-1}) \quad (9.69)$$

for all  $p_n, q_n$  that the inductive process generates. From (9.69)  $p_n(q_{n+1} - q_{n-1}) = q_n(p_{n+1} - p_{n-1})$ . From (9.67)  $\text{g.c.d.}(p_n, q_n) = 1$  and since the coprime representation of a rational is unique, there exists some positive integer  $a_n$  such that  $q_{n+1} - q_{n-1} = a_n q_n$  and  $p_{n+1} - p_{n-1} = a_n p_n$ . So for all  $n \geq 2$  of the inductive process there exist integers  $a_n \geq 1$  such that

$$q_{n+1} = a_n q_n + q_{n-1}, \quad p_{n+1} = a_n p_n + p_{n-1}. \quad (9.70)$$

Multiply the first part of (9.70) by  $\Theta$ , subtract the second and use the alternating signs (9.65). We get

$$|q_{n-1}\Theta - p_{n-1}| = a_n |q_n\Theta - p_n| + |q_{n+1}\Theta - p_{n+1}|. \quad (9.71)$$

Now  $|q_{n+1}\Theta - p_{n+1}| = [q_{n+1}\Theta] < [q_n\Theta] = |q_n\Theta - p_n|$  by (9.62). Hence

$$a_n = \left\lfloor \frac{|q_{n-1}\Theta - p_{n-1}|}{|q_n\Theta - p_n|} \right\rfloor, \quad (9.72)$$

which together with (9.70) gives a procedure to calculate  $p_{n+1}, q_{n+1}$  once the values of  $p_n$  and  $q_n$  are known for  $k \leq n$  where  $n \geq 2$ .

To start the procedure to calculate  $p_n$  and  $q_n$  iteratively from (9.70) and (9.72) we have to know what the first two iterations of the inductive process produce in terms of  $p_n$  and  $q_n$  or prescribe a start that is consistent with the inductive hypothesis that we have made to derive the above properties.

If  $|\Theta| \geq 1$  and  $\Theta$  is not an integer, then we can always write  $\Theta = \lfloor \Theta \rfloor + \Theta'$  with  $0 < \Theta' < 1$  and a best approximation to  $\Theta'$  yields instantaneously a best approximation to  $\Theta$ . So we can assume WROG that  $0 < \Theta < 1$ . If  $0 < \Theta \leq 1/2$  then  $p_1 = 0$ ,  $q_1 = 1$  yields a best approximation to  $\Theta$  for  $D = 1$  and by Exercise 9.9 (ii)  $p_2 = 1$ ,  $q_2 = \lfloor \Theta^{-1} \rfloor$  does the same for  $D = \lfloor \Theta^{-1} \rfloor$ . So if we initialize

$$p_0 = 1, q_0 = 0, p_1 = 0, q_1 = 1, \quad (9.73)$$

then formulas (9.70) and (9.72) produce precisely the respective best approximations to  $\Theta$  for  $n \leq 2$ , the inductive hypothesis applies and so we can continue to use the formulas until the process stops, i.e.  $\Theta = p_{n+1}/q_{n+1}$ , if it stops at all. If  $1/2 < \Theta < 1$  then the initialization (9.73) produces  $p_1 = 0$ ,  $q_1 = 1$  which is, of course, not a best approximation to  $\Theta > 1/2$ . Calculating we get from (9.70) and (9.72)  $p_2 = 1$ ,  $q_2 = 1$  because  $a_1 = 1$ , which gives a best approximation of  $\Theta > 1/2$  for  $D = 1$ . Carrying out one more step in the iterative application of (9.70) and (9.72) with the initialization (9.73) we get  $a_2 = \lfloor \frac{\Theta}{1-\Theta} \rfloor$ ,  $p_3 = a_2$  and  $q_3 = a_2 + 1$ . By Exercise 9.9 (iii)  $p_3/q_3$  is a best approximation to  $\Theta$  for all  $1 \leq q < D = \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1$ . Now the inductive hypothesis applies to  $p_2$ ,  $q_2$  and  $p_3$ ,  $q_3$ , we ignore the first iteration and thus we can continue to use the formulas like in the first case.

If the number  $\Theta$  equals  $r/s$  with integers  $r, s \geq 1$  and  $\text{g.c.d}(r, s) = 1$ , then  $r/s$  is itself a best approximation to  $\Theta$  for all  $1 \leq q < s$ , the  $q_n$  are strictly increasing and thus  $q_n = s$ ,  $p_n = r$  at some point and the process stops. If  $\Theta$  is irrational then  $\lim_{n \rightarrow \infty} \frac{p_n}{q_n} = \Theta$  because by (9.64) we have  $|\Theta - p_n/q_n| < q_n^{-2}$  and  $1 = q_1 \leq q_2 < q_3 < \dots$  for any  $0 < \Theta < 1$ . With the initialization (9.73) it follows from (9.65) and (9.67) that for all  $n \geq 0$

$$(-1)^{n+1}(q_n\Theta - p_n) \geq 0, \quad p_n q_{n+1} - p_{n+1} q_n = (-1)^n. \quad (9.74)$$

Consider now the **best approximation problem** for  $\Theta \in \mathbb{R}$  relative to a prescribed integer number  $D \geq 2$ : we wish to find integer numbers  $p$  and  $1 \leq q \leq D$  such that  $|\Theta - p/q|$  is as small as possible.

### Best Approximation Algorithm ( $\Theta, D$ )

**Step 0:** Set  $a_0 := \lfloor \Theta \rfloor$ ,  $\Theta := \Theta - a_0$ ,  $p_0 := 1$ ,  $q_0 := 0$ ,  $p_1 := 0$ ,  $q_1 := 1$ ,  $n := 1$ .

**Step 1:** **if**  $q_n\Theta = p_n$  **stop** " $p_n/q_n$  is a best approximation".

**if**  $q_n > D$  **go to** Step 3. Set  $a_n := \left\lfloor \frac{|q_{n-1}\Theta - p_{n-1}|}{|q_n\Theta - p_n|} \right\rfloor$ .

**Step 2:** Set  $p_{n+1} := a_n p_n + p_{n-1}$ ,  $q_{n+1} := a_n q_n + q_{n-1}$ . Replace  $n + 1$  by  $n$ , **go to** Step 1.

**Step 3:** Set  $k := \lfloor \frac{D-q_{n-1}}{q_n} \rfloor$ ,  $p'_n := p_{n-1} + kp_n$ ,  $q'_n := q_{n-1} + kq_n$ .

**if**  $|\Theta - p_n/q_n| \leq |\Theta - p'_n/q'_n|$  **stop** " $p_n/q_n$  is a best approximation".

**stop** " $p'_n/q'_n$  is a best approximation".

**Remark 9.15** (Correctness and finiteness) For rational  $\Theta$  and integer  $D \geq 2$  the best approximation algorithm's run time is polynomial in the digital size of its input.

$$\Theta = \Theta_1 = \frac{1}{a_1 + \Theta_2} = \frac{1}{a_1 + \frac{1}{a_2 + \Theta_3}} = \cdots = \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots \frac{1}{a_{N-1} + \frac{1}{a_N}}}}}$$

**Fig. 9.10.** Continued fractions for a rational number

To relate the preceding to the **continued fraction** process define

$$\Theta_n = \frac{|q_n \Theta - p_n|}{|q_{n-1} \Theta - p_{n-1}|} \text{ for } n \geq 1,$$

where we assume again like in the algorithm that the integer part of the original data has been cleared away, i.e.  $0 < \Theta < 1$ . It follows from the initialization (9.73) that  $\Theta_1 = \Theta$  and  $0 \leq \Theta_n < 1$  for  $n \geq 1$  from (9.62). From (9.71) we get

$$\Theta_n^{-1} = a_n + \Theta_{n+1} \text{ for all } n \geq 1 \text{ with } \Theta_n > 0. \quad (9.75)$$

Now suppose that  $\Theta$  is a rational number. Then by the above  $\Theta = p_{N+1}/q_{N+1}$ , say, so that  $\Theta_{N+1} = 0$  and thus by (9.75)  $\Theta_N^{-1} = a_N$ . Consequently we can write  $\Theta$  like in Figure 9.10, which explains the term “continued fraction.” If  $\Theta$  is irrational then  $\Theta_n > 0$  for all  $n \geq 1$  and the continued fraction goes on “forever”, which permits one to find high-precision rational approximations of irrational numbers.

Rational rounding can e.g. be used in the context of establishing the polynomiality of linear programming via the combination of the binary search algorithm 2 and the basic ellipsoid algorithm, see Chapter 9.3.1, to find the optimal objective function value  $z_P$  exactly; see the text.

## 9.7 Optimization and Separation

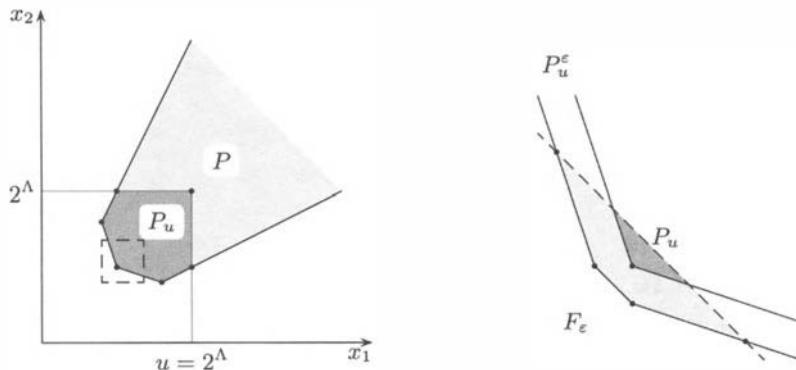
Here the fundamental *polynomial-time equivalence* of optimization and separation for rational polyhedra is established. It is shown that for any rational polyhedron  $P \subseteq \mathbb{R}^n$  the linear optimization problem

$$\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P\}$$

can be solved in time that is polynomial in  $n$ , the facet complexity  $\phi$  of  $P$  and  $\langle \mathbf{c} \rangle$  if and only if the separation problem (9.53) is solvable in time that is polynomial in  $n$ ,  $\phi$  and  $\langle \mathbf{x}^k \rangle$ .

The geometric idea of our construction is simple and illustrated in Figures 9.11 and 9.12 for a full dimensional polyhedron in  $\mathbb{R}^2$ . In Figure 9.11 we depict the situation when  $\mathbf{c}\mathbf{x}$  with  $\mathbf{c} = (-1, -1)$  is maximized, while Figure 9.12 illustrates the basic idea for finding direction vectors in the asymptotic cone of a polyhedron. For any integer  $u \geq 1$  denote

$$P_u = P \cap \{\mathbf{x} \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n\}. \quad (9.76)$$



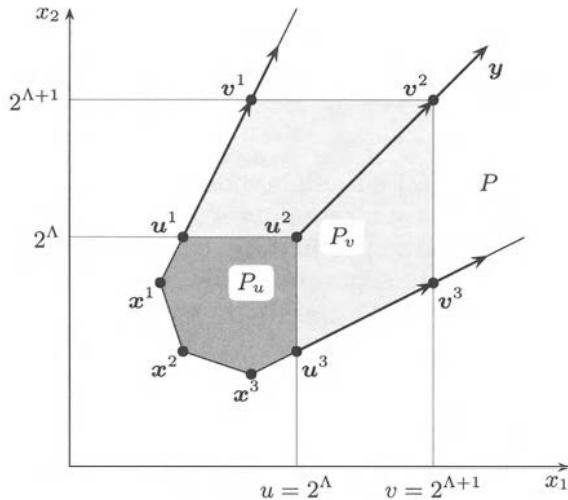
**Fig. 9.11.** Locating the optimum and proving optimality

Its  $\epsilon$ -solidification  $P_u^\epsilon$  with respect to the  $\ell_1$ -norm is either empty or a full dimensional polytope in  $\mathbb{R}^n$ . To find direction vectors in the asymptotic cone we define

$$\Lambda = \phi + 5n\phi + 4n^2\phi + 1. \quad (9.77)$$

Setting  $u = 2^\Lambda > 2^{4n\phi}$  it follows from point 7.5(b) that all extreme points of  $P$  are properly contained in  $P_u$ . Moreover, we get a sufficiently large portion of the “unbounded” region of  $P$  by doubling this value of  $u$  which permits us to find direction vectors in the asymptotic cone of  $P$  by solving two linear optimization problems rather than one.

At the outset the polyhedron  $P \subseteq \mathbb{R}^n$  may be empty, it may have none or several optimal extreme points or the objective function may be unbounded. To encompass all possibilities, we perturb the original objective function so as to achieve uniqueness of the maximizer over the larger polytope  $P_u$  where  $u = 2^{\Lambda+1}$ . To locate the unique maximizer  $x^{max}$  of the perturbed objective function over  $P_u$  where  $u \in \{2^\Lambda, 2^{\Lambda+1}\}$  we let the algorithm run until  $x^{max}$  is the only rational point in the remaining  $\epsilon$ -optimal set with components that have denominators  $q_j \geq 1$  and  $q_j \leq 2^{6n\phi}$ . We have illustrated the basic idea in the second part of Figure 9.11 which “zooms” into the area of the first part depicted by a square and that contains  $x^{max}$ . Running the algorithm “long enough” we find an  $\epsilon$ -optimal rational vector  $\bar{x} \in P_u^\epsilon$  “close” to  $x^{max}$ . Using the best approximation algorithm of Chapter 9.6 we can round  $\bar{x}$  componentwise to obtain  $x^{max}$ , see Remark 9.16. If the maximizer  $x^{max}$  obtained this way satisfies  $|x_j^{max}| < 2^\Lambda$  for all  $1 \leq j \leq n$ , then we are done and an optimal extreme point of  $P$  for the original objective function has been located. If the maximizer  $x^{max}$  satisfies  $|x_j^{max}| = 2^\Lambda$  for some  $1 \leq j \leq n$ , then either the linear optimization problem over  $P$  has an unbounded objective function value or an optimal extreme point of  $P$  may still exist. In either case, we execute the algorithm a second time to optimize the same perturbed objective function over  $P_u$  where  $u = 2^{\Lambda+1}$ . We get a second, unique maximizer  $y^{max}$ , say, over the larger polytope and – since we have used identical objective functions – the difference vector  $y = y^{max} - x^{max}$  belongs to the asymptotic cone of the polyhedron  $P$ . This will let us decide whether the objective function is unbounded or bounded and in the latter case we find an optimizing point as well. Exercise 9.10 reviews the perturbation technique of Chapter 7.5.4 and summarizes part of what we need to establish the validity of our construction. The last part of Exercise 9.10 shows in particular that we can assume WLOG that every nontrivial linear inequality  $hx \leq h_0$  belonging to a linear description of a rational polyhedron  $P$  of facet complexity  $\phi$  satisfies  $\|h\|_\infty = 1$ .



**Fig. 9.12.** Finding a direction vector in the asymptotic cone of  $P$

### 9.7.1 $\varepsilon$ -Optimal Sets and $\varepsilon$ -Optimal Solutions

The following remark makes the first part of the construction precise and gives an analytical meaning to the terms “ $\varepsilon$ -optimal set” and “ $\varepsilon$ -optimal solution”: the set  $F_\varepsilon$  defined in (9.78) is an  $\varepsilon$ -optimal set and any rational  $\bar{x} \in F_\varepsilon$  is an  $\varepsilon$ -optimal solution to the linear optimization problem over  $P_u$ .

**Remark 9.16** Let  $P \subseteq \mathbb{R}^n$  be a rational polytope of facet complexity  $\phi$ ,  $P_u$  be as defined in (9.76) with integer  $u \geq 2^{4n\phi}$  and let  $P_u^\varepsilon$  be the  $\varepsilon$ -solidification of  $P_u$  with respect to the  $\ell_1$ -norm. Let  $x^{max} \in P_u$  be the unique maximizer of  $d\mathbf{x}$  over  $P_u$  and  $z_P = d\mathbf{x}^{max}$ , where  $d$  has rational components and  $\|d\|_\infty = 1$ . Define

$$F_\varepsilon = P_u^\varepsilon \cap \{\mathbf{x} \in \mathbb{R}^n : d\mathbf{x} \geq z_P - \varepsilon\}, \quad (9.78)$$

where  $0 < \varepsilon \leq 2^{-\Psi}$  and  $\Psi = 9n\phi + 12n^2\phi + \langle d \rangle$ . Then  $F_\varepsilon$  is a full dimensional polytope,  $\text{vol}(F_\varepsilon) \geq 2^n\varepsilon^n/n!$  and every extreme point  $y \in F_\varepsilon$  satisfies  $|y_j - x_j^{max}| < 2^{-6n\phi-1}$  for  $1 \leq j \leq n$ . Moreover, rounding any rational  $\bar{x} \in F_\varepsilon$  componentwise by the best approximation algorithm with  $\Theta = \bar{x}_j$  and  $D = 2^{6n\phi}$  we obtain  $x_j^{max}$  and thus the maximizer  $x^{max}$  in time polynomial in  $n, \phi$  and  $\langle \bar{x} \rangle$ .

### 9.7.2 Finding Direction Vectors in the Asymptotic Cone

The following remark makes the second part of the construction precise; see also Exercise 9.11 (i) below.

**Remark 9.17** Let  $P \subseteq \mathbb{R}^n$  be a rational polyhedron of facet complexity  $\phi$ , let  $v > u \geq 2^\Lambda$  be any integers where  $\Lambda$  is defined in (9.77), let  $P_v$  and  $P_u$  be defined as in (9.76) with respect to  $v$  and  $u$ , respectively, and let  $C_\infty$  be the asymptotic cone of  $P$ . Then every extreme point  $x^v \in P_v$  can be written as  $x^v = x + vt$  where  $x, t \in \mathbb{R}^n$  are rational vectors,  $t \in C_\infty$ ,  $\langle t \rangle \leq 4n^2\phi$  and moreover,

$x^u = x + ut \in P_u$  is an extreme point of  $P_u$ . Likewise, if  $x^u \in P_u$  is an extreme point of  $P_u$  and  $x^u = x + ut$ , say, then  $x^v = x^u + (v - u)t \in P_v$  is an extreme point of  $P_v$ .

### 9.7.3 A CCS Ellipsoid Algorithm

The CCS ellipsoid algorithm is a central cut, sliding objective version of the basic ellipsoid algorithm. It takes the number of variables  $n$ , a rational vector  $d$  with  $\|d\|_\infty = 1$ , the facet complexity  $\phi$  and  $P$  as input. “ $P$ ” is an identifier of the polyhedron over which we optimize the linear function  $dx$  and used to communicate with the separation subroutine  $\text{SEPAR}(x, h, h_0, \phi, P)$ .  $u$  specifies the hypercube with which we intersect  $P$ ; see (9.76).  $\varepsilon$ ,  $p$  and  $T$  are the parameters for the  $\varepsilon$ -solidification of  $P$  in the  $\ell_1$ -norm, the required precision for the approximate calculation in terms of binary positions and the number of steps of the algorithm, respectively.

The subroutine  $\text{SEPAR}(x^k, h, h_0, \phi, P)$  of Step 1 returns a most violated separator, i.e., a solution  $(h, h_0)$  to (9.53). The normalization requirement  $\|h\|_\infty = 1$  is no serious restriction at all; see Exercise 9.10 (v). It shows why we use the  $\varepsilon$ -solidification of  $P_u$  or  $P$  in the  $\ell_1$ -norm: by Exercise 9.7(vi) we get  $h^T x \leq h_0 + \varepsilon \|h\|_\infty = h_0 + \varepsilon$  as the corresponding inequality for  $P_u^\varepsilon$  or  $P^\varepsilon$  if  $h^T x \leq h_0$  for all  $x \in P$  and  $(h, h_0)$  was returned by the separation subroutine. So, we “perturb” the feasible set like in Chapter 9.3 by adding a “small enough”  $\varepsilon > 0$  to the right-hand side  $h_0$  of every most violated separator  $(h, h_0)$  that the separation subroutine returns.  $h$  is a column vector and  $u^j \in \mathbb{R}^n$  is the  $j$ -th unit vector.

#### CCS Ellipsoid Algorithm ( $n, d, \phi, P, u, \varepsilon, p, T$ )

**Step 0:** Set  $k := 0$ ,  $x^0 := 0$ ,  $F_0 := nuI_n$ ,  $z_L := -nu\|d\| - 1$ ,  $\bar{z} := z_L$ .

**Step 1:** **if**  $|x_j^k| > u + \varepsilon$  for some  $j \in \{1, \dots, n\}$  **then**  
     set  $h := u^j$  if  $x_j^k > 0$ ,  $h := -u^j$  otherwise.  
     **else**  
         **call**  $\text{SEPAR}(x^k, h, h_0, \phi, P)$ .  
         **if**  $h^T x^k \leq h_0 + \varepsilon$  **then**  
             Set  $h := -d^T$ . **if**  $dx^k > \bar{z}$  **then** set  $\bar{x} := x^k$ ,  $\bar{z} := dx^k$ .  
             **endif**  
         **endif**  
     **endif**

**Step 2:** **if**  $k = T$  **go to** Step 3. Set

$$\begin{aligned} x^{k+1} &\approx x^k - \frac{1}{n+1} \frac{\mathbf{F}_k \mathbf{F}_k^T h}{\|\mathbf{F}_k^T h\|}, \\ \mathbf{F}_{k+1} &\approx \frac{n+1/12n}{\sqrt{n^2-1}} \mathbf{F}_k \left( I_n - \frac{1 - \sqrt{(n-1)/(n+1)}}{h^T \mathbf{F}_k \mathbf{F}_k^T h} (\mathbf{F}_k^T h)(h^T \mathbf{F}_k) \right), \end{aligned}$$

where  $\approx$  means that componentwise the round-off error is at most  $2^{-p}$ .

Replace  $k + 1$  by  $k$  and **go to** Step 1.

**Step 3:** **if**  $\bar{z} = z_L$  **stop** “ $z_P = -\infty$ .  $P$  is empty.”

Round  $\bar{x}$  componentwise to the nearest rational  $x^*$  such that each component of  $x^*$  has a positive denominator less than or equal to  $2^{6n\phi}$ . **stop** “ $x^*$  is an optimal extreme point of  $P_u$ .”

**Remark 9.18** (Correctness and finiteness) Let  $P \subseteq \mathbb{R}^n$  be a rational polyhedron of facet complexity  $\phi$ . Let  $d \in \mathbb{R}^n$  be a rational vector with  $\|d\|_\infty = 1$  such that  $\max\{dx : x \in P_u\}$  has a unique maximizer  $x^{max} \in P_u$  with objective function value  $z_P = dx^{max}$  if  $P \neq \emptyset$ , where  $P_u$  is defined in (9.76) and  $u \geq 2^{4n\phi}$  is any integer. If the subroutine  $\text{SEPAR}(x^k, h, h_0, \phi, P)$  returns a most violated separator  $hx \leq h_0$  for  $x^k$  and  $P$  satisfying  $\|h\|_\infty = 1$ , then the CCS ellipsoid algorithm concludes correctly that  $P_u = P = \emptyset$  or it finds  $x^{max}$  if it is executed with the parameters

$$\varepsilon = 2^{-\Psi}, \quad T = \lceil 6n^2 \log \frac{n^2 u}{\varepsilon} \rceil, \quad p = 14 + n^2 + \lceil 15n \log \frac{n^2 u}{\varepsilon} \rceil,$$

where  $\Psi = 9n\phi + 12n^2\phi + \langle d \rangle$ .

#### 9.7.4 Linear Optimization and Polyhedral Separation

For any polyhedron  $P \subseteq \mathbb{R}^n$  with a linear description  $Hx \leq h$ , say, we denote by

$$P_\infty = \{\mathbf{y} \in \mathbb{R}^n : H\mathbf{y} \leq \mathbf{0}, \|\mathbf{y}\|_\infty \leq 1\}$$

the intersection of the asymptotic cone  $C_\infty$  of  $P$  with the unit sphere in the  $\ell_\infty$ -norm.

**Linear optimization problem:** Given a rational polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$  and a rational vector  $c \in \mathbb{R}^n$  (i) conclude that  $P$  is empty or (ii) find  $x^{max} \in P$  with  $cx^{max} \geq cx$  for all  $x \in P$  or (iii) find  $t \in P_\infty$  with  $ct > 0$  and  $ct \geq cy$  for all  $y \in P_\infty$ .

**Polyhedral separation problem:** Given a rational polyhedron  $P \subseteq \mathbb{R}^n$  of facet complexity  $\phi$  and a rational vector  $z \in \mathbb{R}^n$  (i) conclude that  $z \in P$  or (ii) find a most violated separator for  $z$  and  $P$ , i.e. find a rational vector  $(h, h_0) \in \mathbb{R}^{n+1}$  that solves the problem

$$\max\{hz - h_0 : (h, h_0) \in SP, \|h\|_\infty = 1\},$$

where  $SP = \{(h, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}\}$ .

Neither problem specifies the way in which the polyhedron  $P \subseteq \mathbb{R}^n$  is given: all we need is the information that the polyhedron is a subset of  $\mathbb{R}^n$ , that its facet complexity is at most  $\phi$  and some “identifier” for  $P$  that permits us to communicate to some subroutine for instance.

**Remark 9.19** Let  $P \subseteq \mathbb{R}^n$  be any rational polyhedron of facet complexity  $\phi$ . If there exists an algorithm  $A$ , say, that solves the polyhedral separation problem in time that is bounded by a polynomial in  $n, \phi$  and  $\langle z \rangle$ , then the linear optimization problem can be solved in time that is bounded by a polynomial in  $n, \phi$  and  $\langle c \rangle$ .

To outline the proof that the statement of Remark 9.19 can be reversed as well let

$$S = \{\mathbf{x}^1, \dots, \mathbf{x}^p\} \text{ and } T = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$$

be any minimal generator of  $P$  and denote by  $X$  the  $n \times p$  matrix with columns  $\mathbf{x}^i$ , by  $Y$  the  $n \times r$  matrix with columns  $\mathbf{y}^i$ . Either  $X$  or  $Y$  or both may be void. The set  $SP = \{(h, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{x \in \mathbb{R}^n : hx \leq h_0\}\}$  of separators for  $P$  satisfies

$$SP = \{(h, h_0) \in \mathbb{R}^{n+1} : hX - h_0\mathbf{g} \leq \mathbf{0}, hY \leq \mathbf{0}\}, \tag{9.80}$$

where  $\mathbf{g} \in \mathbb{R}^p$  is a row vector with  $p$  components equal to 1. It follows that  $SP$  is a polyhedral cone in  $\mathbb{R}^{n+1}$  of facet complexity at most  $\phi^* = 4n^2\phi + 3$ . Denote by

$$SP_\infty = \{(h, h_0) \in \mathbb{R}^{n+1} : hX - h_0\mathbf{g} \leq \mathbf{0}, hY \leq \mathbf{0}, -e \leq h \leq e\} \tag{9.81}$$

the polyhedron in  $\mathbb{R}^{n+1}$  over which we need to maximize  $\mathbf{h}\mathbf{z} - h_0$  in order to find a most violated separator for  $\mathbf{z}$  and  $P$ , where  $e \in \mathbb{R}^n$  is a row vector with  $n$  components equal to 1. The polyhedron  $SP_\infty$  contains the halfline defined by  $(0, 1) \in \mathbb{R}^n$  if  $X$  is nonvoid, it contains the line defined by  $(0, \pm 1) \in \mathbb{R}^{n+1}$  if and only if  $X$  is void and every nonzero extreme point  $(\mathbf{h}, h_0)$  of  $SP_\infty$  satisfies  $\|\mathbf{h}\|_\infty = 1$ . By Remark 9.19 we can optimize the linear function  $\mathbf{h}\mathbf{z} - h_0$  over  $SP_\infty$  in polynomial time provided that the polyhedral separation problem for  $SP_\infty$  and any rational  $(f, f_0) \in \mathbb{R}^{n+1}$ , say, can be solved in time that is bounded by a polynomial in  $n$ ,  $\phi$  and  $\langle f \rangle + \langle f_0 \rangle$ . So we need to identify the set of separators for  $SP_\infty$ . We will do so in two steps: first we identify the set  $SP^*$ , say, of separators for  $SP$ .

Since  $SP$  is a polyhedral cone in  $\mathbb{R}^{n+1}$ , its set  $SP^*$  of separators is a subset of all halfspaces of  $\mathbb{R}^{n+1}$  that contain the origin, i.e.

$$SP^* = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : SP \subseteq \{(\mathbf{h}, h_0) \in \mathbb{R}^{n+1} : \mathbf{h}\mathbf{x} - h_0 x_{n+1} \leq 0\}\},$$

because  $SP$  contains the origin of  $\mathbb{R}^{n+1}$  and with every nonzero point the cone  $SP$  contains the entire halfline defined by it. It follows that

$$\begin{aligned} SP^* &= \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \mathbf{h}\mathbf{x} - h_0 x_{n+1} \leq 0 \text{ for all } (\mathbf{h}, h_0) \in SP\} \\ &= \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : H\mathbf{x} - h_0 x_{n+1} \leq 0, -x_{n+1} \leq 0\}, \end{aligned}$$

where  $H\mathbf{x} \leq \mathbf{h}$  is any linear description of the polyhedron  $P \subseteq \mathbb{R}^n$ . The inequality  $x_{n+1} \geq 0$  follows because  $SP$  contains the halfline noted above. If  $P \neq \emptyset$  and  $X$  is void, i.e. if  $P$  contains lines, then we must replace  $x_{n+1} \geq 0$  by the equation  $x_{n+1} = 0$ , because  $SP$  contains the line defined by  $(0, \pm 1) \in \mathbb{R}^{n+1}$  in this case. So if  $P \neq \emptyset$  and  $X$  is nonvoid, then the set  $SP^*$  of separators for  $SP$  is the *homogenization* of the polyhedron  $P$  – see (7.5). If  $P \neq \emptyset$  and  $X$  is void, then the set  $SP^*$  of separators in question is simply the asymptotic cone of  $P$  – see (7.3). Define

$$SP_\infty^* = \{(\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} : H\mathbf{x} - h_0 x_{n+1} \leq 0, -e \leq \mathbf{x} \leq e, 0 \leq x_{n+1} \leq 1\}, \quad (9.82)$$

where  $e \in \mathbb{R}^n$  has  $n$  components equal to 1. So the set  $SP_\infty^*$  of all separators is a nonempty polytope in  $\mathbb{R}^{n+1}$ , every nonzero extreme point of which has an  $\ell_\infty$ -norm of 1.

Consider now  $(\mathbf{x}^0, x_{n+1}^0) \neq (\mathbf{x}^1, x_{n+1}^1) \in SP_\infty^*$  with  $x_{n+1}^0 > 0$  and  $x_{n+1}^1 > 0$ . Then  $(x_{n+1}^1 \mathbf{x}^0, x_{n+1}^0 x_{n+1}^1) \in SP_\infty^*$  and  $(x_{n+1}^0 \mathbf{x}^1, x_{n+1}^0 x_{n+1}^1) \in SP_\infty^*$ . If for some  $(f, f_0) \in \mathbb{R}^{n+1}$  we have

$$f(x_{n+1}^1 \mathbf{x}^0) - f_0 x_{n+1}^0 x_{n+1}^1 > f(x_{n+1}^0 \mathbf{x}^1) - f_0 x_{n+1}^0 x_{n+1}^1 > 0,$$

then the point  $(\mathbf{x}^0, x_{n+1}^0)$  is a “more violated” separator for  $(f, f_0)$  and  $SP$  than  $(\mathbf{x}^1, x_{n+1}^1)$ , because for all  $\lambda \geq 0$   $(\lambda x_{n+1}^1 \mathbf{x}^0, \lambda x_{n+1}^0 x_{n+1}^1) \in SP^*$ ,  $(\lambda x_{n+1}^0 \mathbf{x}^1, \lambda x_{n+1}^0 x_{n+1}^1) \in SP^*$  and the previous inequalities remain true for the entire open halfline, i.e. for all  $\lambda > 0$ .

Thus even if  $f\mathbf{x}^1 - f_0 x_{n+1}^1 > f\mathbf{x}^0 - f_0 x_{n+1}^0 > 0$ , rather than  $f\mathbf{x}^0 - f_0 x_{n+1}^0 > f\mathbf{x}^1 - f_0 x_{n+1}^1$ , for the “original” points  $(\mathbf{x}^0, x_{n+1}^0)$ ,  $(\mathbf{x}^1, x_{n+1}^1) \in SP_\infty^*$  it can happen that  $(\mathbf{x}^0, x_{n+1}^0)$  defines a more violated separator. It follows that we have to scale the “homogenizing” components of two violated separators  $(\mathbf{x}^0, x_{n+1}^0)$  and  $(\mathbf{x}^1, x_{n+1}^1)$  with  $x_{n+1}^0 > 0$  and  $x_{n+1}^1 > 0$  to have *equal* values if we want to decide which one of the two is more violated than the other; see also Exercise 9.12 (iii) below.

Suppose that  $(f, f_0) \in \mathbb{R}^{n+1}$  is given, that  $P \neq \emptyset$  and that algorithm  $B$ , say, solves the linear optimization problem over  $P \subseteq \mathbb{R}^n$ . We run the algorithm with the objective function vector  $\mathbf{c} = f$ .

Assume that algorithm  $B$  finds  $\mathbf{x}^{max} \in P$  with  $f\mathbf{x}^{max} \geq f\mathbf{x}$  for all  $\mathbf{x} \in P$ . If  $f\mathbf{x}^{max} \leq f_0$  then we conclude that  $(f, f_0) \in SP$ . Otherwise,  $f\mathbf{x}^{max} - f_0 > 0$  and  $f\mathbf{x}^{max} - f_0 \geq f\mathbf{x} - f_0$  for all  $\mathbf{x} \in P$ .

Then

$$\mathbf{x}^0 = \alpha \mathbf{x}^{max}, \quad x_{n+1}^0 = \alpha \quad \text{where } \alpha^{-1} = \left\| \begin{pmatrix} \mathbf{x}^{max} \\ 1 \end{pmatrix} \right\|_\infty \quad (9.83)$$

solves the polyhedral separation problem for  $(\mathbf{f}, f_0)$  and  $SP$ ; see the text.

Suppose that algorithm  $B$  finds  $\mathbf{t} \in P_\infty$  with  $\mathbf{f}\mathbf{t} > 0$  and  $\mathbf{f}\mathbf{t} \geq \mathbf{f}\mathbf{y}$  for all  $\mathbf{y} \in P_\infty$ . Then

$$\mathbf{x}^0 = \mathbf{t}, \quad x_{n+1}^0 = 0 \quad (9.84)$$

solves the polyhedral separation problem for  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  and  $SP$ ; see the text.

It follows that if  $P \neq \emptyset$  we can solve the polyhedral separation problem for any rational  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  and the cone  $SP$  in time that is polynomially bounded in  $n$ ,  $\phi$  and  $\langle \mathbf{f} \rangle + \langle f_0 \rangle$  by solving  $\max\{\mathbf{f}\mathbf{x} : \mathbf{x} \in P\}$ . If we conclude that  $(\mathbf{f}, f_0) \in SP$  define  $(\mathbf{x}^0, x_{n+1}^0) = (\mathbf{0}, 0)$ ; otherwise the most violated separator  $(\mathbf{x}^0, x_{n+1}^0)$  for  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  and  $SP$  is given by (9.83) or (9.84), respectively.

It remains to show that we can solve the separation problem for  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  and  $SP_\infty$ .

The polyhedron  $SP_\infty$  differs from the cone  $SP$  by exactly  $2n$  constraints of the form  $-1 \leq h_j \leq 1$  for all  $1 \leq j \leq n$ , which we can check by LIST-and-CHECK in polynomial time. Given  $(\mathbf{x}^0, x_{n+1}^0)$  define

$$\alpha = \mathbf{f}\mathbf{x}^0 - f_0 x_{n+1}^0, \quad \beta = \max_{1 \leq j \leq n} \{0, 1 - f_j\}, \quad \gamma = \max_{1 \leq j \leq n} \{0, 1 + f_j\},$$

where  $f_j$  is the  $j$ -th component of  $\mathbf{f} \in \mathbb{R}^n$ . If  $\max\{\alpha, \beta, \gamma\} = 0$  we conclude that  $(\mathbf{f}, f_0) \in SP_\infty$ . Otherwise, if  $\max\{\alpha, \beta, \gamma\} = \alpha$  then  $(\mathbf{x}^0, x_{n+1}^0)$  is a most violated separator for  $(\mathbf{f}, f_0)$  and  $SP_\infty$ . If  $\max\{\alpha, \beta, \gamma\} = \beta$  then  $(\mathbf{x}^0, x_{n+1}^0) = (\mathbf{u}^k, 1) \in \mathbb{R}^{n+1}$  is a most violated separator for  $(\mathbf{f}, f_0)$  and  $SP_\infty$  where  $k \in \{1, \dots, n\}$  is such that  $\beta = 1 - f_k$ . Likewise, if  $\max\{\alpha, \beta, \gamma\} = \gamma$  then  $(\mathbf{x}^0, x_{n+1}^0) = (-\mathbf{u}^k, 1) \in \mathbb{R}^{n+1}$  is a most violated separator for  $(\mathbf{f}, f_0)$  and  $SP_\infty$  where  $k \in \{1, \dots, n\}$  is such that  $\gamma = 1 + f_k$ .

The preceding combination of some algorithm  $B$  for the linear optimization problem and of LIST-and-CHECK yields a separation routine  $\text{SEPAR}^*(\mathbf{f}, f_0, \mathbf{x}^0, x_{n+1}^0, \phi, SP_\infty)$  that solves the polyhedral separation problem for rational  $(\mathbf{f}, f_0) \in \mathbb{R}^{n+1}$  and  $SP_\infty$  if the underlying polyhedron  $P \subseteq \mathbb{R}^n$  is nonempty. Moreover, if the running time for algorithm  $B$  is bounded by a polynomial in  $n$ ,  $\phi$  and  $\langle \mathbf{c} \rangle$ , then the running time of  $\text{SEPAR}^*$  is evidently bounded by a polynomial in  $n$ ,  $\phi$  and  $\langle \mathbf{f} \rangle + \langle f_0 \rangle$ .

To complete the outline of the proof that we can solve the problem

$$\max\{\mathbf{h}\mathbf{z} - h_0 : (\mathbf{h}, h_0) \in SP_\infty\}$$

for any rational  $\mathbf{z} \in \mathbb{R}^n$  in polynomial time if a polynomial-time algorithm  $B$  for the linear optimization problem is known, we proceed as follows.

We run the algorithm  $B$  a first time with the objective function  $\mathbf{c} = \mathbf{0}$ . If algorithm  $B$  concludes that  $P = \emptyset$  then we declare  $\mathbf{h}\mathbf{z} = z_1 > h_0 = z_1 - 1$  to be a solution to the polyhedral separation problem. Since  $P$  is empty, any inequality that is violated by  $\mathbf{z}$  is evidently a most violated inequality for  $\mathbf{z}$  and  $P$ .

So suppose that we conclude that  $P \neq \emptyset$ . Now the separation subroutine  $\text{SEPAR}^*$  applies and thus by Remark 9.19 we can solve the linear optimization problem

$$\max\{\mathbf{h}\mathbf{z} - h_0 : (\mathbf{h}, h_0) \in SP_\infty\}$$

in time that is polynomially bounded in  $n$ ,  $\phi$  and  $\langle \mathbf{z} \rangle$ .

If  $(\mathbf{h}^{max}, h_0^{max})$  with  $\mathbf{h}^{max}\mathbf{x} - h_0^{max} \geq \mathbf{h}\mathbf{z} - h_0$  for all  $(\mathbf{h}, h_0) \in SP_\infty$  is obtained, then we conclude that  $\mathbf{z} \in P$  if  $\mathbf{h}^{max}\mathbf{z} \leq h_0^{max}$  and otherwise, a most violated separator for  $\mathbf{z}$  and  $P$  has been obtained.

If a finite maximizer  $(\mathbf{h}^{max}, h_0^{max}) \in SP_\infty$  does not exist, then the solution to the above linear optimization problem provides a direction vector in the asymptotic cone of  $SP_\infty$ . The polyhedron  $SP_\infty$  contains the halfline defined by  $(0, 1) \in \mathbb{R}^{n+1}$  along which the objective function tends to  $-\infty$ . Consequently, if the unbounded case arises, then  $SP_\infty$  contains the line defined by  $(0, \pm 1) \in \mathbb{R}^{n+1}$  and the finite generator of  $P$  consists only of halflines, i.e. the matrix  $X$  in the definition of  $SP_\infty$  is void.

So we solve the linear optimization problem

$$\max\{\mathbf{h}\mathbf{z} - h_0 : (\mathbf{h}, h_0) \in SP_\infty, h_0 = 0\}$$

using the separation subroutine SEPAR\*, i.e. we iterate the whole procedure a second time. Now the unbounded case cannot arise and we find a new  $(\mathbf{h}^{max}, 0) \in \mathbb{R}^{n+1}$  with  $\|\mathbf{h}^{max}\|_\infty = 1$  such that  $\mathbf{h}^{max}\mathbf{z} \geq \mathbf{h}\mathbf{z}$  for all  $(\mathbf{h}, 0) \in SP_\infty$ . If  $\mathbf{h}^{max}\mathbf{z} \leq 0$  then we conclude that  $\mathbf{z} \in P$ , whereas otherwise  $(\mathbf{h}^{max}, 0)$  is a most violated separator for  $\mathbf{z}$  and  $P$  since every separator  $(\mathbf{h}, h_0)$  for  $P$  satisfies  $h_0 = 0$  in this case.

The concatenation of polynomials in some variables yields a polynomial in the same variables. Thus the entire procedure can be executed in time that is bounded by some polynomial in  $n$ ,  $\phi$  and  $\langle z \rangle$ .

**Remark 9.20** For any rational polyhedron  $P \subseteq \mathbb{R}^n$  and  $n \geq 2$  the linear optimization problem and the polyhedral separation problem are polynomial-time equivalent problems.

So if either one of the two problems above is solvable in polynomial time, then so is the other. From a theoretical point of view we may thus concentrate on anyone of the two problems to study the algorithmic “tractability” of linear optimization problems over rational polyhedra  $P$  in  $\mathbb{R}^n$ .

Remark 9.19 has several important implications which we do not prove in detail. Among these are that if either problem is polynomially solvable for some rational polyhedron  $P \subseteq \mathbb{R}^n$ , then we can find

- the dimension  $\dim P$  of  $P$ ,
- a linear description of the affine hull  $\text{aff}(P)$  of  $P$ ,
- a linear description of the lineality space  $L_P$  of  $P$ ,
- extreme points and extreme rays of  $P$  if there are any,
- facet-defining linear inequalities for  $P$ , etc

in polynomial time. The latter is of particular importance for the **branch-and-cut** approach to combinatorial optimization problems which rely on finding (parts of) ideal descriptions of the corresponding polyhedra.

## 9.8 Exercises

---

### Exercise 9.1

Show that for  $n = 1$  the ellipsoids  $E_k$  are intervals and that the updating formulas (9.1) and (9.2) become  $x^{k+1} = x^k - \frac{1}{2}F_k \text{sign}(a)$ ,  $F_{k+1} = \frac{1}{2}F_k$  for  $k \geq 0$  where  $ax \leq b$  is any inequality that is violated by  $x^k$  and  $\text{sign}(a) = 1$  if  $a \geq 0$ ,  $-1$  otherwise. (Hint: Note that  $dd^T = 1$ ,  $I_n = 1$  and thus the terms  $n - 1$  cancel.)

---

The inequality  $\|\mathbf{F}^{-1}(\mathbf{x} - \mathbf{x}^0)\| \leq 1$  in one dimension becomes  $|x - x^0| \leq |F|$ , since  $\mathbf{x}$ ,  $\mathbf{x}^0$  and  $\mathbf{F}$  are scalars. WROG we assume that  $F$  is positive. Then the inequality corresponds to the interval  $[x^0 - F, x^0 + F]$ . Let  $ax \leq b$  be a violated inequality by  $x^0$ . We can assume that  $a = \pm 1$  (the case  $a = 0$  is meaningless), since every inequality can be brought in that form. Then the vector  $d$  becomes a scalar  $Fa/|Fa|$ , i.e.  $d = \text{sign}(a) = \pm 1$ . Thus the updating formula for the center is given by

$$x^{k+1} = x^k - \frac{1}{2}F_k \text{sign}(a).$$

Note that  $dd^T$  becomes  $d^2 = 1$  and  $I_n$  becomes  $I_1 = 1$ . Then we calculate from the updating formula for  $F$

$$F_{k+1} = \sqrt{\frac{n^2}{n^2 - 1}} F_k \left( 1 - 1 + \sqrt{\frac{n-1}{n+1}} \right) = \sqrt{\frac{n^2(n-1)}{(n-1)(n+1)^2}} F_k = \frac{n}{n+1} F_k = \frac{1}{2} F_k.$$


---

### Exercise 9.2

Let  $\mathbf{F}$ ,  $\mathbf{R}$  be two  $m \times n$  matrices of reals.

- (i) Show  $\|\mathbf{F}\| = 0$  if and only if  $\mathbf{F} = \mathbf{O}$ ,  $\|\alpha\mathbf{F}\| = |\alpha|\|\mathbf{F}\|$  for all  $\alpha \in \mathbb{R}$ ,  $\|\mathbf{F} + \mathbf{R}\| \leq \|\mathbf{F}\| + \|\mathbf{R}\|$ .
- (ii) Show  $\|I_n\| = \sqrt{n}$  and  $\|\mathbf{F}\|^2 = \text{trace}(\mathbf{F}\mathbf{F}^T)$ .
- (iii) Show  $\|\mathbf{F}\|_2 \leq \|\mathbf{F}\| \leq \sqrt{n}\|\mathbf{F}\|_2$ . (Hint: Use  $\text{trace}(\mathbf{F}^T\mathbf{F}) = \sum \lambda_i$ .)
- (iv) For  $\mathbf{F}$  as before,  $\mathbf{R}$  of size  $n \times p$ ,  $\mathbf{r} \in \mathbb{R}^n$  and  $\alpha \in \mathbb{R}$  show

$$\|\mathbf{F}\mathbf{R}\| \leq \|\mathbf{F}\|\|\mathbf{R}\|, \quad \|\mathbf{F}(I_n - \alpha\mathbf{r}\mathbf{r}^T)\|^2 = \|\mathbf{F}\|^2 - \alpha(2 - \alpha\|\mathbf{r}\|^2)\|\mathbf{F}\mathbf{r}\|^2.$$


---

- (i) We have  $\|\mathbf{F}\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2} = 0$ , which is equivalent to  $f_j^i = 0$  for all  $1 \leq i \leq m$  and  $1 \leq j \leq n$ . Thus  $\mathbf{F} = \mathbf{O}$ .

To prove that  $\|\alpha \mathbf{F}\| = |\alpha| \|\mathbf{F}\|$  we calculate

$$\|\alpha \mathbf{F}\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (\alpha f_j^i)^2} = \sqrt{\alpha^2 \sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2} = |\alpha| \|\mathbf{F}\|$$

To prove that  $\|\mathbf{F} + \mathbf{R}\| \leq \|\mathbf{F}\| + \|\mathbf{R}\|$  we prove the equivalent inequality (since both parts are nonnegative)  $\|\mathbf{F} + \mathbf{R}\|^2 \leq (\|\mathbf{F}\| + \|\mathbf{R}\|)^2$ . We calculate

$$\begin{aligned} \|\mathbf{F} + \mathbf{R}\|^2 &= \sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2 + \sum_{i=1}^m \sum_{j=1}^n (r_j^i)^2 + 2 \sum_{i=1}^m \sum_{j=1}^n f_j^i r_j^i \\ (\|\mathbf{F}\| + \|\mathbf{R}\|)^2 &= \sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2 + \sum_{i=1}^m \sum_{j=1}^n (r_j^i)^2 + 2 \sqrt{\left(\sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2\right) \left(\sum_{i=1}^m \sum_{j=1}^n (r_j^i)^2\right)} \end{aligned}$$

and thus the inequality to prove becomes

$$\sum_{i=1}^m \sum_{j=1}^n f_j^i r_j^i \leq \sqrt{\sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2} \sqrt{\sum_{i=1}^m \sum_{j=1}^n (r_j^i)^2}$$

which follows from the Cauchy-Schwarz inequality.

**(ii)** The first part follows directly from the definition of the Frobenius norm:

$$\|\mathbf{I}_n\| = \sqrt{\sum_{k=1}^n \sum_{j=1}^n \delta_j^k} = \sqrt{\sum_{i=1}^n 1} = \sqrt{n},$$

where  $\delta_j^j = 1$ ,  $\delta_j^k = 0$ . To prove the second part, let  $\mathbf{G} = \mathbf{F}\mathbf{F}^T$ . Then  $g_k^k = \sum_{j=1}^n (f_j^k)^2$ . Now we have

$$\text{trace}(\mathbf{F}\mathbf{F}^T) = \sum_{k=1}^m g_k^k = \sum_{k=1}^m \sum_{j=1}^n (f_j^k)^2 = \|\mathbf{F}\|^2.$$

**(iii)** The matrix  $\mathbf{F}^T \mathbf{F}$  is symmetric. Thus  $\text{trace}(\mathbf{F}^T \mathbf{F}) = \sum_{i=1}^n \lambda_i$  where  $\lambda_i$  are the eigenvalues of  $\mathbf{F}^T \mathbf{F}$ . Since  $\mathbf{F}^T \mathbf{F}$  is also positive semi-definite we have  $\lambda_i \geq 0$  for all  $1 \leq i \leq n$ . Let  $\Lambda = \max\{\lambda_i : 1 \leq i \leq n\}$ . Then we calculate

$$\|\mathbf{F}\|_2 = \sqrt{\Lambda} \leq \sqrt{\sum_{i=1}^n \lambda_i} = \|\mathbf{F}\|$$

where we have used the second part of (ii) for the last equality. Moreover,

$$\|\mathbf{F}\| = \sqrt{\sum_{i=1}^n \lambda_i} \leq \sqrt{n\Lambda} = \sqrt{n} \|\mathbf{F}\|.$$

**(iv)** From the Cauchy-Schwarz inequality we have  $(\mathbf{f}^i \mathbf{r}_j)^2 \leq \|\mathbf{f}^i\|^2 \|\mathbf{r}_j\|^2$ , where  $\mathbf{f}^i$  is the  $i$ -th row of  $\mathbf{F}$  and  $\mathbf{r}_j$  the  $j$ -th column of  $\mathbf{R}$ . Summing over  $j$  first and over  $i$  next, we get

$$\sum_{i=1}^m \sum_{j=1}^p (\mathbf{f}^i \mathbf{r}_j)^2 \leq \sum_{i=1}^m \|\mathbf{f}^i\|^2 \sum_{j=1}^p \|\mathbf{r}_j\|^2$$

which using  $\mathbf{f}^i \mathbf{r}_j = \sum_{k=1}^n f_k^i r_j^k$  gives

$$\sum_{i=1}^m \sum_{j=1}^p \left( \sum_{k=1}^n f_k^i r_j^k \right)^2 \leq \left( \sum_{i=1}^m \sum_{j=1}^p (f_j^i)^2 \right) \left( \sum_{j=1}^p \sum_{i=1}^m (r_j^i)^2 \right).$$

Using the definition of the Frobenius norm we get  $\|\mathbf{FR}\|^2 \leq \|\mathbf{F}\|^2 \|\mathbf{R}\|^2$  which proves  $\|\mathbf{FR}\| \leq \|\mathbf{F}\| \|\mathbf{R}\|$ , since  $\|\cdot\| \geq 0$ . To prove the second relation, we calculate

$$\begin{aligned} \|\mathbf{F}(\mathbf{I}_n - \alpha \mathbf{rr}^T)\|^2 &= \text{trace}(\mathbf{F}(\mathbf{I}_n - \alpha \mathbf{rr}^T)(\mathbf{I}_n - \alpha \mathbf{rr}^T)\mathbf{F}^T) = \text{trace}((\mathbf{F} - \alpha \mathbf{Fr} \mathbf{r}^T)(\mathbf{F}^T - \alpha \mathbf{r} \mathbf{r}^T \mathbf{F}^T)) \\ &= \text{trace}(\mathbf{FF}^T - \alpha \mathbf{Fr} \mathbf{r}^T \mathbf{F} - \alpha \mathbf{Fr} \mathbf{r}^T \mathbf{F}^T + \alpha^2 \mathbf{Fr} \mathbf{r}^T \mathbf{r} \mathbf{r}^T \mathbf{F}^T) \\ &= \text{trace}(\mathbf{FF}^T) - 2\alpha \|\mathbf{Fr}\|^2 + \alpha^2 \|\mathbf{r}\|^2 \|\mathbf{Fr}\|^2 = \|\mathbf{F}\|^2 - \alpha(2 - \alpha \|\mathbf{r}\|^2) \|\mathbf{Fr}\|^2. \end{aligned}$$

### Exercise 9.3

Let  $\mathbf{Q} = \mathbf{FF}^T$  and  $\mathbf{Q}_P = \mathbf{F}_P \mathbf{F}_P^T$  where

$$\mathbf{F}_P = \sqrt{\frac{n^2}{n^2 - 1}} \mathbf{F} \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n-1}{n+1}} \right) \mathbf{dd}^T \right).$$

Show that

$$\begin{aligned} \mathbf{Q}_P &= \frac{n^2}{n^2 - 1} \mathbf{Q} \left( \mathbf{I}_n - \frac{2}{n+1} \frac{\mathbf{aa}^T \mathbf{Q}}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right), \\ \mathbf{Q}_P^{-1} &= \frac{n^2 - 1}{n^2} \left( \mathbf{Q}^{-1} + \frac{2}{n-1} \frac{\mathbf{aa}^T}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right). \end{aligned}$$

For  $\mathbf{Q}_P$  we calculate using (9.13)

$$\begin{aligned} \mathbf{Q}_P &= \mathbf{F}_P \mathbf{F}_P^T = \frac{n^2}{n-1} \mathbf{F} \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n-1}{n+1}} \mathbf{dd}^T \right) \right) \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n-1}{n+1}} \mathbf{dd}^T \right) \right) \mathbf{F}^T \\ &= \frac{n^2}{n^2 - 1} \left[ \mathbf{FF}^T - 2 \left( 1 - \sqrt{\frac{n-1}{n+1}} \right) \mathbf{F} \mathbf{dd}^T \mathbf{F}^T + \left( 1 - \sqrt{\frac{n-1}{n+1}} \right)^2 \mathbf{F} \mathbf{dd}^T \mathbf{dd}^T \mathbf{F}^T \right] \end{aligned}$$

or factoring out the scalar  $\mathbf{d}^T \mathbf{d} = \|\mathbf{d}\|^2 = 1$  from the last term

$$= \frac{n^2}{n^2 - 1} \left[ \mathbf{Q} - \left( 2 - 2\sqrt{\frac{n-1}{n+1}} - 1 - \frac{n-1}{n+1} + 2\sqrt{\frac{n-1}{n+1}} \right) \mathbf{F} \mathbf{d} \mathbf{d}^T \mathbf{F}^T \right]$$

and since  $\mathbf{F} \mathbf{d} \mathbf{d}^T \mathbf{F} = \frac{\mathbf{Q} \mathbf{a} \mathbf{a}^T \mathbf{Q}}{\mathbf{a}^T \mathbf{Q} \mathbf{a}}$  because  $\mathbf{d} = \mathbf{F}^T \mathbf{a} / \|\mathbf{F}^T \mathbf{a}\|$  and  $\|\mathbf{d}\| = 1$

$$= \frac{n^2}{n^2 - 1} \mathbf{Q} \left( \mathbf{I}_n - \frac{2}{n+1} \frac{\mathbf{a} \mathbf{a}^T \mathbf{Q}}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right).$$

For  $\mathbf{Q}_P^{-1}$  we calculate using the formula (9.22) for  $\mathbf{F}_P^{-1}$

$$\begin{aligned} \mathbf{Q}_P^{-1} &= (\mathbf{F}_P^{-1})^T \mathbf{F}_P^{-1} = \frac{n-1}{n^2} (\mathbf{F}^{-1})^T \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n+1}{n-1}} \mathbf{d} \mathbf{d}^T \right) \right) \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{n+1}{n-1}} \mathbf{d} \mathbf{d}^T \right) \right) \mathbf{F}^{-1} \\ &= \frac{n^2 - 1}{n^2} \left[ (\mathbf{F}^{-1})^T \mathbf{F}^{-1} - 2 \left( 1 - \sqrt{\frac{n+1}{n-1}} \right) (\mathbf{F}^{-1})^T \mathbf{d} \mathbf{d}^T \mathbf{F}^{-1} + \left( 1 - \sqrt{\frac{n+1}{n-1}} \right)^2 (\mathbf{F}^{-1})^T \mathbf{d} \mathbf{d}^T \mathbf{d} \mathbf{d}^T \mathbf{F}^{-1} \right] \end{aligned}$$

or factoring out the scalar  $\mathbf{d}^T \mathbf{d} = \|\mathbf{d}\|^2 = 1$  from the last term

$$\begin{aligned} &= \frac{n^2 - 1}{n^2} \left[ \mathbf{Q}^{-1} - \left( 2 - 2\sqrt{\frac{n+1}{n-1}} - 1 - \frac{n+1}{n-1} + 2\sqrt{\frac{n+1}{n-1}} \right) (\mathbf{F}^{-1})^T \mathbf{d} \mathbf{d}^T \mathbf{F}^{-1} \right] \\ &= \frac{n^2 - 1}{n^2} \mathbf{Q}^{-1} \left( \mathbf{I}_n + \frac{2}{n+1} \frac{\mathbf{a} \mathbf{a}^T}{\mathbf{a}^T \mathbf{Q} \mathbf{a}} \right) \end{aligned}$$

where we have used  $\|\mathbf{d}\| = 1$  and  $\mathbf{d}^T \mathbf{F}^{-1} = \mathbf{a}^T / \|\mathbf{F}^T \mathbf{a}\|$  which follows from the definition of  $\mathbf{d}$ ; see (9.12).

---

### Exercise 9.4

Let  $\mathbf{Q}_k = \mathbf{F}_k \mathbf{F}_k^T$  be the positive definite matrix that defines the ellipsoid  $E_k$ . Denote by  $\lambda_{min}^k$  the smallest and by  $\lambda_{max}^k$  the largest eigenvalue of  $\mathbf{Q}_k$ . Prove that  $\lambda_{min}^k \leq R^2 2^{-k/2n^2}$  and  $\lambda_{max}^k \geq R^2 2^{-2k/n^2}$  for all  $k$  of the basic ellipsoid algorithm. (Hint: Use (9.35).)

---

Since  $\mathbf{Q}_k$  is positive definite and nonsingular we have  $\lambda_{min}^k > 0$ . From  $\mathbf{Q}_k = \mathbf{F}_k \mathbf{F}_k^T$  we have that the eigenvalues of  $\mathbf{F}_k$  are  $|\mu_i| = \sqrt{\lambda_i}$  and thus  $|\det \mathbf{F}| = \prod_{i=1}^n \sqrt{\lambda_i}$ . It follows that

$$(\lambda_{min}^k)^{n/2} \leq |\det \mathbf{F}| \leq (\lambda_{max}^k)^{n/2}.$$

From (9.35) we have

$$(\lambda_{min}^k)^{n/2} \leq |\det \mathbf{F}| \leq R^n 2^{-\frac{k}{4n}} \Rightarrow \lambda_{min}^k \leq R^2 2^{-\frac{k}{2n^2}}$$

and similarly

$$(\lambda_{\max}^k)^{n/2} \geq |\det \mathbf{F}| \geq R^n 2^{-\frac{k}{n}} \Rightarrow \lambda_{\max}^k \geq R^2 2^{-\frac{2k}{n^2}}.$$


---

### Exercise 9.5

- (i) Write a computer program of the DCS ellipsoid algorithm in a computer language of your choice for the linear programming problem (LP).
  - (ii) Solve the problems of Exercises 5.1, 5.9 and 8.2(ii) using your computer program.
- 

**(i)** The following listing is an implementation of the algorithm in MATLAB. The required input is as in the simplex algorithm; see Exercise 5.2.

```
%%%%%
%% This is the implementation of the DCS Ellipsoid algorithm
%% as found on pages 309-310.
%%
%% NAME      : dcsel
%% PURPOSE: Solve the LP: max {cx: a~x <= b, x >=0}
%% INPUT    : The matrix a~and the vectors c and b.
%% OUTPUT   : z : the optimal value
%%             x : the optimal solution
%%             k : the number of iterations
%%%%%
[m,n]=size(A);
A = [A;-eye(n)];
b = [b zeros(1,n)];
[m,n]=size(A);
zl=-10000;
zu=10000;
eps=10^(-9);
R=500;
Vf=10^(-6);

R0=sqrt(n)*(1+R/2);
veps=eps*ones(m,1);
z=zl+1;
z0=z;
x = (R/2)*ones(n,1);
H=R0*eye(n);
```

```

k=0;
f0=(1+1/n)^(-(n+1)/2) * (1-1/n)^(-(n-1)/2);
V=R0^n * pi^(n/2) / gamma(1+n/2);

while (1<2)
    slack=b'+veps- A*x ;
    [mxv,j]=min(slack);
    if (mxv >= 0)
        xstar=x+H'*c'/norm(c*H);
        help=A*(xstar-x);
        if (all(help <= 0)), error('Unbounded'), end;
        lambda=10000;
        for i=1:m,
            if (help(i) > 0)
                if (slack(i) /help(i) <= lambda)
                    lambda=slack(i)/help(i);
                end;
            end;
        end;
    if (lambda >= 1), error('Unbounded'), end;
    if (c*(x+lambda*(xstar-x)) > z)
        xbar=x+lambda*(xstar-x);
        z=c*xbar;
    end;
end;
if ((c*x > z & z0 > zl+1) | (mxv <0 & z0-z > mxv))
    theta=b(j)+eps;
    alpha=(A(j,:)*x-theta)/norm(A(j,:)*H);
    r=A(j,:)';
else
    alpha=(z-c*x)/norm(c*H);
    r=-c';
    z0=z;
end;
if ((alpha >=1 | V < Vf) & (z < zu & z > zl))
    fprintf('Optimal solution found in %d iterations, Vol= %8.4f\n',k-1,V);
    fprintf('%8.4f',xbar);
    fprintf('\nz=%8.4f\n',z);
    return;
elseif (z <= zl)
    fprintf('Infeasible');
    return;
elseif (z >= zu)
    fprintf('Unbounded');
    return;
end;

```

```

xold=x;
x=x-(1+n*alpha)*H'*r/((n+1)*norm(H'*r));
quan1=sqrt(((n-1)*(1-alpha))/((n+1)*(1+alpha)));
quan2=n*sqrt((1-alpha^2)/(n^2-1));
H=quan2*H*(eye(n)-(1-quan1)*H'*r*r'*H/norm(H'*r)^2);
V=(1-alpha)*f0*V*(1-alpha^2)^( (n-1)/2 );
fprintf(' %10.5f ',x);
fprintf(' cx=%10.5f, z=%10.5f\n',c*x,z);
z1=c*(x-xold);
help1=A*(x-xold);
if (all(help1 < 0) & z1 > 0), error('Unbounded'), end;
slack1=b'+veps-A*xold;
mumax=-10000;
mumin=10000;
substep=1;
for i=1:m,
    if (help1(i) > 0 )
        if (slack1(i)/help1(i) <=mumin)
            mumin=slack1(i)/help1(i);
        end;
    elseif (help1(i) < 0)
        if (slack1(i)/help1(i) >= mumax)
            mumax=slack1(i)/help1(i);
        end;
    else
        if (slack(i) < 0)
            substep=0;
        end;
    end;
end;
if (mumin < mumax & substep > 0)
    substep=0;
end;
if (substep > 0 )
    if (z1 < 0)
        mubar=mumax;
    else
        mubar=mumin;
    end;
    if (c*xold+mubar*z1 > z)
        xbar=xold+mubar*(x-xold);
        z=c*xbar;
    end;
end;
k=k+1;
end;

```

(ii) For the data of Exercise 5.1 we get

```
>> clear
>> psdat
>> dcsel
-4.03134  165.32289  -88.70846   80.64577  cx= 294.36367, z=-9999.00000
-10.02109  30.06256   -30.06291   10.02074  cx= -30.06467, z=-9999.00000
-52.62810  42.08574   12.86073   -5.27118  cx= 61.90156, z=-9999.00000
  9.55652   35.42522   -3.34090   -46.70324  cx= 18.61864, z=-9999.00000
  3.54364   -1.94823   -14.64155    9.78702  cx= -37.74956, z=-9999.00000
-3.83235   -9.72340    3.63685    4.58637  cx= -13.11476, z=-9999.00000
-8.02842    7.99811    2.99137   -2.24117  cx= 15.42061, z=-9999.00000
  3.97518    4.26213    1.43981   -7.85914  cx= 10.77769, z=-9999.00000
  2.39358   -1.88535   -0.05693    4.51311  cx= 7.92960, z=-9999.00000
  2.06213    2.37324    0.89350    4.29667  cx= 23.41133, z= 8.96852
-0.41061    5.66114   -1.79996    2.86110  cx= 14.68454, z= 8.96852
-2.33922    3.31667    2.20371    1.33717  cx= 16.76076, z= 8.96852
  2.51569    1.14168    1.38714   -0.54868  cx= 12.90763, z= 8.96852
  2.19221    5.93326    1.48589   -1.14267  cx= 25.84241, z= 14.90422
.
.
.
  0.00350    9.34385    0.65380    0.00101  cx= 30.65576, z= 30.64666
  0.00065    9.33560    0.66299   -0.00319  cx= 30.65366, z= 30.64666
-0.00233    9.32698    0.66508    0.00744  cx= 30.65146, z= 30.64666
  0.00564    9.32288    0.66461    0.00532  cx= 30.64897, z= 30.64666
  0.00410    9.33853    0.65410    0.00306  cx= 30.64632, z= 30.65073
  0.00258    9.34019    0.65755    0.00082  cx= 30.65757, z= 30.65073
  0.00066    9.33448    0.66385   -0.00201  cx= 30.65616, z= 30.65073
Optimal solution found in 119 iterations, Vol= 0.0000
  0.0021  9.3386  0.6593  0.0000
z= 30.6572
>>
```

For Exercise 5.9 with  $n = 3$ ,  $a = b = 2$  and  $c = 5$ , we get

```
>> clear
>> psdat
>> dcsel
-85.49883   82.25058   208.06265  cx= 30.56848, z=-9999.00000
  11.12853  -64.29617   171.42596  cx= 87.34772, z=-9999.00000
  1.20391   12.52250   126.74039  cx= 156.60104, z=-9999.00000
-10.39740   2.76268    74.50551  cx= 38.44128, z=-9999.00000
  1.90067  -6.57926   24.50713  cx= 18.95129, z=-9999.00000
  1.38161   3.95126   -6.60592  cx= 6.82304, z=-9999.00000
  0.58555   2.21549   17.22113  cx= 23.99432, z=-9999.00000
-0.03957   0.85243    9.95152  cx= 11.49810, z=-9999.00000
-0.53385  -0.22534   29.23529  cx= 26.64921, z= 19.92770
  1.33403  -1.70743   24.00886  cx= 25.93012, z= 19.92770
```

```

0.98365 -3.93039 26.10890 cx= 22.18273, z= 19.92770
0.15010 1.90863 13.56085 cx= 17.97850, z= 19.92770
-0.29774 1.49264 22.51259 cx= 24.30692, z= 20.09956
-0.92891 0.90634 24.64934 cx= 22.74638, z= 20.09956
0.71750 -0.20129 18.03809 cx= 20.50551, z= 20.09956
0.55850 -0.93332 23.94083 cx= 24.30821, z= 21.48207
0.29328 1.24644 19.78129 cx= 23.44728, z= 21.48207
.
.
.
-0.00120 -0.00039 25.00568 cx= 25.00008, z= 24.99545
0.00141 -0.00352 25.00029 cx= 24.99888, z= 24.99545
0.00057 0.00211 24.99013 cx= 24.99661, z= 24.99545
-0.00020 0.00127 24.99280 cx= 24.99452, z= 24.99545
Optimal solution found in 68 iterations, Vol= 0.0000
0.0002 0.0017 24.9913
z= 24.9957
>>

```

For the data of Exercise 8.2(ii) we get

```

>> clear
>> psdat
>> dcsel
194.74376 -26.28120 cx= 26.28120, z=-9999.00000
-0.42637 25.28755 cx= -25.28755, z=-9999.00000
-20.12962 -1.44852 cx= 1.44852, z= -14.24470
34.80955 1.12224 cx= -1.12224, z= -14.24470
11.15640 9.73883 cx= -9.73883, z= 0.00000
-8.55908 -8.48696 cx= 8.48696, z= 0.00000
14.71900 -5.87632 cx= 5.87632, z= 0.00000
6.88480 -1.39700 cx= 1.39700, z= 0.00000
14.14111 5.56176 cx= -5.56176, z= 0.00000
6.40835 -1.85392 cx= 1.85392, z= 0.00000
8.98593 0.61797 cx= -0.61797, z= 0.00000
8.12674 -0.20599 cx= 0.20599, z= 0.00000
8.41314 0.06866 cx= -0.06866, z= 0.00000
8.31767 -0.02289 cx= 0.02289, z= 0.00000
8.34949 0.00763 cx= -0.00763, z= 0.00000
8.33889 -0.00254 cx= 0.00254, z= 0.00000
8.34242 0.00085 cx= -0.00085, z= 0.00000
8.34124 -0.00028 cx= 0.00028, z= 0.00000
8.34164 0.00009 cx= -0.00009, z= 0.00000
8.34151 -0.00003 cx= 0.00003, z= 0.00000
8.34155 0.00001 cx= -0.00001, z= 0.00000
8.34153 -0.00000 cx= 0.00000, z= 0.00000
8.34154 0.00000 cx= -0.00000, z= 0.00000
8.34154 0.00000 cx= -0.00000, z= 0.00000

```

```

8.34154 - 0.00000 cx= 0.00000, z= 0.00000
8.34154 0.00000 cx= -0.00000, z= 0.00000
8.34154 -0.00000 cx= 0.00000, z= 0.00000
Optimal solution found in 27 iterations, Vol= 0.0000
10.8264 -0.0000
z= 0.0000
>>

```

---

### Exercise 9.6

(i) Let  $\mathbf{H}_{k+1}$  be as defined in (9.45), i.e.,

$$\mathbf{H}_{k+1} = n \sqrt{\frac{1 - \alpha_k^2}{n^2 - 1}} \mathbf{H}_k \left( \mathbf{I}_n - \left( 1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}} \right) \frac{(\mathbf{H}_k^T \mathbf{r})(\mathbf{r}^T \mathbf{H}_k)}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right).$$

Like we proved (9.20) show that

$$\det \mathbf{H}_{k+1} = \left( 1 + \frac{1}{n} \right)^{-\frac{n+1}{2}} \left( 1 - \frac{1}{n} \right)^{-\frac{n-1}{2}} (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k) \det \mathbf{H}_k.$$

(ii) Define  $\mathbf{G}_k = \mathbf{H}_k \mathbf{H}_k^T$  and show using (9.45) that

$$\mathbf{G}_{k+1} = \frac{n^2(1 - \alpha_k^2)}{n^2 - 1} \mathbf{G}_k \left( \mathbf{I}_n - \frac{2(1 + n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right),$$

$$\mathbf{G}_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left( \mathbf{G}_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1-\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right).$$

(iii) Show  $\Gamma(1 + n/2) = (n/2)!$  for all even  $n \geq 1$ ,  $\Gamma(1 + n/2) = \frac{n! \sqrt{\pi}}{\lfloor \frac{n}{2} \rfloor! 2^n}$  for all odd  $n \geq 1$ .

---

(i) Letting  $\alpha = 1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}$  the proof of (9.20) applies unchanged since  $0 \leq \alpha_k < 1$ . Thus replacing  $F_P$  by  $\mathbf{H}_{k+1}$ ,  $F$  by  $\mathbf{H}_k$  and  $d$  by  $\mathbf{H}_k^T \mathbf{r} / \|\mathbf{H}_k^T \mathbf{r}\|$ , since  $\|d\| = 1$  we get

$$\det \mathbf{H}_{k+1} = \left( \frac{n^2(1 - \alpha_k^2)}{n^2 - 1} \right)^{n/2} \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}} \det \mathbf{H}_k.$$

To bring this to the required form we have

$$\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}} = \sqrt{\frac{(n-1)n(1-\alpha_k)^2}{(n+1)n(1-\alpha_k^2)}} = \left( 1 - \frac{1}{n} \right)^{1/2} \left( 1 + \frac{1}{n} \right)^{-1/2} (1 - \alpha_k)(1 - \alpha_k^2)^{-1/2}$$

and

$$\frac{n^2}{n^2 - 1}(1 - \alpha_k^2) = \frac{n}{n+1} \frac{n}{n-1}(1 - \alpha_k^2) = \left(1 + \frac{1}{n}\right)^{-1} \left(1 - \frac{1}{n}\right)^{-1} (1 - \alpha_k^2).$$

Thus after grouping of terms we get

$$\det \mathbf{H}_{k+1} = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k) \det \mathbf{H}_k.$$

**(ii)** Using the definition of  $\mathbf{H}_{k+1}$  we calculate:

$$\begin{aligned} \mathbf{G}_{k+1} &= \mathbf{H}_{k+1} \mathbf{H}_{k+1}^T = n^2 \frac{1 - \alpha_k^2}{n^2 - 1} \left( \mathbf{H}_k - \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right) \frac{\mathbf{H}_k \mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right) \\ &\quad \left( \mathbf{H}_k^T - \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right) \frac{\mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right) \\ &= n^2 \frac{1 - \alpha_k^2}{n^2 - 1} \left[ \mathbf{H}_k \mathbf{H}_k^T - 2 \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right) \left( \frac{\mathbf{H}_k \mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right) \right. \\ &\quad \left. + \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)^2 \frac{\mathbf{H}_k \mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T}{\|\mathbf{H}_k^T \mathbf{r}\|^4} \right] \end{aligned}$$

and factoring out the term  $\mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T \mathbf{r} = \|\mathbf{H}_k^T \mathbf{r}\|^2$  from the numerator of the last fraction

$$\begin{aligned} &= \frac{n^2(1-\alpha_k^2)}{n^2 - 1} \left( \mathbf{G}_k - \left(2 \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right) - \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)^2\right) \frac{\mathbf{H}_k \mathbf{H}_k^T \mathbf{r} \mathbf{r}^T \mathbf{H}_k \mathbf{H}_k^T}{\|\mathbf{H}_k^T \mathbf{r}\|^2} \right) \\ &= \frac{n^2(1-\alpha_k^2)}{n^2 - 1} \left( \mathbf{G}_k - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{G}_k \mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right) = \frac{n^2(1-\alpha_k^2)}{n^2 - 1} \mathbf{G}_k \left( \mathbf{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right) \end{aligned}$$

For the inverse  $\mathbf{G}_{k+1}^{-1}$  we have

$$\mathbf{G}_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left( \mathbf{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right)^{-1} \mathbf{G}_k^{-1}.$$

Using Exercise 4.1 (ii) with  $\mathbf{u} = -\frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)(\mathbf{r}^T \mathbf{G}_k \mathbf{r})} \mathbf{r}$  and  $\mathbf{v} = \mathbf{G}_k \mathbf{r}$  we calculate  $\mathbf{v}^T \mathbf{u} \neq -1$  because  $\alpha_k \neq 1$  and thus

$$\begin{aligned} \left( \mathbf{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right)^{-1} &= \mathbf{I}_n + \frac{1}{1 - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r}^T \mathbf{G}_k \mathbf{r}}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}}} \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \\ &= \mathbf{I}_n + \frac{2(1+n\alpha_k)}{n+n\alpha_k + 1 + \alpha_k - 2 - 2n\alpha_k} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \\ &= \mathbf{I}_n + \frac{2(1+n\alpha_k)}{(n-1)(1-\alpha_k)} \frac{\mathbf{r} \mathbf{r}^T \mathbf{G}_k}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}}. \end{aligned}$$

It follows that

$$\mathbf{G}_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left( \mathbf{G}_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1 - \alpha_k)} \frac{\mathbf{r}\mathbf{r}^T}{\mathbf{r}^T \mathbf{G}_k \mathbf{r}} \right).$$

**(iii)** For  $n > 1$  and even, i.e.  $n = 2k$  with  $k \geq 1$  is integer we have

$$\Gamma(1 + n/2) = \Gamma(1 + k) = k\Gamma(k) = k! = \left(\frac{n}{2}\right)!$$

where we have used elementary properties of the gamma function; see Chapter 7.7. For  $n \geq 1$  and odd, i.e.  $n = 2k + 1$  where  $k \geq 0$  is integer we have

$$\Gamma\left(\frac{2k+1}{2}\right) = \Gamma\left(\frac{2k-1}{2} + 1\right) = \frac{2k-1}{2}\Gamma\left(\frac{2k-1}{2}\right),$$

where the last equality follows from the fact that  $2k - 1$  is even. Thus we get

$$\Gamma\left(1 + \frac{n}{2}\right) = \Gamma\left(\frac{2k+3}{2}\right) = \frac{1}{2^{k+1}}\Gamma\left(\frac{1}{2}\right) \prod_{\ell=0}^k (2\ell + 1).$$

Using the identity  $\prod_{\ell=0}^k (2\ell + 1) = \frac{(2k+1)!}{k!2^k}$  and that  $\Gamma(\frac{1}{2}) = \sqrt{\pi}$  we get  $\Gamma\left(1 + \frac{n}{2}\right) = \frac{n!}{[\frac{n}{2}]!2^n}\sqrt{\pi}$ .

---

### Exercise 9.7

Let  $P \subseteq \mathbb{R}^n$  be a polyhedron and define  $P_\varepsilon^\infty = \{z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_\infty \leq \varepsilon\}$ .

- (i) Show that  $P_\varepsilon^\infty$  is a polyhedron in  $\mathbb{R}^n$  and that  $P \neq \emptyset$  implies  $\dim P_\varepsilon^\infty = n$  for  $\varepsilon > 0$ .
- (ii) Show that if  $z \in P_\varepsilon^\infty$  is an extreme point then there exists an extreme point  $x \in P$  such that  $z_i - x_i = \pm \varepsilon$  for all  $1 \leq i \leq n$ .
- (iii) Let  $y \in \mathbb{R}^n$ . Show that  $x + (y) \in P_\varepsilon^\infty$  for some  $x \in P_\varepsilon^\infty$  if and only if there exists  $\tilde{x} \in P$  such that  $\tilde{x} + (y) \in P$ . Show that the asymptotic cone  $C_\infty$  of  $P$  is the asymptotic cone of  $P_\varepsilon^\infty$ .
- (iv) Show that  $P_\varepsilon^\infty$  has at most  $p2^n$  extreme points where  $p$  is the number of extreme points of  $P$ .
- (v) Suppose that the facet complexity of  $P$  is  $\phi$  and its vertex complexity is  $\nu$ . Show that for rational  $\varepsilon \geq 0$  the polyhedron  $P_\varepsilon^\infty$  has a vertex complexity of  $2\nu + 2n\langle\varepsilon\rangle$  and a facet complexity of  $3\phi + 2\langle\varepsilon\rangle$ .
- (vi) Define  $P_\varepsilon^1$  by

$$P_\varepsilon^1 = \{z \in \mathbb{R}^n : \exists x \in P \text{ such that } \|x - z\|_1 \leq \varepsilon\}.$$

Show that  $P_\varepsilon^1$  is a polyhedron,  $\mathbf{h}x \leq h_0 + \varepsilon \|\mathbf{h}\|_\infty$  for all  $x \in P_\varepsilon^1$  if  $\mathbf{h}x \leq h_0$  for all  $x \in P$  and  $\mathbf{h}x \leq h_0 - \varepsilon \|\mathbf{h}\|_\infty$  for all  $x \in P$  if  $\mathbf{h}x \leq h_0$  for all  $x \in P_\varepsilon^1$ . (Hint: Use that  $\sum_{j=1}^n |x_j| \leq \varepsilon$  if and only if  $\sum_{j=1}^n \delta_j x_j \leq \varepsilon$  for the  $2^n$  vectors  $(\delta_1, \dots, \delta_n)$  with  $\delta_j \in \{+1, -1\}$  for all  $1 \leq j \leq n$ .)

- (vii) Show that if  $z \in P_\epsilon^1$  is an extreme point then  $z = x \pm \epsilon u^i$  where  $x \in P$  is an extreme point of  $P$  and  $u^i \in \mathbb{R}^n$  is the  $i$ -th unit vector for some  $1 \leq i \leq n$ . Show that  $P_\epsilon^1$  has at most  $2np$  extreme points if  $P$  has  $p$  extreme points.
- (viii) Suppose  $P$  has a facet complexity of  $\phi$  and a vertex complexity of  $\nu$ . Show that  $P_\epsilon^1$  has vertex complexity of  $2\nu + 2\langle \epsilon \rangle$  and a facet complexity of  $3\phi + 2\langle \epsilon \rangle$ .

(ix) Prove that

$$\text{if } P \neq \emptyset \text{ then } \text{vol}(P_\epsilon^1) \geq \frac{2^n \epsilon^n}{n!} > \frac{\epsilon^n \pi^{n/2}}{n^n \Gamma(1+n/2)} \text{ for all } \epsilon > 0.$$

(Hint: Use that  $P_\epsilon^1 \supseteq x + \{z \in \mathbb{R}^n : \|z\|_1 \leq 1\}$  and  $P_\epsilon^1 \supseteq B(x, r = \epsilon/n)$  for every  $x \in P$ .)

(x) Show that Remark 9.13 remains correct if we replace the  $\ell_1$ -norm by the  $\ell_\infty$ -norm.

---

**(i)** Let  $S = \{x^1, \dots, x^p\}$ ,  $T = \{r^1, \dots, r^t\}$  be any finite generator of  $P$ . Then every  $x \in P$  can be written as  $x = X\mu + R\nu$  where  $X = (x^1 \dots x^p)$  is an  $n \times p$  matrix,  $R = (r^1 \dots r^t)$  an  $n \times t$  matrix,  $\mu \geq 0$ ,  $f\mu = 1$ ,  $\nu \geq 0$  and  $f = (1, \dots, 1) \in \mathbb{R}^p$ . If  $P$  is pointed, then  $x^1, \dots, x^p$  are the extreme points and  $r^1, \dots, r^t$  are the direction vectors of the extreme rays of  $P$ . Let  $e^T = (1, \dots, 1) \in \mathbb{R}^n$ . Then  $P_\epsilon^\infty$  is the image of the polyhedron

$$PP_\epsilon = \{(z, \mu, \nu) \in \mathbb{R}^{n+p+t} : -\epsilon e \leq X\mu + R\nu - z \leq \epsilon e, f\mu = 1, \mu \geq 0, \nu \geq 0\}$$

under the linear transformation with the matrix  $L = (I_n \ O_p \ O_t)$ , i.e.  $P_\epsilon^\infty$  is the image of  $PP_\epsilon$  when we project out the variables  $\mu$  and  $\nu$ . By point 7.3(g) it follows that  $P_\epsilon^\infty$  is a polyhedron. To prove the dimension of  $P_\epsilon^\infty$ , suppose that  $P \neq \emptyset$  and let  $x \in P$ . Then the points  $x, x + \epsilon u^1, \dots, x + \epsilon u^n$  are  $n+1$  affinely independent points in  $P_\epsilon^\infty$  where  $u^i$  is the  $i$ -th unit vector in  $\mathbb{R}^n$ . Thus  $\dim P_\epsilon^\infty = n$ .

**(ii)** Let  $z \in P_\epsilon^\infty$  and  $x \in P$  be such that  $z_i - x_i = \pm \epsilon$  for all  $1 \leq i \leq n$  and suppose that  $z$  is an extreme point of  $P_\epsilon^\infty$  but  $x$  is not an extreme point of  $P$ . It follows that  $x = \mu x^1 + (1-\mu)x^2$  where  $0 < \mu < 1$ ,  $x^1, x^2 \in P$  and  $x^1 \neq x^2$ ,  $x^1 \neq x \neq x^2$ . We then get  $z = x \pm \epsilon e = \mu x^1 + (1-\mu)x^2 \pm (\mu + 1 - \mu)\epsilon e = \mu(x^1 \pm \epsilon e) + (1-\mu)(x^2 \pm \epsilon e)$ . From the definition of  $P_\epsilon$  it follows that  $z^1 = x^1 \pm \epsilon e \in P_\epsilon^\infty$  and  $z^2 = x^2 \pm \epsilon e \in P_\epsilon^\infty$ . Moreover,  $z^1 \neq z^2$  and thus  $z = \mu z^1 + (1-\mu)z^2$  with  $0 < \mu < 1$  contradicts the assumption that  $z$  is an extreme point of  $P_\epsilon^\infty$ .

**(iii)** From Remark 9.12 (ii) we have that the asymptotic cones of  $P$  and  $P_\epsilon^\infty$  are the same, since every valid inequality  $hx \leq h_0$  for one of the polyhedra gives rise to a valid inequality of the form  $hx \leq h'_0$  for the other and vice versa. An inequality is valid for  $P$ , if it is satisfied by every  $x \in P$ . To prove the first part suppose that  $(y) \in P_\epsilon^\infty$  is a halfline of  $P_\epsilon^\infty$ , i.e.  $hy \leq 0$  for all  $h$  such that  $hx \leq h_0$  is valid for  $P_\epsilon^\infty$ . By part (ii) of Remark 9.12 we have that  $hx \leq h_0 - \epsilon \|h\|_1$  is valid for  $P$  for all such  $h$ . Suppose that  $(y)$  is not a halfline of  $P$ , i.e., there exists a valid inequality  $hx \leq h_0$  for  $P_\epsilon^\infty$  such that for some  $x \in P$  there exists  $\alpha > 0$  such that  $h(x + \alpha y) > h_0 - \epsilon \|h\|_1$ . Then  $\alpha hy > 0$  for some  $\alpha > 0$ , i.e.,  $hy > 0$  which contradicts the assumption that  $(y)$  is a halfline of  $P_\epsilon^\infty$ . On the other hand, suppose that  $(y)$  is a halfline of  $P$ . Then  $hy \leq 0$  for all  $h$  such that  $hx \leq h_0$  is valid for  $P$ . By point (ii) of Remark 9.12 we have that  $hx \leq h_0 + \epsilon \|h\|_1$  for all  $x \in P_\epsilon^1$  for all such  $(h, h_0)$ . Suppose that  $(y)$  is not a halfline of  $P_\epsilon^\infty$ . Then there exists a valid inequality  $hx \leq h_0$  for  $P$  such that for some  $x \in P_\epsilon^\infty$  there exists  $\alpha > 0$  such that  $h(x + \alpha y) > h_0 + \epsilon \|h\|_1$ . Consequently

we have  $h_0 \leq h_0 + \varepsilon \|\mathbf{h}\|_1 < \mathbf{h}(\mathbf{x} + \alpha\mathbf{y}) \leq h_0 + \alpha\mathbf{h}\mathbf{y}$ , i.e.  $\alpha\mathbf{h}\mathbf{y} > 0$  for some  $\alpha > 0$  which contradicts the assumption that  $(\mathbf{y})$  is a halfline of  $P$ .

**(iv)** From part (ii) we have that for each extreme point  $\mathbf{x}$  of  $P$  there exists an extreme point  $\mathbf{z}$  of  $P_\varepsilon^\infty$  such that  $z_i = x_i \pm \varepsilon$ , for  $1 \leq i \leq n$ . That is each component of  $\mathbf{x}$  can be either increased or decreased by  $\varepsilon$ . So for each extreme point of  $P$  there are at most  $2^n$  different points  $\mathbf{z}$  that can be constructed as above. Thus if  $P$  has  $p$  extreme points, then  $P_\varepsilon^\infty$  has at most  $p2^n$  extreme points.

**(v)** From Remark 9.12 (ii) we have that  $\mathbf{h}\mathbf{z} \leq h_0 + \varepsilon$  is a valid inequality for  $P_\varepsilon^\infty$  if  $\mathbf{h}\mathbf{x} \leq h_0$  is valid for  $P$ . Thus for the facet complexity of  $P_\varepsilon^\infty$  we calculate

$$\langle \mathbf{h} \rangle + \langle h_0 + \varepsilon \|\mathbf{h}\|_1 \rangle = \langle \mathbf{h} \rangle + 2\langle h_0 \rangle + 2\varepsilon + 2\langle \|\mathbf{h}\|_1 \rangle \leq 3\langle \mathbf{h} \rangle + 2\langle h_0 \rangle + 2\langle \varepsilon \rangle + 2(1-n) \leq 3(\langle \mathbf{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle \leq 3\phi + 2\langle \varepsilon \rangle$$

where we have used  $\langle \|\mathbf{h}\|_1 \rangle \leq \langle \mathbf{h} \rangle - n + 1$ ; see Chapter 7.5. For the vertex complexity, from part (ii) of this exercise we have that if  $\mathbf{z}$  is an extreme point of  $P_\varepsilon^\infty$  then  $z_i = x_i \pm \varepsilon$  for  $1 \leq i \leq n$  where  $\mathbf{x}$  is an extreme point of  $P$ . We calculate

$$\langle \mathbf{z} \rangle = \sum_{i=1}^n \langle z_i \rangle = \sum_{i=1}^n \langle x_i \pm \varepsilon \rangle \leq 2 \sum_{i=1}^n (\langle x_i \rangle + \langle \varepsilon \rangle) \leq 2 \sum_{i=1}^n \langle x_i \rangle + 2n\langle \varepsilon \rangle = 2\langle \mathbf{x} \rangle + 2n\langle \varepsilon \rangle \leq 2\nu + 2n\langle \varepsilon \rangle .$$

**(vi)** Let  $\Delta = (\delta^k)_{k=1}^{2^n}$  be the matrix with rows the  $2^n$  vectors with components  $\pm 1$ . The constraint  $\|\mathbf{x} - \mathbf{z}\|_1 \leq \varepsilon$  is written as  $\|\mathbf{x} - \mathbf{z}\|_1 = \sum_{i=1}^n |x_i - z_i| \leq \varepsilon$  and it is equivalent to the constraints  $\Delta(\mathbf{x} - \mathbf{z}) \leq \varepsilon \mathbf{e}$ ; see also Exercise 2.2(ii). So we have that

$$P_\varepsilon^1 = \{\mathbf{z} \in \mathbb{R}^n : \exists \mathbf{x} \in P \text{ with } \Delta(\mathbf{x} - \mathbf{z}) \leq \varepsilon \mathbf{e}\} .$$

Since  $P$  is a polyhedron, let like in part (i)  $\mathbf{X}$ ,  $\mathbf{R}$  be the matrices corresponding to a finite generator of  $P$ . Then

$$\mathbf{x} = \mathbf{X}\boldsymbol{\mu} + \mathbf{R}\boldsymbol{\nu} \text{ where } \boldsymbol{\mu} \geq 0, \mathbf{f}\boldsymbol{\mu} = 1, \boldsymbol{\nu} \geq 0$$

and  $\mathbf{f} = (1, \dots, 1)^T \in \mathbb{R}^p$ .  $P_\varepsilon^1$  is the image of the polyhedron

$$PP_\varepsilon^1 = \{(z, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^{n+p+t} : \Delta(\mathbf{X}\boldsymbol{\mu} + \mathbf{R}\boldsymbol{\nu} - \mathbf{z}) \leq \varepsilon \mathbf{e}, \mathbf{f}\boldsymbol{\mu} = 1, \boldsymbol{\mu} \geq 0, \boldsymbol{\nu} \geq 0\}$$

under the linear transformation  $\mathbf{L} = (I_n \ \mathbf{O}_p \ \mathbf{O}_t)$  where  $\mathbf{O}_p$  is  $n \times p$  and  $\mathbf{O}_t$  is  $n \times t$ , i.e. when we project out  $\boldsymbol{\mu}$  and  $\boldsymbol{\nu}$ . It follows by point 7.3(g) that  $P_\varepsilon^1$  is a polyhedron.

Suppose now that  $\mathbf{z} \in P_\varepsilon^1$ . Then there exists  $\mathbf{x} \in P$  such that  $\sum_{i=1}^n |x_i - z_i| \leq \varepsilon$ . We calculate  $\mathbf{h}\mathbf{z} - \mathbf{h}\mathbf{x} = \mathbf{h}(\mathbf{x} - \mathbf{z}) \leq \sum_{i=1}^n |h_i||z_i - x_i| \leq \sum_{i=1}^n \|\mathbf{h}\|_\infty |z_i - x_i| \leq \varepsilon \|\mathbf{h}\|_\infty$ . Since for every  $\mathbf{x} \in P$  we have  $\mathbf{h}\mathbf{x} \leq h_0$  it follows that  $\mathbf{h}\mathbf{z} \leq h_0 + \varepsilon \|\mathbf{h}\|_\infty$  for all  $\mathbf{z} \in P_\varepsilon^1$ . On the other hand, suppose that  $\mathbf{h}\mathbf{z} \leq h_0$  for all  $\mathbf{z} \in P_\varepsilon^1$  but there exists  $\mathbf{x} \in P$  such that  $\mathbf{h}\mathbf{x} > h_0 - \varepsilon \|\mathbf{h}\|_\infty$ . Let  $\mathbf{z}$  be such that  $z_i = x_i$  for all  $i \neq k$  and  $z_k = x_k + \varepsilon$  if  $h_k \geq 0$ ,  $z_k = x_k - \varepsilon$  if  $h_k < 0$ , where  $1 \leq k \leq n$  is such that  $|h_k| = \|\mathbf{h}\|_\infty$  and  $1 \leq i \leq n$ . Thus if  $h_k \geq 0$  we have

$$\mathbf{h}\mathbf{z} = \mathbf{h}\mathbf{x} + h_k\varepsilon = \mathbf{h}\mathbf{x} + \varepsilon \|\mathbf{h}\|_\infty > h_0 - \varepsilon \|\mathbf{h}\|_\infty + \varepsilon \|\mathbf{h}\|_\infty = h_0$$

and if  $h_k < 0$  that

$$\mathbf{h}\mathbf{z} = \mathbf{h}\mathbf{x} - \varepsilon h_k = \mathbf{h}\mathbf{x} + \varepsilon \|\mathbf{h}\|_\infty > h_0 - \varepsilon \|\mathbf{h}\|_\infty + \varepsilon \|\mathbf{h}\|_\infty = h_0 ,$$

i.e.,  $\mathbf{h}z > h_0$  for some  $z \in P$  which is a contradiction.

**(vii)** Suppose that  $z = \mathbf{x} \pm \varepsilon \mathbf{u}^i$  is an extreme point of  $P_\varepsilon^1$  but  $\mathbf{x} \in P$  is not an extreme point of  $P$ . Then there exist  $0 < \mu < 1$  and  $\mathbf{x}^1 \neq \mathbf{x} \neq \mathbf{x}^2 \neq \mathbf{x}^1$  such that  $\mathbf{x} = \mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2$ . But then we have

$$\begin{aligned} z &= \mathbf{x} \pm \varepsilon \mathbf{u}^i = \mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2 \pm \varepsilon \mathbf{u}^i = \mu\mathbf{x}^1 + (1 - \mu)\mathbf{x}^2 \pm \mu\varepsilon \mathbf{u}^i \pm (1 - \mu)\varepsilon \mathbf{u}^i \\ &= \mu(\mathbf{x}^1 \pm \varepsilon \mathbf{u}^i) + (1 - \mu)(\mathbf{x}^2 \pm \varepsilon \mathbf{u}^i) = \mu z^1 + (1 - \mu)z^2 \end{aligned}$$

where  $z^k = \mathbf{x}^k \pm \varepsilon \mathbf{u}^i$  for  $k = 1, 2$ . From the definition of  $P_\varepsilon^1$  it follows that  $z^1, z^2 \in P_\varepsilon^1$  and since  $z^1 \neq z^2$ , because  $\mathbf{x}^1 \neq \mathbf{x}^2$ , we have that  $z$  is the convex combination of two points in  $P_\varepsilon^1$  which contradicts the assumption that  $z$  is an extreme point.

For each extreme point  $\mathbf{x} \in P$  there are  $2n$  distinct points of the form  $z = \mathbf{x} \pm \varepsilon \mathbf{u}^i$ . Thus if  $P$  has  $p$  extreme points,  $P_\varepsilon^1$  has at most  $2np$  extreme points.

**(ix)** From the previous part we have that the extreme points of  $P_\varepsilon^1$  are of the form  $z = \mathbf{x} \pm \varepsilon \mathbf{u}^i$  where  $\mathbf{x}$  is an extreme point of  $P$ . Since the vertex complexity of  $P$  is  $\langle \mathbf{x} \rangle \leq \nu$ , we compute

$$\langle z \rangle = \langle \mathbf{x} \pm \varepsilon \mathbf{u}^i \rangle = \left\langle \sum_{\substack{j=1 \\ j \neq i}}^n x_j \right\rangle + \langle x_i \pm \varepsilon \rangle \leq 2 \sum_{j=1}^n \langle x_j \rangle + 2\langle \varepsilon \rangle = 2\langle \mathbf{x} \rangle + 2\langle \varepsilon \rangle \leq 2\nu + 2\langle \varepsilon \rangle .$$

For the facet complexity we have that for any valid inequality  $\mathbf{h}z \leq h_0 + \varepsilon \|\mathbf{h}\|_\infty$  we have  $\langle \mathbf{h} \rangle + \langle h_0 \rangle \leq \phi$  and thus we compute

$$\begin{aligned} \langle \mathbf{h} \rangle + \langle h_0 + \varepsilon \|\mathbf{h}\|_\infty \rangle &\leq \langle \mathbf{h} \rangle + 2\langle h_0 \rangle + 2\langle \varepsilon \rangle + 2\langle \|\mathbf{h}\|_\infty \rangle \leq 2(\langle \mathbf{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle + \langle \|\mathbf{h}\|_\infty \rangle \\ &\leq 2(\langle \mathbf{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle + \langle \mathbf{h} \rangle \leq 3(\langle \mathbf{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle \leq 3\phi + 2\langle \varepsilon \rangle . \end{aligned}$$

**(ix)** Since  $P_\varepsilon^1 \supseteq \mathbf{x} + \{z \in \mathbb{R}^n : \|z\|_1 \leq 1\}$  and  $P_\varepsilon^1 \supseteq B(\mathbf{x}, \varepsilon/n)$  we have like in (9.55)

$$vol(P_\varepsilon^1) \geq vol(B^1) > vol(B(\mathbf{x}, \varepsilon/n)) = \frac{\pi^{n/2} \varepsilon^n}{\Gamma(1 + n/2)n^n} .$$

Thus, we have to show that the volume of the sphere in  $\ell_1$ -norm,  $B^\varepsilon$  is given by  $2^n \varepsilon^n / n!$ . To this end, it suffices to show that  $V_n = vol(B^1) = 2^n / n!$ . We have

$$\begin{aligned} V_n &= \int_{-1}^1 V_{n-1}(1 - |x_n|)dx_n = \int_{-1}^1 f_{n-1}(1 - |x_n|)^{n-1}dx_n = f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \int_{-1}^1 |x_n|^j dx_n \\ &= f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \frac{2}{j+1} = 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \frac{(n-1)!}{j!(n-1-j)!} \frac{1}{j+1} \\ &= 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \frac{n!}{(j+1)!(n-1-j)!} \frac{1}{n} = 2f_{n-1} \sum_{j=0}^{n-1} (-1)^j \binom{n}{j+1} \frac{1}{n} = 2 \frac{f_{n-1}}{n} \sum_{\ell=1}^n (-1)^{\ell-1} \binom{n}{\ell} \\ &= 2 \frac{f_{n-1}}{n} (-1) \sum_{\ell=1}^n (-1)^\ell \binom{n}{\ell} = -2 \frac{f_{n-1}}{n} \left( \sum_{\ell=0}^n (-1)^\ell \binom{n}{\ell} - (-1)^0 \binom{n}{0} \right) \\ &= -2 \frac{f_{n-1}}{n} ((1-1)^n - 1) = 2 \frac{f_{n-1}}{n} . \end{aligned}$$

So we have  $f_n = \frac{2}{n} f_{n-1}$  and thus  $f_n = 2^n/n$ .

(x) The proof of Remark 9.13 goes through unchanged when the vectors  $d^1, d^2$  are the vectors of the  $\ell_\infty$ -norms (rather than the  $\ell_1$ -norms) of the corresponding rows of  $H_1, H_2$ . This follows because the estimation (7.17) applies as well to  $\|\mathbf{h}^i\|_\infty$  and thus the assertion follows.

---

### \*Exercise 9.8

Let  $\|\cdot\|_P$  be a polytopal norm on  $\mathbb{R}^n$ , i.e.,  $\|\cdot\|_P$  is a norm in  $\mathbb{R}^n$  and its “unit sphere”  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_P \leq 1\}$  is a polytope.

(i) Show that the “dual norm”  $\|\mathbf{y}\|_P^* = \max\{\mathbf{y}^T \mathbf{x} : \|\mathbf{x}\|_P \leq 1\}$  is a norm on  $\mathbb{R}^n$ .

(ii) Let  $\tilde{\mathbf{a}}^i \mathbf{x} \leq b_i$  for  $1 \leq i \leq m$  be any linear description of  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_P \leq 1\}$  and  $\mathbf{a}^i = \tilde{\mathbf{a}}^i / \|\tilde{\mathbf{a}}^i\|_P^*$  for  $1 \leq i \leq m$ . Show that  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^i \mathbf{x} \leq 1, 1 \leq i \leq m\}$ , that  $\mathbf{0} \in \text{relint } B_P$  and  $\dim B_P = n$ .

(iii) Show that  $\|\cdot\|_P^*$  is a polytopal norm on  $\mathbb{R}^n$ .

(iv) Prove Hölder’s inequality  $\mathbf{y}^T \mathbf{x} \leq \|\mathbf{y}\|_P^* \|\mathbf{x}\|_P$ .

(v) Let  $P_\epsilon^P$  be defined by  $P_\epsilon^P = \{z \in \mathbb{R}^n : \exists \mathbf{x} \in P \text{ such that } \|z - \mathbf{x}\|_P \leq \epsilon\}$ . Show that  $P_\epsilon^P$  is a polyhedron. Show that  $\mathbf{h} \mathbf{x} \leq h_0 + \epsilon \|\mathbf{h}\|_P^*$  for all  $\mathbf{x} \in P_\epsilon^P$  if  $\mathbf{h} \mathbf{x} \leq h_0$  for all  $\mathbf{x} \in P$ . Show that  $\mathbf{h} \mathbf{x} \leq h_0 - \epsilon \|\mathbf{h}\|_P^*$  for all  $\mathbf{x} \in P$  if  $\mathbf{h} \mathbf{x} \leq h_0$  for all  $\mathbf{x} \in P_\epsilon^P$ .

---

(i) For any  $\mathbf{y} \in \mathbb{R}^n$ , let  $\mathbf{y}' = \mathbf{y}/\|\mathbf{y}\|_P$ . Then  $\mathbf{y}' \in \mathbb{R}^n$ , by the homogeneity of  $\|\cdot\|_P$   $\|\mathbf{y}'\|_P \leq 1$  and  $\|\mathbf{y}'\|_P \leq 1$  and thus from the definition of  $\|\cdot\|_P$  we have that  $\|\mathbf{y}\|_P^* \geq \mathbf{y}^T \mathbf{y}' = \mathbf{y}^T \mathbf{y} / \|\mathbf{y}\|_P = \|\mathbf{y}\|^2 / \|\mathbf{y}\|_P$  where  $\|\mathbf{y}\|$  is the Euclidean norm of  $\mathbf{y}$ , i.e.  $\|\mathbf{y}\|_P^*$  is the ratio of two nonnegative numbers and thus it is nonnegative which is zero if and only if  $\|\mathbf{y}\| = 0$ , i.e. if and only if  $\mathbf{y} = 0$ . The homogeneity follows trivially since  $\|\alpha \mathbf{y}\|_P^* = \max\{\alpha \mathbf{y}^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\} = \alpha \max\{\mathbf{y}^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\} = \alpha \|\mathbf{y}\|_P^*$  for all  $\mathbf{y} \in \mathbb{R}^n$  and  $\alpha \geq 0$ . Finally we have  $\|\mathbf{y} + \mathbf{z}\|_P^* = \max\{(\mathbf{y} + \mathbf{z})^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\} \leq \max\{\mathbf{y}^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\} + \max\{\mathbf{z}^T \mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_P \leq 1\} = \|\mathbf{y}\|_P^* + \|\mathbf{z}\|_P^*$ , i.e. the triangle inequality holds and thus  $\|\cdot\|_P^*$  is a norm.

(ii) We are given that  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \tilde{\mathbf{a}}^i \mathbf{x} \leq b_i \text{ for } i = 1, \dots, m\}$ . From the definition (9.57) of the dual norm we have  $\|\mathbf{a}^i\|_P^* = \max\{\mathbf{a}^i \mathbf{x} : \mathbf{x} \in B_P\} \leq b_i$  and thus  $\tilde{\mathbf{a}}^i \mathbf{x} \leq \|\tilde{\mathbf{a}}^i\|_P^* \mathbf{x}$  for all  $\mathbf{x} \in B_P$  and all  $1 \leq i \leq m$ . Since  $\|\cdot\|_P^* \geq 0$  and  $\tilde{\mathbf{a}}^i \neq \mathbf{0}$  we have that  $\|\tilde{\mathbf{a}}^i\|_P^* > 0$  and thus dividing by  $\|\tilde{\mathbf{a}}^i\|_P^*$  we get  $\mathbf{a}^i \mathbf{x} \leq 1$  for all  $\mathbf{x} \in B_P$  and  $1 \leq i \leq m$ , i.e.  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^i \mathbf{x} \leq 1 \text{ for } 1 \leq i \leq m\}$ . Since  $\mathbf{a}^i \mathbf{0} < 1$  for  $1 \leq i \leq m$ ,  $\mathbf{0} \in \text{relint } B_P$ . Let  $\|\mathbf{a}\| = \max\{\|\mathbf{a}^i\| : 1 \leq i \leq m\}$  and  $\epsilon = 1/\|\mathbf{a}\|$ . By the Cauchy-Schwarz inequality,  $\mathbf{a}^i \mathbf{x} \leq \|\mathbf{a}^i\| \|\mathbf{x}\| \leq \|\mathbf{a}^i\| / \|\mathbf{a}\| \leq 1$  for  $1 \leq i \leq m$  and for all  $\mathbf{x} \in \mathbb{R}^n$  with  $\|\mathbf{x}\| \leq \epsilon$ . Consequently,  $B_P \supseteq B(\mathbf{0}, r = \epsilon)$ , i.e.,  $B_P$  contains a ball with center  $\mathbf{0} \in \mathbb{R}^n$  and radius  $r = \epsilon > 0$ , and thus  $\dim B_P = n$ .

(iii) Since  $\|\cdot\|_P$  is a polytopal norm the set  $B_P = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_P \leq 1\}$  is a polytope with  $\dim B_P = n$  and  $\mathbf{0} \in \text{relint } B_P$ . Let  $S_P = \{\mathbf{x}^1, \dots, \mathbf{x}^p\}$  be the set of the extreme points of  $B_P$  and  $A\mathbf{x} \leq \mathbf{b}$  be a linear description of  $B_P$ . Then we have that  $\|\mathbf{y}\|_P^* = \max\{\mathbf{y}^T \mathbf{x} : A\mathbf{x} \leq \mathbf{b}\} = \mathbf{y}^T \mathbf{x}'$  where  $\mathbf{x}' \in S_P$ .

Thus  $B_P^* = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_P^* \leq 1\} = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^T \mathbf{x}^i \leq 1 \text{ for all } 1 \leq i \leq p\}$ . Since  $\dim B_P = n$  and  $0 \in \text{relint } B_P$  it follows that there exists  $\boldsymbol{\mu} \in \mathbb{R}^p$ ,  $\boldsymbol{\mu} \geq 0$  such that  $\sum_{i=1}^p \mu_i \mathbf{x}^i = \pm \mathbf{u}^i$  where  $\mathbf{u}^i$  is the  $i$ -th unit vector in  $\mathbb{R}^n$  and  $1 \leq i \leq n$ . Consequently, by the duality theorem of linear programming  $0 \leq \max\{\mathbf{y}^T(\pm \mathbf{u}^i) : \mathbf{y}^T \mathbf{x}^i \leq 1 \text{ for } i = 1, \dots, p\} = \min\{\mathbf{e}^T \boldsymbol{\mu} : \sum_{i=1}^p \mu_i \mathbf{x}^i = \pm \mathbf{u}^i, \boldsymbol{\mu} \geq 0\} < \infty$  for  $1 \leq i \leq n$  and thus  $B_P^*$  is a bounded set of  $\mathbb{R}^n$ , i.e., the “unit sphere” in the  $\|\cdot\|_P^*$ -norm is a polytope.

**(iv)** The inequality holds trivially as equality if  $\mathbf{x} = 0$ . From the homogeneity of  $\|\cdot\|_P$  we have that for every  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} \neq 0$ ,  $\mathbf{x}' = \mathbf{x}/\|\mathbf{x}\|_P$  satisfies  $\mathbf{x}' \in \mathbb{R}^n$  and  $\|\mathbf{x}'\|_P \leq 1$ . Thus from the definition (9.57) of the dual norm it follows that  $\mathbf{y}^T \mathbf{x}' \leq \|\mathbf{y}\|_P^*$ , i.e.  $\mathbf{y}^T \mathbf{x}/\|\mathbf{x}\|_P \leq \|\mathbf{y}\|_P^*$  and since  $\|\mathbf{x}\|_P > 0$  we get  $\mathbf{y}^T \mathbf{x} \leq \|\mathbf{y}\|_P^* \|\mathbf{x}\|_P$  for all  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{y} \in \mathbb{R}^n$ .

**(v)** Let  $P_\epsilon^P = \{z \in \mathbb{R}^n : \exists \mathbf{x} \in P \text{ such that } \|z - \mathbf{x}\|_P \leq \epsilon\}$ . Since  $\|\cdot\|_P$  is a polytopal norm there exists a linear description of the set  $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_P \leq \epsilon\}$  and thus, like in the proof of Remark 9.12,  $P_\epsilon^P$  is the image of a polyhedron and thus a polyhedron itself. Suppose that  $\mathbf{h}\mathbf{x} \leq h_0$  for all  $\mathbf{x} \in P$  and let  $z \in P_\epsilon^P$ . Then there exists  $\mathbf{x} \in P$  such that  $\|z - \mathbf{x}\|_P \leq \epsilon$ . From Hölder's inequality we get  $\mathbf{h}(z - \mathbf{x}) \leq \|\mathbf{h}\|_P^* \|z - \mathbf{x}\|_P \leq \epsilon \|\mathbf{h}\|_P^*$  and thus  $\mathbf{h}z \leq \mathbf{h}\mathbf{x} + \epsilon \|\mathbf{h}\|_P^* \leq h_0 + \epsilon \|\mathbf{h}\|_P^*$  for all  $z \in P_\epsilon^P$ . Suppose now that  $\mathbf{h}z \leq h_0$  for all  $z \in P_\epsilon^P$  and let  $\mathbf{x}^* \in \mathbb{R}^n$  be such that  $\|\mathbf{h}\|_P^* = \max\{\mathbf{h}\mathbf{x} : \|\mathbf{x}\|_P \leq 1\} = \mathbf{h}\mathbf{x}^*$ . Assume that there exists  $\mathbf{y} \in P$  such that  $\mathbf{h}\mathbf{y} > h_0 - \epsilon \|\mathbf{h}\|_P^*$ . Let  $z = \mathbf{y} + \epsilon \mathbf{x}^*$  and thus  $\|z - \mathbf{y}\|_P = \epsilon \|\mathbf{x}^*\|_P \leq \epsilon$ , i.e.,  $z \in P_\epsilon^P$ . But then  $\mathbf{h}z = \mathbf{h}\mathbf{y} + \epsilon \mathbf{h}\mathbf{x}^* > h_0 - \epsilon \|\mathbf{h}\|_P^* + \epsilon \|\mathbf{h}\|_P^* = h_0$  is a contradiction.

---

### \*Exercise 9.9

Let  $0 < \Theta < 1$ .

- (i) Show that  $\Theta \leq \lfloor \Theta^{-1} \rfloor^{-1} < 2\Theta$  and  $\Theta \lfloor \Theta^{-1} \rfloor > 1 - \Theta$ .
  - (ii) Show that  $\lfloor \Theta^{-1} \rfloor^{-1}$  is a best approximation to  $\Theta$  for all  $1 \leq q < \lfloor \Theta^{-1} \rfloor$ .
  - (iii) Suppose  $1/2 < \Theta < 1$ . Show that  $r/s$  where  $r = \lfloor \frac{\Theta}{1-\Theta} \rfloor$  and  $s = \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1$  is a best approximation to  $\Theta$  for all  $1 \leq q < s$ .
  - (iv) Show that  $q_n \geq 2^{(n-1)/2}$  for  $n \geq 2$  for the integers  $q_n$  generated by the inductive process.
  - (v) Suppose  $0 < \Theta \neq \Theta' < 1$  and that the integers  $p_n, q_n$  and  $p'_n, q'_n$  generated by the respective inductive processes are such that  $p_n = p'_n$  and  $q_n = q'_n$  for  $1 \leq n \leq N$ , say. Show that  $|\Theta - \Theta'| \leq 2^{-N+1}$ .
- 

**(i)** From the definition of the lower integer part of a number we have  $\lfloor \Theta^{-1} \rfloor \leq \Theta^{-1}$  and since  $0 < \Theta < 1$ , we get  $\Theta \leq \lfloor \Theta^{-1} \rfloor^{-1}$  and the left part of the first inequality follows. For the right part we have:  $\Theta^{-1} < \lfloor \Theta^{-1} \rfloor + 1 \leq \lfloor \Theta^{-1} \rfloor + \lfloor \Theta^{-1} \rfloor = 2\lfloor \Theta^{-1} \rfloor$  and thus  $\lfloor \Theta^{-1} \rfloor^{-1} < 2\Theta$ , where we have used the definition of the lower integer part of a number in the first inequality and the inequality  $\lfloor \Theta^{-1} \rfloor \geq 1$  in the second. To prove that  $\Theta \lfloor \Theta^{-1} \rfloor > 1 - \Theta$  we have from the definition of the lower integer part of a number that  $\Theta^{-1} < \lfloor \Theta^{-1} \rfloor + 1$ . Multiplying both sides by  $\Theta > 0$  we get  $1 < \Theta \lfloor \Theta^{-1} \rfloor + \Theta$  and thus  $\Theta \lfloor \Theta^{-1} \rfloor > 1 - \Theta$ .

**(ii)** Applying the definition (9.60) of the best approximation with  $p = 1$  and  $D = \lfloor \Theta^{-1} \rfloor$  we have to show that (a)  $[\Theta \lfloor \Theta^{-1} \rfloor] = |\Theta \lfloor \Theta^{-1} \rfloor - 1|$  and (b)  $[q\Theta] > [\Theta \lfloor \Theta^{-1} \rfloor]$  for all  $1 \leq q < \lfloor \Theta^{-1} \rfloor$ . From the

first inequality of part (i) we have that  $\frac{1}{2} < \Theta[\Theta^{-1}] \leq 1$  and thus  $[\Theta[\Theta^{-1}]] = [\Theta[\Theta^{-1}]] - \Theta[\Theta^{-1}] < \Theta[\Theta^{-1}] - [\Theta[\Theta^{-1}]]$ , i.e.  $[\Theta[\Theta^{-1}]] = 1 - \Theta[\Theta^{-1}]$  and (a) follows. To prove (b) we have to show that  $[q\Theta] > 1 - \Theta[\Theta^{-1}]$  for all  $1 \leq q < [\Theta^{-1}]$ . First we note that  $q\Theta < [\Theta^{-1}]\Theta$  for all  $1 \leq q < [\Theta^{-1}]$  since  $\Theta > 0$ . Thus, in particular,  $0 < q\Theta < 1$  and  $[q\Theta] = 0$ ,  $[q\Theta] = 1$ . If  $[q\Theta] = 1 - q\Theta$  then we have  $[q\Theta] = 1 - q\Theta > 1 - [\Theta^{-1}]\Theta$ . If  $[q\Theta] = q\Theta$ , we have  $[q\Theta] = q\Theta \geq \Theta > 1 - \Theta[\Theta^{-1}]$  where we have used the second inequality of part (i) and thus (b) follows, and the proof of (ii) is complete.

**(iii)** To show that  $\frac{r}{s}$  with  $r = \lfloor \frac{\Theta}{1-\Theta} \rfloor$ ,  $s = \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1$  is a best approximation to  $\Theta$  for all  $1 \leq q < s$  if  $1/2 < \Theta < 1$  we have to prove

(a)  $[s\Theta] = |s\Theta - r|$  and (b)  $[q\Theta] > [s\Theta]$  for all integer  $q$  with  $1 \leq q < s$ .

We observe first, from  $\frac{\Theta}{1-\Theta} \geq \lfloor \frac{\Theta}{1-\Theta} \rfloor$  and  $1 - \Theta > 0$ , that  $s\Theta \geq \lfloor \frac{\Theta}{1-\Theta} \rfloor$  and thus  $|s\Theta| \geq \lfloor \frac{\Theta}{1-\Theta} \rfloor$ . On the other hand,  $s\Theta < s = r + 1$  since  $\Theta < 1$  and thus  $[s\Theta] = r$ . To prove (a) we have to show that  $\min\{s\Theta - [s\Theta], [s\Theta] - s\Theta\} = s\Theta - [s\Theta]$ . This is equivalent to  $s\Theta - [s\Theta] \leq [s\Theta] - s\Theta$ , which is true if  $s\Theta$  is integer, and so we can assume that  $s\Theta$  is not integer. Thus we need to show  $s\Theta \leq [s\Theta] + 1/2$ , i.e.,  $\left(1 + \lfloor \frac{\Theta}{1-\Theta} \rfloor\right)\Theta \leq \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1/2$ . Let  $x = \frac{\Theta}{1-\Theta}$ . Then  $\Theta = \frac{x}{1+x}$  and the assertion reads  $(1 + [x])\frac{x}{1+x} \leq [x] + 1/2$  or equivalently,  $x \leq 2[x] + 1$  which is trivially true for all  $x \geq 0$ . Consequently, part (a) follows.

To prove part (b), we first prove

(b1)  $q\Theta - [q\Theta] > s\Theta - r$  for all integer  $q$  with  $1 \leq q \leq r$ .

We claim  $q\Theta - [q\Theta] \geq (q+1)\Theta - [(q+1)\Theta]$  or equivalently,  $[(q+1)\Theta] \geq [q\Theta] + \Theta$  for all integer  $q \in [1, r]$ . Since  $q \leq \lfloor \frac{\Theta}{1-\Theta} \rfloor$  we have  $q \leq \frac{\Theta}{1-\Theta}$  and thus  $q \leq (q+1)\Theta$ . Since  $q$  is integer we get  $q \leq \lfloor (q+1)\Theta \rfloor$  and thus  $[(q+1)\Theta] \geq q\Theta\Theta^{-1} \geq [q\Theta]\Theta^{-1} > [q\Theta]$  since  $0 < \Theta < 1$  if  $[q\Theta] > 0$ . If  $[q\Theta] = 0$  then  $q = 1$  and the assertion is true as well because  $\Theta > 1/2$ . Consequently, the claim follows and thus, it suffices to prove (b1) for  $q = r$ , i.e.,  $r\Theta - [r\Theta] > (r+1)\Theta - r$  because  $s = r+1$ . But  $r > \Theta + [r\Theta]$  is trivially true since  $r \geq 1$  is integer and  $0 < \Theta < 1$ . Hence (b1) follows. We are left with proving

(b2)  $[q\Theta] - q\Theta > s\Theta - r$  for all integer  $q$  with  $1 \leq q \leq r$ .

Suppose first that  $\lfloor \frac{\Theta}{1-\Theta} \rfloor = \frac{\Theta}{1-\Theta}$ . Then  $s\Theta - r = 0$  and assume (b2) is wrong. Let  $q \in [1, r]$  be the smallest integer with  $[q\Theta] - q\Theta = 0$ . It follows that  $\Theta = p/q$  where  $1 \leq p = [q\Theta] < q$ . But  $\Theta = \frac{1}{r+1}$  and thus  $p(r+1) = rq$ . Hence we get  $r = \frac{p}{q-p}$  and  $r+1 = \frac{q}{q-p}$ , i.e., both  $p$  and  $q$  are divisible by the integer  $q-p \geq 1$ . If  $q-p = 1$ , then  $p=r$  and  $q=r+1$ . Otherwise,  $q-p \geq 2$  contradicts our assumption that  $q$  is the smallest integer with the required property. Consequently, such  $q$  does not exist in the range  $1, \dots, r$  and (b2) follows. Now suppose  $\frac{\Theta}{1-\Theta}$  is not integer. Like in the proof of part (b1) we conclude  $q \leq \lfloor (q+1)\Theta \rfloor$ . If  $(q+1)\Theta$  is integer, then from  $0 < \Theta < 1$ , we get  $q = (q+1)\Theta$ , i.e.,  $q = \frac{\Theta}{1-\Theta}$ , which contradicts the assumption that  $\frac{\Theta}{1-\Theta}$  is not integer. Consequently,  $(q+1)\Theta$  is not integer. Thus  $[q\Theta] \leq [\Theta[(q+1)\Theta]] \leq \lfloor (q+1)\Theta \rfloor < \lceil (q+1)\Theta \rceil$  and hence from  $0 < \Theta < 1$ ,  $\lceil (q+1)\Theta \rceil - (q+1)\Theta \geq [q\Theta] - q\Theta$  for all integer  $q \in [1, r]$ . Consequently, it suffices to prove (b2) for  $q = 1$ , i.e.,  $1 - \Theta > \left(1 + \lfloor \frac{\Theta}{1-\Theta} \rfloor\right)\Theta - \lfloor \frac{\Theta}{1-\Theta} \rfloor$ . Using  $x = \frac{\Theta}{1-\Theta}$  the assertion is equivalent to  $1 + [x] > x$  which is trivially true because  $x$  is a positive noninteger. Thus (b2) follows in this case as well, and hence the proof of part (iii) is complete.

**(iv)** We prove the assertion by induction. From the construction of the sequence of integers  $q_i$  we have  $q_1 = 1 < q_2 < \dots$ . Thus since  $q_2 > 1$  and integer,  $q_2 \geq 2 > 2^{1/2}$  and the assertion is

true for  $n = 2$ . Suppose that for  $n = k \geq 2$  we have  $q_k \geq 2^{(k-1)/2}$ . By (9.70)  $q_{k+1} = a_k q_k + q_{k-1} \geq q_k + q_{k-1} \geq 2^{(k-1)/2} + 2^{(k-2)/2} = 2^{(k-2)/2}(2^{1/2} + 1) > 2^{(k-2)/2}2 = 2^{k/2}$  and thus the assertion follows for  $n = k + 1$  and the inductive proof is complete. (Note that here we use the inductive process *without* the particular initialization (9.73), which “shifts” the index  $n$  of the inductive process by 1 if  $1 > \Theta > 1/2$ .)

(v) Assume WROG that  $0 < \Theta < \Theta' < 1$  and that the inductive process carries out at least  $N \geq 1$  iterations. By (9.65), i.e., because the signs of  $q_n\Theta - p_n$  and  $q'_n\Theta' - p'_n$  alternate, we have either  $\Theta' \leq p_N/q_N$  or  $p_N/q_N \leq \Theta$ . Since  $\Theta' \neq \Theta$ , the process continues for at least one more iteration for either  $\Theta$  or  $\Theta'$  or both. Suppose that  $\Theta' \leq p_N/q_N$ . Then  $p_{N+1}/q_{N+1} \leq \Theta$  and  $0 < \Theta' - \Theta \leq \frac{p_N}{q_N} - \frac{p_{N+1}}{q_{N+1}} = \frac{1}{q_N q_{N+1}} \leq 2^{-N+\frac{1}{2}}$ , by part (iv). Suppose that  $p_N/q_N \leq \Theta$ . Then  $p'_{N+1}/q'_{N+1} \geq \Theta'$  and  $0 < \Theta' - \Theta \leq \frac{p'_{N+1}}{q'_{N+1}} - \frac{p'_N}{q'_N} = \frac{1}{q'_{N+1} q'_N} \leq 2^{-N+\frac{1}{2}}$  by part (iv) as well because  $p'_N = p_N$  and  $q'_N = q_N$  by assumption. Thus  $|\Theta - \Theta'| \leq 2^{-N+\frac{1}{2}} \leq 2^{-N+1}$  as we have asserted.

---

### \*Exercise 9.10

Let  $P \subseteq \mathbb{R}^n$  be a rational polyhedron of facet complexity  $\phi$ , let  $P_u = P \cap \{\mathbf{x} \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n\}$  for some integer  $u \geq 1$  and let  $\mathbf{c} \in \mathbb{R}^n$  be any rational vector.

- (i) Every extreme point  $\mathbf{x} \in P_u$  is a rational vector with components  $x_j = p_j/q_j$  with integers  $0 \leq |p_j| < u2^{6n\phi+1}$  and  $1 \leq q_j < 2^{6n\phi}$  for  $1 \leq j \leq n$ .
  - (ii) Any two extreme points  $\mathbf{x}, \mathbf{y} \in P_u$  with  $\mathbf{c}\mathbf{x} > \mathbf{c}\mathbf{y}$  satisfy  $\mathbf{c}\mathbf{x} > \mathbf{c}\mathbf{y} + 2^{-12n^2\phi-\langle \mathbf{c} \rangle}$ .
  - (iii) For  $\Delta \geq 1 + u2^{6n\phi+12n^2\phi+\langle \mathbf{c} \rangle+1}$  let  $\tilde{d}_j = \Delta^n c_j + \Delta^{n-j}$  for  $1 \leq j \leq n$  and  $\tilde{\mathbf{d}} = (\tilde{d}_1, \dots, \tilde{d}_n)$ . Then the linear optimization problem  $\max\{\tilde{\mathbf{d}}\mathbf{x} : \mathbf{x} \in P_u\}$  has a unique maximizer  $\mathbf{x}^{max} \in P_u$  and  $\mathbf{c}\mathbf{x}^{max} = \max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in P_u\}$ .
  - (iv) Define  $\mathbf{d} = \tilde{\mathbf{d}}/\|\tilde{\mathbf{d}}\|_\infty$  where  $\tilde{\mathbf{d}}$  is defined in part (iii). Then  $\langle \mathbf{d} \rangle \leq 3.5n(n-1)\lceil \log_2 \Delta \rceil + 2(n-1)\langle \mathbf{c} \rangle$  and thus for  $u = 2^{\Lambda+1}$  and the smallest  $\Delta$  satisfying the condition of part (iii) we have  $\langle \mathbf{d} \rangle \leq 3.5n(n-1)\phi(16n^2 + 11n\phi + 1) + (3.5n + 2)(n-1)\langle \mathbf{c} \rangle + 14n(n-1)$ , where  $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$ .
  - (v) Let  $(\mathbf{h}, h_0)$  belong to a linear description of  $P$ ,  $\|\mathbf{h}\|_\infty > 0$  and  $\langle \mathbf{h} \rangle + \langle h_0 \rangle \leq \phi$ . Show that  $\langle \tilde{\mathbf{h}} \rangle + \langle \tilde{h}_0 \rangle \leq n\phi + 2$  where  $\tilde{\mathbf{h}} = \mathbf{h}/\|\mathbf{h}\|_\infty$  and  $\tilde{h}_0 = h_0/\|\mathbf{h}\|_\infty$ .
- 

- (i) If  $\mathbf{x} \in P_u$  is an extreme point of  $P$ , then the assertion follows from point 7.5(b). So suppose that  $\mathbf{x} \in P_u$  is an extreme point of  $P_u$  that is not an extreme point of  $P$ . Then by point 7.2(b)  $\mathbf{x}$  is determined uniquely by a system of equations

$$\begin{pmatrix} \mathbf{I}_k & \mathbf{O} \\ \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \end{pmatrix} = \begin{pmatrix} u\mathbf{g}^* \\ \mathbf{f}^* \end{pmatrix}$$

where  $\mathbf{I}_k$  is the  $k \times k$  identity matrix with  $1 \leq k \leq n$ ,  $\mathbf{F}_1, \mathbf{F}_2$  are  $(n-k) \times k$  and  $(n-k) \times (n-k)$  matrices,  $\mathbf{g}^*$  is a vector with entries +1 if  $x_j = u$ , -1 if  $x_j = -u$  and  $\mathbf{f}^*$  has  $n-k$  components.

Moreover, every row  $(\mathbf{f}^i, f_i)$  of  $(\mathbf{F}_1 \ \mathbf{F}_2 \ \mathbf{f}^*)$  satisfies  $\langle \mathbf{f}^i \rangle + \langle f_i \rangle \leq \phi$  and if  $k < n$  then  $\det \mathbf{F}_2 \neq 0$ . If  $k = n$  then  $\mathbf{F}_2$  is empty and we define  $\det \mathbf{F}_2 = 1$ . Denote by  $\mathbf{G}$  the  $n \times n$  matrix of this equation system and suppose the components of  $\mathbf{x}$  are indexed to agree with the above partitioning into  $\mathbf{x}^1$  and  $\mathbf{x}^2$ . By  $\mathbf{g}_j$  we denote the  $j$ -th column of  $(\mathbf{I}_k \ \mathbf{O})$ , by  $\mathbf{f}_j$  the  $j$ -th column of  $(\mathbf{F}_1 \ \mathbf{F}_2)$  and by  $\mathbf{u}_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ . If we let

$$\mathbf{G}_j = \mathbf{G} + \mathbf{u}_j^T \left( \begin{pmatrix} u\mathbf{g}^* \\ \mathbf{f}^* \end{pmatrix} - \begin{pmatrix} \mathbf{g}_j \\ \mathbf{f}_j \end{pmatrix} \right),$$

then by Cramer's rule  $x_j = \det \mathbf{G}_j / \det \mathbf{G}$  and we need to estimate the digital sizes of the determinants. From formula (7.18) we get  $\langle \det \mathbf{G} \rangle \leq 2\langle \mathbf{G} \rangle - n^2 \leq 2n\phi - n^2$ . Moreover,  $\det \mathbf{G}$  is a rational number of digital size less than  $2n\phi$  and thus there exist integers  $p, q$  with  $0 \leq |p| < 2^{2n\phi}$ ,  $1 \leq q_j < 2^{2n\phi}$  such that  $\det \mathbf{G} = p/q$ . Suppose that  $1 \leq j \leq k$ . We calculate  $\det \mathbf{G}_j = \pm u \det \mathbf{F}_2$ . Moreover, by the same reasoning as before  $\langle \det \mathbf{F}_2 \rangle < 2n\phi$  is correct and thus there exist integers  $p_j, q_j$  with  $0 \leq |p_j| < 2^{n\phi}$ ,  $1 \leq q_j < 2^{2n\phi}$  such that  $\det \mathbf{F}_2 = p_j/q_j$ . It follows that  $x_j = \pm u p_j q_j / q_j p$  satisfies  $u p_j q$  integer,  $0 \leq |u p_j q| < u 2^{4n\phi} < u 2^{6n\phi+1}$  and  $1 \leq |q_j p| < 2^{4n\phi} < 2^{6n\phi}$  since  $u$  is integer and thus the assertion follows in this case. Suppose now that  $k+1 \leq j \leq n$ . From the formula for the determinant of a partitioned matrix of Chapter 2.2 we calculate

$$\begin{aligned} \det \mathbf{G}_j &= \det(\mathbf{F}_2 + \mathbf{v}_j^T (\mathbf{f}^* - \mathbf{f}_j) - u \mathbf{F}_1 \mathbf{v}_j^T \mathbf{g}^*) \\ &= (\det \mathbf{F}_2)(\det(\mathbf{I}_k + \mathbf{F}_2^{-1} \mathbf{v}_j^T (\mathbf{f}^* - \mathbf{f}_j) - u \mathbf{F}_2^{-1} \mathbf{F}_1 \mathbf{v}_j^T \mathbf{g}^*)) \\ &= (\det \mathbf{F}_2)(f_* - ug_*), \end{aligned}$$

where  $\mathbf{v}_j \in \mathbb{R}^{n-k}$  is the  $j$ -th unit vector,  $f_* = \mathbf{v}_j^T \mathbf{F}_2^{-1} \mathbf{f}^*$  and  $g_* = \mathbf{v}_j^T \mathbf{F}_2^{-1} \mathbf{F}_1 \mathbf{g}^*$ . We calculate

$$f_* \det \mathbf{F}_2 = \det(\mathbf{F}_2 + \mathbf{v}_j^T (\mathbf{f}^* - \mathbf{f}_j))$$

by factoring out  $\mathbf{F}_2$  and thus by (7.18)  $\langle f_* \det \mathbf{F}_2 \rangle < 2n\phi$  is correct, i.e., there exist integer  $p_j, q_j$  with  $0 \leq |p_j| < 2^{2n\phi}$ ,  $1 \leq q_j < 2^{2n\phi}$  such that  $f_* \det \mathbf{F}_2 = p_j/q_j$ . We calculate also

$$-g_* \det \mathbf{F}_2 = \det \left( \begin{pmatrix} \mathbf{I}_k & \mathbf{O} \\ \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} + \mathbf{u}_j^T \begin{pmatrix} \mathbf{g}^* \\ -\mathbf{f}_j \end{pmatrix} \right)$$

by applying the determinant formula for partitioned matrices first and then factoring out  $\mathbf{F}_2$ . Since  $\phi \geq n+1$  it follows that the matrix on the right is a rational number of digital size less than  $2n\phi$  and thus there exist integers  $r, s$  with  $0 \leq |r| < 2^{2n\phi}$  and  $1 \leq s < 2^{2n\phi}$  such that  $-g_* \det \mathbf{F}_2 = r/s$ . Consequently, since  $u \geq 1$  is integer  $x_j = q(s p_j + u + q_j) / p s q_j$  satisfies the assertion and the proof of (i) is complete.

**(ii)** By part (i) of this exercise we have  $x_j = p_j/q_j$  and  $y_j = r_j/s_j$  with integer numbers  $p_j, q_j, r_j, s_j$  satisfying  $0 \leq |p_j|, |r_j| < u 2^{6n\phi+1}$  and  $1 \leq q_j, s_j < 2^{6n\phi}$  for  $1 \leq j \leq n$ . Let  $c_j = a_j/b_j$  with integer  $a_j$  and  $b_j \geq 1$  for  $1 \leq j \leq n$  since  $c$  is rational. Since  $c\mathbf{x} > c\mathbf{y}$  it follows that  $c(\mathbf{x} - \mathbf{y}) \geq (\prod_{j=1}^n q_j s_j b_j)^{-1} > 2^{-12n^2\phi - \langle c \rangle}$  because  $\prod_{j=1}^n b_j < 2^{\langle c \rangle}$ .

**(iii)** The proof of this part of the exercise goes like the proof of Exercise 7.14(iii).

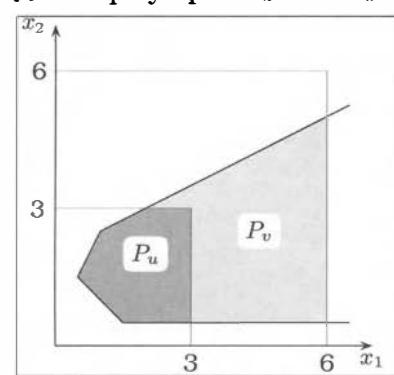
(iv) The proof follows from Exercise 7.14(iv) and by a simple substitution of the values for  $\Delta$ ,  $\Lambda$ ,  $u$  and the rough estimation  $1 + 2^{3+\alpha} < 2^{4+\alpha}$  for  $\alpha \geq 0$ .

(v) Let  $h_j = \frac{p_j}{q_j}$  with integer  $p_j, q_j$  satisfying  $0 \leq |p_j| < 2^\phi$ ,  $1 \leq q_j < 2^\phi$  and  $|\frac{p_j}{q_\ell}| \geq |\frac{p_i}{q_j}|$  for  $1 \leq j \leq n$ . Then  $\tilde{h}_j = \frac{p_j q_\ell}{q_j p_\ell}$ ,  $\tilde{h}_\ell = 1$ ,  $\langle \tilde{h}_j \rangle \leq \langle p_j \rangle + \langle q_j \rangle + \langle p_\ell \rangle + \langle q_\ell \rangle$  and  $\langle \tilde{h}_\ell \rangle = 2$  where  $0 \leq j \neq \ell \leq n$ . Consequently,  $\langle \tilde{h} \rangle + \langle \tilde{h}_0 \rangle \leq \langle h \rangle + \langle h_0 \rangle + (n-1)(\langle p_\ell \rangle + \langle q_\ell \rangle) + 2 \leq n\phi + 2$ .

### \*Exercise 9.11

- Let  $P = \{x \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 1, x_1 + x_2 \geq 2, -x_1 + 2x_2 \leq 4, 2x_2 \geq 0\}$  and  $P_u$  be as defined in (9.76). Find the maximizer  $x^{max}$  of  $\max\{x_1 + x_2 : x \in P_u\}$  for  $u = 3$  and the corresponding  $y^{max}$  for  $v = 2u$ . Does  $y^{max} - x^{max}$  belong to the asymptotic cone of  $P$ ? If not, what is the smallest possible value of  $u$  that works? What is the theoretical value that you get for  $u$  using  $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$ ?
- Suppose that the direction vector  $t \in P_\infty$  of the proof of Remark 9.19 satisfies  $ct = 0$  and let  $T_- = \{y \in T : cy = 0, \|y\|_\infty = 1\}$  where  $(S, T)$  is a minimal generator of the polyhedron  $P \subseteq \mathbb{R}^n$  such that  $\langle x_j \rangle \leq 4n\phi$  for all  $j$  and  $x \in S \cup T$ . Prove that  $t \succeq y$  for all  $y \in T_-$ , i.e. that  $t$  is lexicographically greater than or equal to every  $y \in T_-$ .
- Determine the facet and vertex complexity of the polytopes  $S_n$  and  $C_n$  of Exercise 7.2 and of  $H_n$  and  $O_n$  of Exercise 7.7.
- Find polynomial-time algorithms that solve the polyhedral separation problem over  $S_n$ ,  $C_n$ ,  $H_n$  and  $O_n$ .

(i) The polytopes  $P_3$  and  $P_6$  are shown in the figure. Maximizing the function  $x_1 + x_2$  over  $P_3$  we get  $x^{max} = (3, 3)$  while maximizing it over  $P_6$  we get  $y^{max} = (6, 5)$ . The difference vector  $y^{max} - x^{max} = (3, 2)$  and it is not in the asymptotic cone  $C_\infty$  of  $P$ , where  $C_\infty = \{y \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 0, x_1 + x_2 \geq 0, -x_1 + 2x_2 \leq 0, 2x_2 \geq 0\}$  since it violates the third inequality. Selecting  $u \geq 4$  we have the maximizer lying on the extreme ray  $-x_1 + 2x_2 = 4$  of the polyhedron  $P$  and thus  $u = 4$  and  $v = 2u = 8$  will give  $x^{max} = (4, 4)$  and  $y^{max} = (8, 6)$ , and thus  $y^{max} - x^{max} = (4, 2) \in C_\infty$ . To compute the theoretical value of  $u$ , we first calculate  $\phi$  for the polyhedron  $P$ . Since  $\langle 0 \rangle = 1$ ,  $\langle 1 \rangle = \langle -1 \rangle = 2$ ,  $\langle 2 \rangle = 3$  and  $\langle 4 \rangle = \langle -4 \rangle = 4$ , we have that  $\phi \geq \max\{\langle -4 \rangle + \langle 2 \rangle + \langle 1 \rangle, \langle 1 \rangle + \langle 1 \rangle + \langle 2 \rangle, \langle -1 \rangle + \langle 2 \rangle + \langle 4 \rangle, \langle 0 \rangle + \langle 2 \rangle + \langle 1 \rangle\} = 9$ . So selecting  $\phi = 9$  we get from  $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$  with  $n = 2$  that  $\Lambda = 244$  and thus  $u = 2^{244}$  which is a horribly big number and evidently much bigger than required in this case.



(ii) By assumption the unique maximizer  $x^{max}$  of  $\max x$  over the polytope  $P_u$  for  $u = 2^\Lambda$  satisfies  $x_j^{max} = \pm u$  for at least one  $j \in \{1, \dots, n\}$  and thus  $x^{max}$  is the unique solution to a system of

equations of the form

$$\begin{pmatrix} \pm I_k & \mathbf{O} \\ F_1 & F_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^{max} \\ \mathbf{x}_1^{max} \end{pmatrix} = \begin{pmatrix} ue_k \\ f^* \end{pmatrix}, \quad (1)$$

where  $\pm I_k$  is some  $k \times k$  matrix with  $1 \leq k \leq n$  having  $+1$  or  $-1$  on its main diagonal (according to  $x_j = u$  or  $x_j = -u$ ), zeros elsewhere,  $F_1, F_2$  are  $(n-k) \times k$  and  $(n-k) \times (n-k)$  matrices,  $e_k$  is a vector of  $k$  ones and  $f^*$  has  $n-k$  components. Every row  $(f^i, f_i)$  of  $(F \ F_2 \ f^*)$  satisfies  $\langle f^i \rangle + \langle f_i \rangle \leq \phi$  and  $\det F_2 \neq 0$ , where by convention  $\det F_2 = 1$  if  $F_2$  is empty. Moreover,  $y^{max}$  satisfies (1) with  $u$  replaced by  $v = 2^{\Lambda+1}$  and  $t = 2^{-\Lambda}(y^{max} - x^{max}) \in C_\infty$  satisfies  $\langle t_j \rangle \leq 4n\phi$  for  $1 \leq j \leq n$ . (For more detail than given in the proof of Remark 9.17 on the estimation of  $\langle t_j \rangle$  see the proof of Exercise 9.10(i).) From the uniqueness of the respective maximizers  $x^{max}, y^{max}$  and the assumptions that  $x_j = \pm u$  for at least one index  $j$  and  $\Lambda > 4n\phi$ , it follows that every basis defining  $x^{max}$  is of the form (1) and thus by the duality theory of linear programming

$$d = \lambda \begin{pmatrix} \pm I_k & \mathbf{O} \\ F_1 & F_2 \end{pmatrix} \text{ with } \lambda \geq 0 \quad (2)$$

for some such basis defining  $x^{max}$  and  $y^{max}$ , respectively. Consequently, a basis satisfying (1) and (2) exists. Since by construction  $t \in C_\infty$  and  $t \neq 0$ , it follows that  $\tilde{t} = t/\|t\|_\infty \in P_\infty$  is an extreme point of  $P_\infty$ . More precisely, by dropping some of the constraints of  $P_\infty$  we have

$$P_\infty \subseteq OC(\tilde{t}, \widetilde{H}) = \left\{ \mathbf{y} \in \mathbb{R}^n : \begin{pmatrix} \pm I_k & \mathbf{O} \\ F_1 & F_2 \end{pmatrix} \mathbf{y} \leq \begin{pmatrix} e_k \\ \mathbf{0} \end{pmatrix} \right\},$$

i.e.,  $OC(\tilde{t}, \widetilde{H})$  is the displaced outer cone with apex at  $\tilde{t}$  containing all of  $P_\infty$  (see the end of Chapter 7.5.4), and  $\tilde{t}$  is an extreme point of  $OC(\tilde{t}, \widetilde{H})$ . From (2) it follows that  $\tilde{t}$  maximizes  $d\mathbf{y}$  over  $OC(\tilde{t}, \widetilde{H})$  and thus by the outer inclusion principle,  $\tilde{t}$  maximizes  $d\mathbf{y}$  over  $P_\infty$ . The matrix  $\widetilde{H}$  defining  $P_\infty$  is given by  $H$ ,  $I_n$  and  $-I_n$  and thus  $P_\infty$  has a facet complexity of  $\phi$  – just like the polyhedron  $P$ . Hence  $\langle \tilde{t}_j \rangle \leq 4n\phi$  for  $1 \leq j \leq n$ . Since  $d\tilde{t} \geq d\mathbf{y}$  for all  $\mathbf{y} \in P_\infty$ , it follows like in Exercise 7.14(iii) that  $\tilde{t}$  is the unique maximizer of  $d$  over  $P_\infty$  because the number  $\Delta$  that we use to prove Remark 9.19 is greater than the number  $1 + 2^{4n\phi+8n^2\phi+\lfloor c \rfloor + 1}$  that suffices to guarantee uniqueness. But then by Exercise 7.14(v)  $\tilde{t} \succeq \mathbf{y}$  for all  $\mathbf{y} \in T_+$  and the proof of part (ii) is complete.

**(iii)** The digital size of the inequality  $-x_j \leq 0$  equals  $n+2$  since we also store zero coefficients. Likewise the digital size of  $x_j \leq 1$  is  $n+3$  and of  $\sum_{j=1}^n x_j \leq 1$  is  $2(n+1)$ . Consequently, the facet complexity of the polytope  $S_n$  is  $\phi = 2(n+1)$  and that of  $C_n$  is  $\phi = n+3$ , where  $n \geq 1$  is arbitrary. The polyhedron  $H_n$  of Exercise 7.7(i) has the same facet complexity as  $S_n$ , i.e.,  $\phi = 2(n+1)$ , while the polytope  $O_n$  of Exercise 7.7(ii) has  $\phi = 2n+1 + \lceil \log_2 n \rceil$ .

**(iv)** Since  $S_n$  has  $n+1$  constraints and  $C_n$  has  $2n$  constraints, the (trivial) algorithm LIST-and-CHECK is a polynomial-time separation algorithm for  $S_n$  and  $C_n$ , respectively, no matter what rational  $\mathbf{y} \in \mathbb{R}^n$  is given as input. (LIST-and-CHECK is just that; you list all inequalities and check them one by one for violation.) Since both  $H_n$  and  $O_n$  have exponentially many inequalities, LIST-and-CHECK does not work in either case since it may, in the worst case, require exponential time

in  $n$  to execute it. In the case of  $H_n$  every constraint is of the form  $\mathbf{h}\mathbf{x} = \sum_{j=1}^n \delta_j x_j \leq 1$  where  $\delta_j \in \{0, 1\}$  for  $1 \leq j \leq n$  and  $\|\mathbf{h}\|_\infty = 1$ , except for the trivial constraint  $\mathbf{0}\mathbf{x} \leq 1$  which is never violated. The polyhedral separation problem for  $H_n$  is

$$\max \left\{ \sum_{j=1}^n \delta_j z_j - 1 : \delta_j \in \{0, 1\} \text{ for } 1 \leq j \leq n \right\}$$

where  $\mathbf{z} \in \mathbb{R}^n$  is a rational vector. To solve the problem we scan the vector  $\mathbf{z}$  and set  $\delta_j = 1$  if  $z_j > 0$ ,  $\delta_j = 0$  otherwise. This separation algorithm is linear in  $n$  and  $\langle \mathbf{z} \rangle$  and thus a polynomial-time algorithm for the polyhedral separation problem for  $H_n$ .

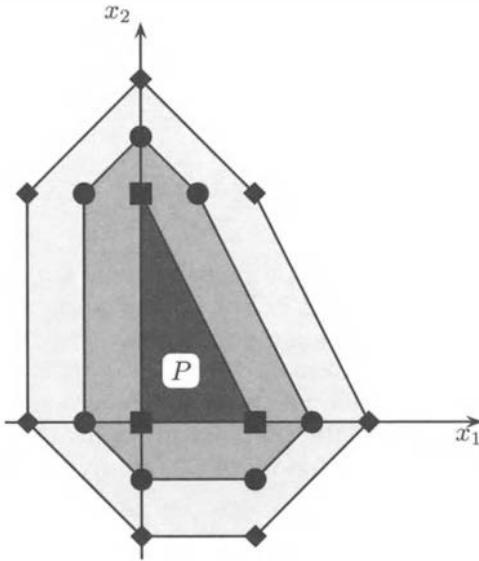
In the case of the polytope  $O_n$  we can check the  $2n$  constraints  $0 \leq x_j \leq 1$  for  $1 \leq j \leq n$  by the algorithm LIST-and-CHECK in polynomial time. So we can assume WROG that the rational vector  $\mathbf{z} \in \mathbb{R}^n$  for which we want to solve the polyhedral separation problem satisfies  $0 \leq z_j \leq 1$ . The separation problem for the remaining exponentially many constraints of  $O_n$  is

$$\begin{aligned} \max \{ & \sum_{j \in N_1} z_j - \sum_{j \in N-N-1} z_j - |N_1| + 1 : N_1 \subseteq N, |N_1| \text{ even} \} \\ &= 1 - \min \{ \sum_{j \in N_1} (1 - z_j) + \sum_{j \in N-N_1} z_j : N_1 \subseteq N, |N_1| \text{ even} \}, \end{aligned}$$

and a violated constraint is obtained if the objective function of the minimization problem is less than one. To solve the problem we order the components of  $\mathbf{z}$  in decreasing order which requires time that is polynomial in  $n$  and  $\langle \mathbf{z} \rangle$ . E.g. the sorting algorithm HEAPSORT requires  $\mathcal{O}(n \log n)$  operations in the worst case. So we can assume WROG that  $1 \geq z_1 \geq z_2 \geq \dots \geq z_k \geq 1/2 > z_{k+1} \geq \dots \geq z_n \geq 0$ , where  $0 \leq k \leq n$ . Finding the index  $k$  or verifying that  $k = 0$  can be done by scanning the ordered vector  $\mathbf{z}$  once, i.e., in time that is linear in  $n$  and  $\langle \mathbf{z} \rangle$ . If the index  $k$  is even we set  $N_1^* = \{1, \dots, k\}$ . If  $k$  is odd we set  $N_1^* = \{1, \dots, k-1\}$  if  $z_k + z_{k+1} < 1$ ,  $N_1^* = \{1, \dots, k+1\}$  otherwise. By construction  $|N_1^*|$  is even,  $z_{i_1} + z_{i_2} < 1$  for all  $i_1 \neq i_2 \notin N_1^*$  and  $z_{i_1} + z_{i_2} \geq 1$  for all  $i_1, i_2 \in N_1^*$  in all cases. We claim that  $N_1^*$  solves the minimization problem. Suppose not and let  $S \subseteq N$  be an optimal solution. Then  $|S|$  is even,  $S \neq N_1^*$  and

$$z(S) = \sum_{j \in S} (1 - z_j) + \sum_{j \in N-S} z_j < \sum_{j \in N_1^*} (1 - z_j) + \sum_{j \in N-N_1^*} z_j = z(N_1^*).$$

If  $|S| = |N_1^*|$  then by construction  $\sum_{j \in S} z_j \leq \sum_{j \in N_1^*} z_j$  and  $\sum_{j \in N-N_1^*} z_j \leq \sum_{j \in N-S} z_j$ . But then  $z(N_1^*) \leq z(S)$  and thus if  $z(S) < z(N_1^*)$  then  $|S| - |N_1^*|$  is an even number different from zero. Suppose first that  $|S| \geq 2 + |N_1^*|$ . Then there exists  $i_1 \neq i_2 \in S$  such that  $i_1, i_2 \notin N_1^*$ . Let  $S' = S - \{i_1, i_2\}$ . We compute  $z(S') = z(S) - 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$  by the optimality of  $S$  and thus  $z_{i_1} + z_{i_2} \geq 1$  which is a contradiction because  $i_1 \neq i_2 \notin N_1^*$ . Suppose now that  $|S| + 2 \leq |N_1^*|$ . Then there exists  $i_1 \neq i_2 \in N_1^*$  such that  $i_1 \notin S$ ,  $i_2 \notin S$ . Let  $S' = S \cup \{i_1, i_2\}$ . We compute  $z(S') = z(S) + 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$  by the optimality of  $S$  and thus  $z_{i_1} + z_{i_2} \leq 1$ . Since  $i_1, i_2 \in N_1^*$  we get  $z_{i_1} + z_{i_2} = 1$  and thus  $z(S') = z(S)$ . Consequently,  $S'$  is optimal as well as  $|S| < |S'| \leq |N_1^*|$ , we can reapply the reasoning and after finitely many steps we arrive at a contradiction because the cardinality of an optimal  $S$  is bounded by  $|N_1^*|$  in this case. Consequently, the claim follows. The polyhedral separation problem for  $O_n$  can thus be solved in  $\mathcal{O}(\langle \mathbf{z} \rangle n \log n)$  time for any rational  $\mathbf{z} \in \mathbb{R}^n$ .



**Fig. 9.13.**  $\varepsilon$ -solidifications of  $P$  for  $\varepsilon = 0, 1/2$  and  $1$

### \*Exercise 9.12

- Consider the polytope  $P = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \leq 2, x_1 \geq 0, x_2 \geq 0\}$ . Find minimal generators for the corresponding  $SP$ ,  $SP_\infty$  and  $SP_{\infty}^*$  as defined in (9.80), (9.81) and (9.82). Show that every nonzero extreme point  $(h, h_0)$  of  $SP_\infty$  defines a facet  $hx \leq h_0 + \varepsilon$  of the  $\varepsilon$ -solidification  $P_\varepsilon^1$  of  $P$  in the  $\ell_1$ -norm and vice versa, that every facet  $hx \leq h_0 + \varepsilon$  with  $\|h\|_\infty = 1$  of  $P_\varepsilon^1$  defines an extreme point  $(h, h_0)$  of  $SP_\infty$  where  $\varepsilon > 0$ .
- Do the same as in part (i) of this exercise for the polyhedron  $P = \{x \in \mathbb{R}^2 : 2x_1 - x_2 = 0, x_1 \geq 1\}$ .
- Do the same as in part (i) of this exercise for the polyhedron  $P = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \geq 5, x_1 - x_2 \geq -2, x_2 \geq 1\}$ . In addition, let  $(f, f_0) = (-1, -1, -6)$  and solve the linear program  $\max\{-x_1 - x_2 + 6x_3 : (x_1, x_2, x_3) \in SP_{\infty}^*\}$ . Does its optimal solution yield a most violated separator for  $(f, f_0)$  and  $SP$ ? If not, what is the most violated separator in this case?
- Let  $P \subseteq \mathbb{R}^n$  be any nonempty, line free polyhedron and  $P_\varepsilon^1$  its  $\varepsilon$ -solidification with respect to the  $\ell_1$ -norm where  $\varepsilon > 0$ . Show that the extreme points of  $SP_\infty$  as defined in (9.81) are in one-to-one correspondence with the facets of  $P_\varepsilon^1$ . What happens if  $P$  is permitted to have lines?

- The polytope  $P$ , see Figure 9.13, has three extreme points  $x^1 = (1, 0)$ ,  $x^2 = (0, 2)$  and  $x^3 = (0, 0)$ . Consequently, the  $h_0$ -polar  $SP$  of  $P$  is given by  $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - h_0 \leq 0, 2h_2 - h_0 \leq 0, h_0 \geq 0\}$ .

$0, -h_0 \leq 0\}$ . Running the double description algorithm (or by hand calculation) we find that  $SP$  is pointed and has three extreme rays  $(-1, 0, 0)$ ,  $(0, -1, 0)$  and  $(2, 1, 2)$ .  $SP$  is the set of all separators for  $P$  and the set of normed separators  $SP_\infty$  for  $P$  is obtained from  $SP$  by intersecting  $SP$  with the constraints  $-1 \leq h_j \leq 1$  for  $j = 1, 2$  as we are working with the  $\ell_1$ -norm. Using the homogenization (7.5) and running the double description algorithm (or by hand calculation) we find that  $SP_\infty$  is pointed and has a minimal generator consisting of the eight extreme points

$$(0, 0, 0), (-1, 0, 0), (0, -1, 0), (-1, -1, 0), (1, 1, 2), (1, -1, 1), (1, \frac{1}{2}, 1), (-1, 1, 2)$$

and the extreme ray given by  $(0, 0, 1)$ . The set of normed separators  $SP_\infty^*$  for  $SP$  given by (9.82) is the polytope

$$SP_\infty^* = \{\mathbf{x} \in \mathbb{R}^3 : 2x_1 + x_2 - 2x_3 \leq 0, -x_1 \leq 0, -x_2 \leq 0, -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1, 0 \leq x_3 \leq 1\}.$$

Using the double description algorithm (or by hand calculation) we find that a minimal generator has the following six extreme points

$$(0, 0, 0), (0, 0, 1), (0, 1, \frac{1}{2}), (0, 1, 1), (1, 0, 1), (\frac{1}{2}, 1, 1).$$

To answer the second part of this problem we first calculate the  $\varepsilon$ -solidification  $P_\varepsilon^1$  in the  $\ell_1$ -norm. To do so we proceed like in the proof of Exercise 9.7(vi). To calculate  $P_\varepsilon^1$  we thus have to project out the  $\mu$ -variables from the polyhedron

$$\begin{aligned} PP_\varepsilon^1 = \{(\mathbf{z}, \boldsymbol{\mu}) \in \mathbb{R}^4 : & -z_1 - z_2 + \mu_1 + 2\mu_2 \leq \varepsilon, -z_1 + z_2 + \mu_1 - 2\mu_2 \leq \varepsilon, z_1 - z_2 - \mu_1 + 2\mu_2 \leq \varepsilon, \\ & z_1 + z_2 - \mu_1 - 2\mu_2 \leq \varepsilon, \mu_1 + \mu_2 \leq 1, \mu_1 \geq 0, \mu_2 \geq 0\} \end{aligned}$$

where we have used that  $\mathbf{x} = \mathbf{0}$  is an extreme point of  $P$ . To do so we need a minimal generator of the cone (for the general definition see(7.8))

$$C = \{\mathbf{u} \in \mathbb{R}^7 : u_1 - u_2 + u_3 - u_4 + u_5 - u_6 = 0, 2u_1 + 2u_2 - 2u_3 - 2u_4 + u_5 - u_7 = 0, \mathbf{u} \geq \mathbf{0}\}.$$

Running the double description algorithm we find that  $C$  is pointed and has the following ten extreme rays

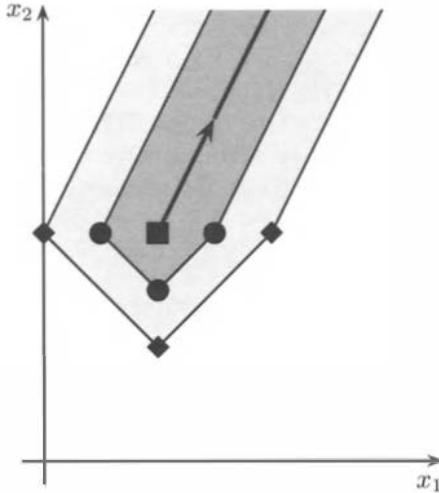
$$(1, 0, 0, 1, 0, 0, 0), (0, 1, 1, 0, 0, 0, 0), (1, 0, 1, 0, 0, 2, 0), (0, 0, 1, 0, 2, 3, 0), (0, 0, 0, 0, 1, 1, 1), (0, 0, 0, 1, 2, 1, 0), (0, 1, 0, 3, 4, 0, 0), (0, 1, 0, 0, 1, 0, 3), (1, 1, 0, 0, 0, 0, 4), (1, 0, 0, 0, 1, 2).$$

Consequently, we find (besides some trivial redundant inequalities) that  $P_\varepsilon^1$  is given by

$$\begin{aligned} P_\varepsilon^1 = \{\mathbf{x} \in \mathbb{R}^2 : & 2x_1 + x_2 \leq 2 + 2\varepsilon, x_1 + x_2 \leq 2 + \varepsilon, -x_1 + x_2 \leq 2 + \varepsilon, \\ & x_1 - x_2 \leq 1 + \varepsilon, -x_1 - x_2 \leq \varepsilon, -x_1 \leq \varepsilon, -x_2 \leq \varepsilon\}. \end{aligned}$$

From Figure 9.13, we see that every inequality of the linear description of  $P_\varepsilon^1$  corresponds to a nonzero extreme point of  $SP_\infty$  and vice versa. Note, however, that you have to normalize the first inequality of  $P_\varepsilon^1$  to get the correspondence. The extreme points of  $P_\varepsilon^1$  for  $\varepsilon \geq 0$  are

$$(-\varepsilon, 0), (0, -\varepsilon), (1, -\varepsilon), (1 + \varepsilon, 0), (\varepsilon, 2), (0, 2 + \varepsilon), (-\varepsilon, 2).$$



**Fig. 9.14.**  $\varepsilon$ -solidifications of  $P$  for  $\varepsilon = 0, 1/2$  and  $1$

(ii) The polyhedron  $P$ , see Figure 9.14, is a flat consisting of the extreme point  $x = (1, 2)$  and the direction vector  $y = (1, 2)$ . Consequently, the  $h_0$ -polar  $SP$  of  $P$  is the cone  $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 + 2h_2 - h_0 \leq 0, h_1 + 2h_2 \leq 0\}$ . Running the double description algorithm (or by hand calculation) we find that  $SP$  is a blunt cone, the basis of the lineality space of  $SP$  is given by  $(-2, 1, 0)$  and the conical part of  $SP$  is generated by  $(0, 0, 1), (-1, 0, -1)$ . So a minimal generator of  $SP$  is given by  $\{(-2, 1, 0), (2, -1, 0), (0, 0, 1), (-1, 0, -1)\}$ . The set  $SP_\infty$  of normed separators for  $P$  is obtained by intersecting  $SP$  with the constraints  $-1 \leq h_j \leq 1$  for  $j = 1, 2$  as we are working with the  $\ell_1$ -norm. Using the homogenization (7.5) and running the double description algorithm, we find that  $SP_\infty$  is a pointed polyhedron. Its minimal generator consists of the four extreme points  $(-1, \frac{1}{2}, 0), (1, -\frac{1}{2}, 0), (-1, -1, -3),$  and  $(1, -1, -1)$ , and the extreme ray given by  $(0, 0, 1)$ . The set of normed separators  $SP_\infty^*$  for  $SP$  is the polytope

$$SP_\infty^* = \{x \in \mathbb{R}^3 : 2x_1 - x_2 = 0, -x_1 + x_3 \leq 0, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1\}.$$

Using the double description algorithm we find that a minimal generator of  $SP_\infty^*$  has the following three extreme points  $(0, 0, 0), (\frac{1}{2}, 1, 0)$ , and  $(\frac{1}{2}, 1, \frac{1}{2})$ . To answer the second part of this problem we calculate the  $\varepsilon$ -solidification  $P_\varepsilon^1$  of  $P$  in the  $\ell_1$ -norm. To do so we proceed like in part (i). To calculate  $P_\varepsilon^1$  we thus have to project out variable  $\nu_1$  from the polyhedron

$$\begin{aligned} PP_\varepsilon^1 = \{&(z, \nu_1) \in \mathbb{R}^3 : -z_1 - z_2 + 3\nu_1 \leq -3 + \varepsilon, z_1 - z_2 + \nu_1 \leq -1 + \varepsilon, -z_1 + z_2 - \nu_1 \leq 1 + \varepsilon, \\ &z_1 + z_2 - 3\nu_1 \leq 3 + \varepsilon, \nu_1 \geq 0\} \end{aligned}$$

where we have simply eliminated the  $\mu$  variable since it must equal one. We thus need a minimal generator of the cone

$$C = \{u \in \mathbb{R}^5 : 3u_1 + u_2 - u_3 - 3u_4 - u_5 = 0, u \geq 0\}.$$

Running the double description algorithm we get the following six extreme rays

$$(1, 0, 0, 1, 0), (1, 0, 3, 0, 0), (0, 1, 0, 0, 1), (0, 3, 0, 1, 0), (0, 1, 1, 0, 0), (1, 0, 0, 0, 3).$$

Consequently, we find that (up to some redundant inequalities)  $P_\varepsilon^1$  is given by

$$P_\varepsilon^1 = \{x \in \mathbb{R}^2 : -2x_1 + x_2 \leq 2\varepsilon, 2x_1 - x_2 \leq 2\varepsilon, -x_1 - x_2 \leq -3 + \varepsilon, x_1 - x_2 \leq -1 + \varepsilon\}.$$

From Figure 9.14 we see that after normalization every inequality of the linear description of  $P_\varepsilon^1$  corresponds to a nonzero extreme point of  $SP_\infty$  and vice versa. Note that as in part (i) you have to normalize the first and second inequalities of  $P_\varepsilon^1$  to get the correspondence. The three extreme points of  $P_\varepsilon^1$  for  $\varepsilon \geq 0$  are  $(1 - \varepsilon, 2)$ ,  $(1, 2 - \varepsilon)$ ,  $(1 + \varepsilon, 2)$ . In addition we need the direction vector  $y = (1, 2)$  of the extreme ray of  $P$  for a minimal pointwise description of  $P_\varepsilon^1$ .

**(iii)** The polyhedron  $P$ , see Figure 9.15, is an unbounded set having two extreme points  $(1, 3)$  and  $(2, 1)$ , and two direction vectors  $(1, 0)$ , and  $(1, 1)$  for its extreme rays. Consequently, the  $h_0$ -polar  $SP$  of  $P$  is the cone  $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - 3h_2 - h_0 \leq 0, 2h_1 - h_2 - h_0 \leq 0, h_1 \leq 0, h_1 + h_2 \leq 0\}$ . Running the double description algorithm we find that  $SP$  is a pointed cone having four extreme rays  $(0, 0, 1)$ ,  $(0, -1, 3)$ ,  $(-2, 1, -5)$  and  $(-1, 1, -3)$ . The set  $SP_\infty$  of the normed separators for  $P$  is obtained by intersecting  $SP$  with the constraints  $-1 \leq h_j \leq 1$  for  $j = 1, 2$ . Using the homogenization (7.5) and running the double description algorithm we find that  $SP_\infty$  is a pointed polyhedron. Its minimal generator consists of the five extreme points

$$(0, 0, 0), (0, -1, 3), (-1, -1, 2), (-1, \frac{1}{2}, -\frac{5}{2}), (-1, 1, -3)$$

and the extreme ray given by the direction vector  $(0, 0, 1)$ . The set of normed separators  $SP_\infty^*$  for  $SP$  is the polytope  $SP_\infty^* = \{x \in \mathbb{R}^3 : 2x_1 + x_2 - 5x_3 \geq 0, x_1 - x_2 + 2x_3 \geq 0, x_2 - x_3 \geq 0, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1\}$ . Running the double description algorithm we find that the (quasi-unique) minimal generator of  $SP_\infty^*$  has the six extreme points

$$(0, 0, 0), (1, 0, 0), (1, 1, 0), (1, \frac{1}{2}, \frac{1}{2}), (1, 1, \frac{3}{5}), (\frac{1}{3}, 1, \frac{1}{3}).$$

To answer the second part of this problem, we calculate the  $\varepsilon$ -solidification  $P_\varepsilon^1$  of  $P$  in the  $\ell_1$ -norm. To do so we proceed like in parts (i) and (ii). To calculate  $P_\varepsilon^1$  we thus have to project out the  $\mu$  and  $\nu$  variables from the polyhedron

$$\begin{aligned} PP_\varepsilon^1 = \{(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^6 : & z_1 + z_2 - 4\mu_1 - 3\mu_2 - \nu_1 - 2\nu_2 \leq \varepsilon, -z_1 + z_2 - 2\mu_1 + \mu_2 + \nu_1 \leq \varepsilon, \\ & z_1 - z_2 + 2\mu_1 - \mu_2 - \nu_1 \leq \varepsilon, -z_1 - z_2 + 4\mu_1 + 3\mu_2 + \nu_1 + 2\nu_2 \leq \varepsilon, \mu_1 + \mu_2 = 1, \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\nu} \geq \mathbf{0}\}. \end{aligned}$$

To carry out the projection we calculate the cone (7.8):

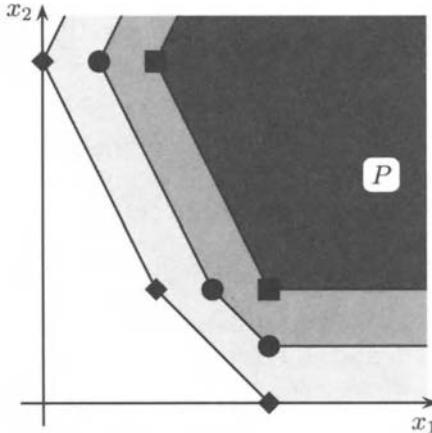
$$\begin{aligned} C = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^9 : & -4u_1 - 2u_2 + 2u_3 + 4u_4 - u_5 + v_1 = 0, -3u_1 + u_2 - u_3 + 3u_4 - u_6 + v_1 = 0, \\ & -u_1 + u_2 - u_3 + u_4 - u_7 = 0, -2u_1 + 2u_4 - u_8 = 0, \mathbf{u} \geq \mathbf{0}\}. \end{aligned}$$

Running the double description algorithm we get the seven extreme rays

$$\begin{aligned} (1, 0, 0, 1, 0, 0, 0, 0, 0), (0, 1, 1, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 1, 1, 0, 0, 1), (0, 1, 0, 0, 0, 3, 1, 0, 2), \\ (0, 0, 0, 1, 1, 0, 1, 2, -3), (0, 1, 0, 3, 0, 0, 4, 6, -10), (0, 0, 1, 1, 4, 0, 0, 2, -2), \end{aligned}$$

and thus we calculate

$$P_\varepsilon^1 = \{x \in \mathbb{R}^2 : 2x_1 + x_2 \geq 5 - 2\varepsilon, x_1 - x_2 \geq -2 - \varepsilon, x_2 \geq 1 - \varepsilon, x_1 + x_2 \geq 3 - \varepsilon\}.$$



**Fig. 9.15.**  $\varepsilon$ -solidifications of  $P$  for  $\varepsilon = 0, 1/2$  and  $1$

From Figure 9.15 we see that after normalization every inequality of the linear description of  $P_\varepsilon^1$  corresponds to a nonzero extreme point of  $SP_\infty^*$  and vice versa. The extreme points of  $P_\varepsilon^1$  are  $(1 - \varepsilon, 3)$ ,  $(2 - \varepsilon, 1)$ ,  $(2, 1 - \varepsilon)$ . In addition we need the two direction vectors of the extreme rays of  $P$  for a minimal pointwise description of  $P_\varepsilon^1$ .

From the above pointwise description of  $SP_\infty^*$  we find that  $\tilde{x} = (1, 1, \frac{3}{5})$  is the unique optimal solution to  $\max\{\mathbf{f}x - f_0 x_{n+1} : (x, x_{n+1}) \in SP_\infty^*\}$  and since  $f_1 x_1 + f_2 x_2 - f_0 x_3 = 1.6 > 0$  the point  $\tilde{x}$  separates  $(\mathbf{f}, f_0)$  from the cone  $SP$ . To find the most violated separator for  $(\mathbf{f}, f_0)$  and  $SP$ , we apply the procedure described on pages 344-346; see (9.83) and (9.84). Solving  $\max\{\mathbf{f}x : x \in P\}$  we find the (unique) optimizer  $x^{max} = (2, 1)$  and thus (9.83) applies. We get  $\alpha = 1/2$  and thus  $x^0 = (1, \frac{1}{2}, \frac{1}{2})$  is a most violated separator for  $(\mathbf{f}, f_0)$  and  $SP$ , i.e., a most violated separator for  $(\mathbf{f}, f_0)$  and the cone  $SP$  cannot be found by solving the linear program  $\max\{\mathbf{f}x - f_0 x_{n+1} : (x, x_{n+1}) \in SP_\infty^*\}$ .

**(iv)** Let  $(\mathbf{h}, h_0)$  be an extreme point of  $SP_\infty$ . Then  $\mathbf{h}x \leq h_0$  for all  $x \in P$  and by Exercise 9.7(iv)  $\mathbf{h}x \leq \tilde{h}_0 = h_0 + \varepsilon$  for all  $x \in P_\varepsilon^1$  since  $\|\mathbf{h}\|_\infty = 1$ . Since  $\varepsilon > 0$  and by Exercise 9.7(ix)  $\dim P_\varepsilon^1 = n$ , there exists  $(\mathbf{f}, \tilde{f}_0) \in \mathbb{R}^{n+1}$  such that  $\|\mathbf{f}\|_\infty = 1$ ,  $\mathbf{f}x \leq \tilde{f}_0$  defines a facet of  $P_\varepsilon^1$  and  $A = \{x \in P_\varepsilon^1 : \mathbf{h}x = \tilde{h}_0\} \subseteq B = \{x \in P_\varepsilon^1 : \mathbf{f}x = \tilde{f}_0\}$ . By Exercise 9.7(vi)  $\mathbf{f}x \leq f_0 = \tilde{f}_0 - \varepsilon$  for all  $x \in P$  and thus  $(\mathbf{f}, f_0) \in SP_\infty$ . From  $A \subseteq B$  it follows that  $\mathbf{h}x^i = h_0$  implies  $\mathbf{f}x^i = f_0$  for  $1 \leq i \leq p$  and likewise  $\mathbf{h}y^i = 0$  implies  $\mathbf{f}y^i = 0$  for  $1 \leq i \leq r$ . Suppose  $h_j = \pm 1$  for some  $j \in \{1, \dots, n\}$ . Since  $P$  is pointed,  $\mathbf{h}x^i = h_0$  for some  $i \in \{1, \dots, p\}$ . But  $x^i \pm \varepsilon u_j \in P_\varepsilon^1$ ,  $\mathbf{h}(x^i \pm \varepsilon u_j) = h_0 \pm \varepsilon h_j = \tilde{h}_0$  and thus  $\mathbf{f}(x^i \pm \varepsilon u_j) = f_0 \pm \varepsilon f_j = \tilde{f}_0 = f_0 + \varepsilon$  implies  $f_j = h_j$ . Since  $(\mathbf{h}, h_0)$  is an extreme point it follows that  $(\mathbf{h}, h_0) = (\mathbf{f}, f_0)$ , i.e.,  $\mathbf{h}x \leq h_0 + \varepsilon$  defines a facet of  $P_\varepsilon^1$ . To show the reverse statement, suppose  $(\mathbf{h}, \tilde{h}_0) \in \mathbb{R}^{n+1}$  defines a facet of  $P_\varepsilon^1$ . We can assume WROG that  $\|\mathbf{h}\|_\infty = 1$  and thus  $(\mathbf{h}, h_0) \in SP_\infty$  where  $h_0 = \tilde{h}_0 - \varepsilon$ . Denote by  $(\mathbf{h}^i, h_0^i)$  for  $1 \leq i \leq s$  the extreme points of  $SP_\infty$ . Since  $SP_\infty$  has exactly one halfline, it follows that  $(\mathbf{h}, h_0) = \sum_{i=1}^s \mu_i (\mathbf{h}^i, h_0^i) + \lambda(\mathbf{0}, 1)$  with  $\mu_i \geq 0$ ,  $\sum_{i=1}^s \mu_i = 1$  and  $\lambda \geq 0$ . Suppose  $\lambda > 0$ . Then  $\mathbf{h}x \leq h_0 - \lambda$  for all  $x \in P$ , because  $(\mathbf{h}, h_0 - \lambda)$  is a nonnegative combination of  $\mathbf{h}^i x \leq h_0^i$  - which by the first part define facets of  $P$ . But then  $\mathbf{h}x \leq h_0 - \lambda + \varepsilon < \tilde{h}_0$  for all  $x \in P_\varepsilon^1$  shows the contradiction. Consequently,  $\lambda = 0$  and thus  $(\mathbf{h}, \tilde{h}_0) = \sum_{i=1}^s \mu_i (\mathbf{h}^i, \tilde{h}_0^i)$  with

$\mu_i \geq 0$ ,  $\sum_{i=1}^s \mu_i = 1$ , where  $\tilde{h}_0^i = h_0^i + \varepsilon$  for  $1 \leq i \leq s$ . Since  $\dim P_\varepsilon^1 = n$  it follows that  $(\mathbf{h}, \tilde{h}_0) = (\mathbf{h}^i, \tilde{h}_0^i)$  for some  $i \in \{1, \dots, s\}$  since the linear description of a full dimensional polyhedron by its facets is unique *modulo* the multiplication by positive scalars; see page 129 of the book. Consequently,  $(\mathbf{h}, h_0)$  defines an extreme point of  $SP_\infty$  and the proof is complete.

As we did not utilize the extremality of  $\mathbf{x}^1, \dots, \mathbf{x}^p$  in the above argument it follows that the statement about the correspondence remains correct if  $P$  contains lines. If  $p = 0$ , then the feasible  $\mathbf{x}^i$  needed to prove that  $f_j = h_j$  can be chosen to equal 0 since  $0 \in P$  in this case.

# 10. Combinatorial Optimization: An Introduction

Sempre avanti...<sup>1</sup>  
Italian saying.

Combinatorial optimization problems arise typically in the form of a **mixed-integer linear program**

$$(MIP) \quad \max\{cx + dy : Ax + Dy \leq b, x \geq 0 \text{ and integer}, y \geq 0\},$$

where  $A$  is any  $m \times n$  matrix of reals,  $D$  is an  $m \times p$  matrix of reals,  $b$  is a column vector with  $m$  real components and  $c$  and  $d$  are real row vectors of length  $n$  and  $p$ , respectively. If  $n = 0$  then we have a linear program. If  $p = 0$  then we have a **pure integer program**. The variables  $x$  that must be integer-valued are the *integer variables* of the problem, the variables  $y$  the *real or flow variables*. There are frequently explicit upper bounds on either the integer or real variables or both. In many applications the integer variables model yes/no decisions, i.e., they assume only the values of zero or one. In this case the problem (MIP) is a mixed zero-one or a zero-one linear program depending on  $p > 0$  or  $p = 0$ .

Another way in which combinatorial problems arise goes as follows: given some finite ground set  $E = \{1, \dots, g\}$  of  $g$  distinct elements let  $\mathcal{F}$  be a finite family of not necessarily distinct subsets  $F \subseteq E$  that satisfy certain well-defined conditions. Let  $c_e$  for all  $e \in E$  be the “cost” of element  $e$  and define  $c_F = \sum_{e \in F} c_e$  to be the cost of  $F \in \mathcal{F}$ . We want to find  $F^* \in \mathcal{F}$  such that the cost of  $F^*$  is minimal, i.e.,

$$\min\left\{\sum_{e \in F} c_e : F \in \mathcal{F}\right\}. \quad (10.1)$$

Let  $\mathbb{R}^E$  (rather than  $\mathbb{R}^{|E|}$ ) denote the  $|E|$ -dimensional real space of vectors of length  $|E|$ . With every element  $F \in \mathcal{F}$  we associate a 0-1 point  $x^F = (x_e^F)_{e \in E} \in \mathbb{R}^E$  as follows

$$x_e^F = \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{if not.} \end{cases} \quad (10.2)$$

$x^F$  is the *incidence vector* or *characteristic vector* of  $F \subseteq E$ . Then (10.1) becomes  $\min\{cx^F : F \in \mathcal{F}\}$ , where  $c = (c_e)_{e \in E}$  is a row vector. To solve (10.1) we need to find the minimum of a linear objective function over a finite set of  $|\mathcal{F}|$  zero-one points in  $\mathbb{R}^E$ . In most cases of interest it is not difficult to find a “formulation” in terms of linear relations of this problem that, together with the requirement that the variables  $x_e$  be zero or one, express the conditions defining  $\mathcal{F}$ , i.e. we can bring (10.1) into the form of (MIP). The “size” of such a formulation in terms of the parameter  $|E|$  of the number of variables may be exponential in  $|E|$ . An example to this effect is the set of zero-one points that correspond to extreme points of the polytope  $O_n$  of Exercise 7.7 in which case  $E = \{1, \dots, n\}$  and  $\mathcal{F} = \{F \subseteq E : |F| \text{ is odd}\}$ . See also Appendix C.

## 10.1 The Berlin Airlift Model Revisited

To be concrete, let us consider one way by which we can approach the solution of the Berlin airlift model when all of its variables are required to be integers; see Exercise 1.4 (Minicase IV).

<sup>1</sup>Forwards, forwards,.....

We start by solving the linear program displayed in Table 1.10 and obtain the solution displayed in the left part of Table 1.11. All but the two variables  $P_{i_1} = 7.311$  and  $P_{n_1} = 453.789$  are integer-valued. If *all* of the variables had assumed integer values only, we could have stopped. This is not the case and it is trivially correct that either  $P_{i_1} \leq 7$  or  $P_{i_1} \geq 8$  in every integer solution to the problem. To get an integer solution, we create two corresponding new problems from the original one, both of which we put on a “problem stack”. We call variable  $P_{i_1}$  the “branching” variable, because we used it to split the original problem into two problems. We select one of the two problems from the stack and solve a new linear program.

Let us take the problem with the added constraint  $P_{i_1} \leq 7$  from the stack. The new linear program solution yields an objective function value of  $46,826.168 > 46,784.867$  which is the old value. In the new linear programming solution we get  $P_{i_1} = 7$ , but e.g.  $P_{i_2} = 6.211$ . Now we are ready to iterate: we select  $P_{i_2}$  as the “next” branching variable, we create two new problems where we require  $P_{i_2} \leq 6$  in one of the two and  $P_{i_2} \geq 7$  in the other, and put both of them on the problem stack which now has three problems on it. Then we select a problem from the stack and continue.

In Figure 10.1 we display the binary search tree that results from the iterative application of the basic idea. The nodes of the tree are numbered in the order of their creation. If a linear program in the search is infeasible, we can evidently drop the problem from consideration. We can do likewise, when the linear programming solution is integer or when the objective function value exceeds a best value for an integer solution obtained so far. This way we can “prune” (or fathom nodes of) the search tree.

The method that we have just described is called **branch-and-bound**. It dates from the 1950's and is the only method that commercial packages have implemented until fairly recently (about mid 1990's). The essential ingredients into a branch-and-bound algorithm for the solution of (MIP) are

- a computer program for the solution of linear programs or an LP solver, for short,
- a set of choice rules for branching variable and problem selection, and
- an accounting mechanism that keeps track of the search tree.

We let you figure out which choice rules we have used to produce the search tree of Figure 10.1.

Another way to obtain integrality of the solution goes as follows: since the solution obtained from the first linear program is basic and noninteger, it follows that there must be *cutting planes* or *separating hyperplanes*, i.e., some linear inequalities that are satisfied by all integer solutions to the problem and which cut off the linear programming optimum. The problem is to find such cuts for the Berlin airlift example.

To simplify notation let us set  $x_i = P_{i_1}$  and  $y_i = P_{n_1}$  for  $1 \leq i \leq 4$ . After eliminating the cargo variables, the constraints for crew management are of the general form

$$\begin{aligned} -x_{i-1} + x_i - y_{i-1} + \alpha^{-1}y_i &= d_i && \text{for } 1 \leq i \leq T, \\ x_0 = y_0 &= 0, \quad x_i \geq 0, \quad y_i \geq 0, \quad x_i \text{ and } y_i \text{ integer} && \text{for } 1 \leq i \leq T, \end{aligned} \tag{10.3}$$

where  $\alpha > 1$  and  $d_i$  for  $1 \leq i \leq T$  are integers, which shows a definite “structural” pattern. In our case  $\alpha = 20$ ,  $d_1 = 30$ ,  $d_2 = -450$ ,  $d_3 = -210$ ,  $d_4 = -240$  and  $T = 4$ . Multiplying each equation by  $\alpha$

and transforming the constraint set by multiplying it with the matrix

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ -\alpha & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \alpha & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \alpha^{T-1} & \alpha^{T-2} & \cdots & 1 \end{pmatrix}$$

we get the following *equivalent* system of equations

$$\begin{aligned} (\alpha - 1) \sum_{j=1}^{i-1} \alpha^{i-j} x_j + \alpha x_i + y_i &= \sum_{j=1}^i \alpha^{i+1-j} d_j \quad \text{for } 1 \leq i \leq T, \\ x_i \geq 0, \ y_i \geq 0, \ x_i \text{ and } y_i \text{ integer} &\quad \text{for } 1 \leq i \leq T. \end{aligned} \tag{10.4}$$

Since all variables must be nonnegative in (10.4) it follows that every feasible solution satisfies

$$(\alpha - 1) \sum_{j=1}^{i-1} \alpha^{i-j} x_j \leq \sum_{j=1}^i \alpha^{i+1-j} d_j \quad \text{for } 2 \leq i \leq T.$$

Dividing both sides of the inequality by  $\alpha(\alpha - 1)$  it follows from the integrality of  $\alpha$  that

$$\sum_{j=1}^{i-1} \alpha^{i-1-j} x_j \leq \left\lfloor \sum_{j=1}^i \alpha^{i-j} d_j / (\alpha - 1) \right\rfloor \quad \text{for } 2 \leq i \leq T \tag{10.5}$$

are inequalities that every nonnegative integer solution to (10.4) must satisfy.

In the case of the Berlin airlift model we get from (10.5) three inequalities in the original variables

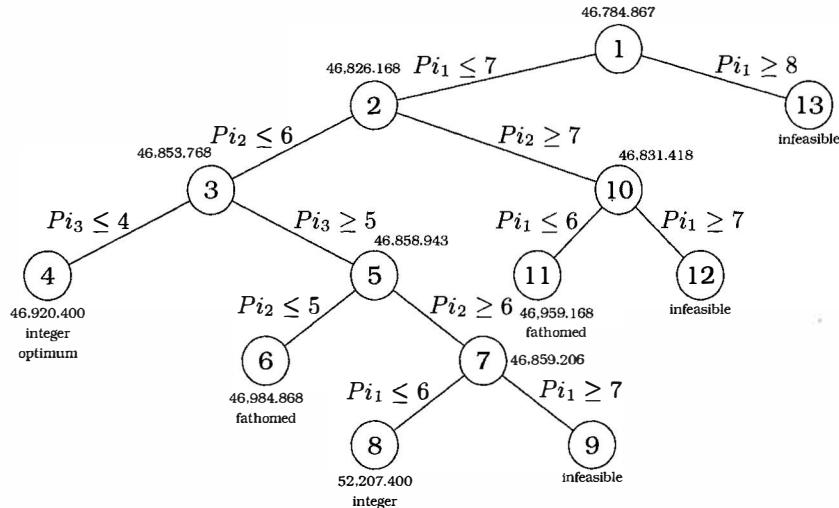
$$Pi_1 \leq 7, \ 20Pi_1 + Pi_2 \leq 146, \ 400Pi_1 + 20Pi_2 + Pi_3 \leq 2924, \tag{10.6}$$

that every feasible *integer* solution to the problem must satisfy. As you know from part (v) of Minicase IV, adding these cuts to the linear program and solving the augmented linear program, we find the integer solution without branching at all.

This proceeding can be generalized and is known as the **cutting plane approach** to mixed-integer programming. Classical cutting planes developed in 1950's and early 1960's had, at best, mixed computational success, but are ultimately responsible for the most successful approach to mixed-integer programming, **branch-and-cut**, which combines cutting planes with branching in the solution process.

Figure 10.2 shows a flow-chart of a typical branch-and-cut problem solver for the maximization of a mixed-integer program (MIP). It has four major building blocks besides a branching mechanism that works just like traditional branch-and-bound.

- In the *preprocessor* the problem is automatically inspected and the current formulation is improved.
- The *LP solver* solves the linear programs that are encountered during the course of calculation.

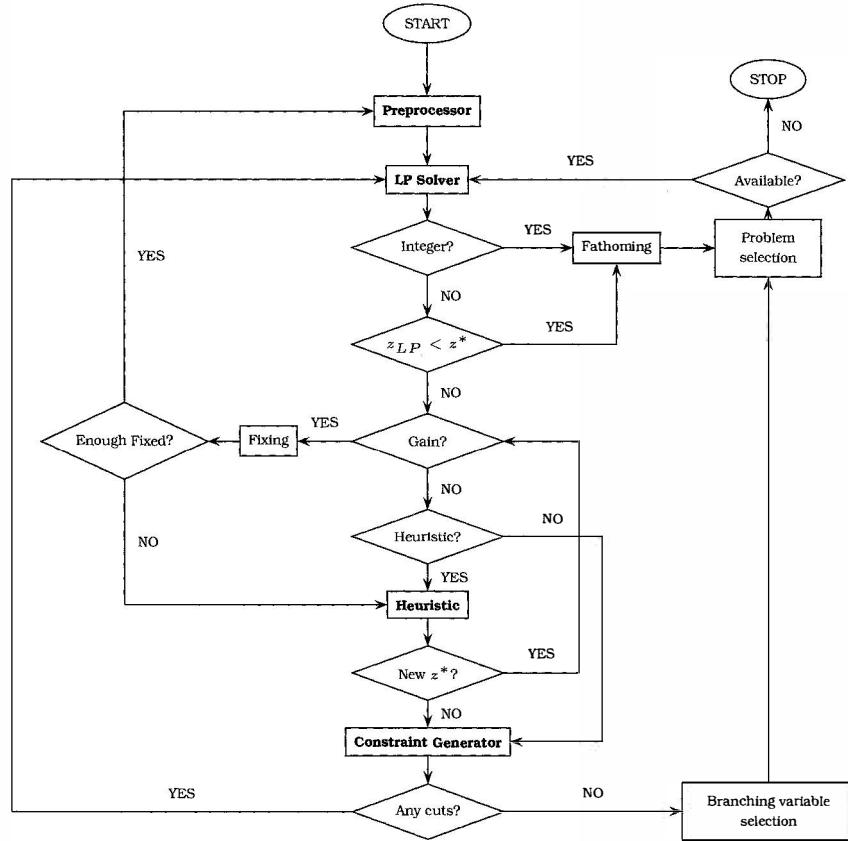


**Fig. 10.1.** Search tree for the Berlin airlift model

- The *heuristic* attempts to find good feasible solutions to the problem at hand. It can be a stand-alone procedure or better, it can be based on the current linear program. By fixing certain variables e.g. via *rounding*, solving the linear program that results and repeating it is often possible to find good feasible solutions reasonably quickly.
- The *constraint generator* is the motor of the system. It generates cuts like (10.6) and adds them to the current linear program which is subsequently reoptimized – like in the dynamic simplex algorithm. Also like in the dynamic simplex algorithm constraints that are not needed are *purged* from the active set so as to keep the linear programs small.

Sophisticated branch-and-cut solvers also add/drop columns, i.e. they have a column generator subroutine as well – which is not shown in Figure 10.2 to keep things simple. The constraint generator incorporates the results of a theoretical analysis that must precede the numerical solution of difficult combinatorial problems. If the constrained generator finds no violated constraints – due to incomplete knowledge of the problem to be solved – the problem solver resorts to branching, just like in branch-and-bound.

In the flow-chart of Figure 10.2 the symbol  $z^*$  refers to the objective function value of the “best” integer or mixed-integer solution to the maximization problem (MIP) obtained so far,  $z_{LP}$  is the objective function value of the current linear program. By design the branch-and-cut solver works with a lower bound to the problem (as soon as a feasible solution to (MIP) is known) and an upper bound provided for by the solution of the linear programs. Using e.g. the reduced cost of the linear program at the “root” node of the search tree it may become possible to “fix” certain variables at their upper or lower bounds without loosing an optimal solution. If a “sufficiently” large number of variables has been fixed, then the problem is preprocessed again, etc. For more detail we refer you to the references to this section of the text that deal with branch-and-cut.



**Fig. 10.2.** Flow-chart of a branch-and-cut problem solver

## 10.2 Complete Formulations and Their Implications

Given a mixed integer linear program (MIP) denote by

$$P(\mathbf{A}, \mathbf{D}, \mathbf{b}) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{Ax} + \mathbf{Dy} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}\} \quad (10.7)$$

the polyhedron of the linear programming relaxation of the constraint set of (MIP) and by

$$(\text{MIP}_{LP}) \quad \max\{\mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} : (\mathbf{x}, \mathbf{y}) \in P(\mathbf{A}, \mathbf{D}, \mathbf{b})\}$$

the linear programming relaxation of (MIP). The set feasible solutions to (MIP) is a *discrete mixed set* in  $\mathbb{R}^{n+p}$  (or if  $p = 0$  a *discrete set* in  $\mathbb{R}^n$ ) that we denote by

$$DM = P(\mathbf{A}, \mathbf{D}, \mathbf{b}) \cap (\mathbb{Z}^n \times \mathbb{R}^p), \quad (10.8)$$

where  $\mathbb{Z}^n$  is the *lattice* of all  $n$ -tuples  $(x_1, \dots, x_n)$  with  $x_i$  *integer* for  $i = 1, \dots, n$  and  $\mathbb{Z}^n \times \mathbb{R}^p$  is the usual cross product; see Figure 10.3 and the figures in Exercise 10.2 for examples of discrete mixed sets in  $\mathbb{R}^2$ .

For all “real world” problems we are assuming that we know a linear description  $A\mathbf{x} + D\mathbf{y} \leq \mathbf{b}$ ,  $\mathbf{x} \geq \mathbf{0}$ ,  $\mathbf{y} \geq \mathbf{0}$  such that  $DM$  satisfies (10.8).  $A'$ ,  $D'$ ,  $\mathbf{b}'$  different from  $A$ ,  $D$ ,  $\mathbf{b}$  may exist that describe the same underlying discrete mixed set  $DM$ . We call any finite set of linear inequalities that “model” the discrete mixed set  $DM$  a *formulation* of the underlying problem. Thus by relation (10.7) a formulation is a *polyhedron* in  $\mathbb{R}^{n+p}$  and vice versa, any polyhedron in  $\mathbb{R}^{n+p}$  whose intersection with  $\mathbb{Z}^n \times \mathbb{R}^p$  equals the discrete mixed set  $DM$  is a formulation of the underlying problem; see the text for more on “formulations”.

Call a formulation  $P(A, D, b)$  a **complete formulation** for (MIP) if the corresponding linear programming relaxation ( $MIP_{LP}$ ) solves (MIP) no matter what  $(c, d) \in \mathbb{R}^{n+p}$  is used for the objective function of (MIP). Before addressing the *existence* of a *complete formulation* for (MIP) consider an example.

**Example.** Suppose we have  $n+1$  zero-one variables and that we wish to formulate the implication “if  $x_j > 0$  for some  $j \in \{1, \dots, n\}$  then  $x_{n+1} = 1$ ” like we do when we model a fixed cost or a set-up cost in some larger setting. We can formulate this “compactly” using a single linear constraint

$$(F_1) \quad \begin{aligned} \sum_{j=1}^n x_j &\leq Kx_{n+1} \\ 0 \leq x_j &\leq 1, \quad x_j \text{ integer for } j = 1, \dots, n+1, \end{aligned}$$

where  $K \geq n$  is arbitrary, but we can also formulate the problem in “disaggregated” form

$$(F_2) \quad \begin{aligned} x_j &\leq x_{n+1} && \text{for } j = 1, \dots, n \\ 0 \leq x_j &\leq 1, \quad x_j \text{ integer for } j = 1, \dots, n+1. \end{aligned}$$

In Exercise 10.3 you are asked to prove that  $(F_2)$  is (locally) a *complete formulation* and that it is *better* than  $(F_1)$ .

Since we optimize *linear* objective functions over the discrete mixed set  $DM$  we can *convexify*. Define

$$P_I(\mathbf{A}, \mathbf{D}, \mathbf{b}) = \text{conv}(DM) \tag{10.9}$$

to be the convex hull of  $DM$ . From the linearity of the objective function

$$\max\{\mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} : (\mathbf{x}, \mathbf{y}) \in DM\} = \max\{\mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} : (\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})\},$$

– provided that a maximizer of (MIP) *exists*. It is not difficult to give examples where this is not the case. If  $\alpha \in \mathbb{R}$  is a positive *irrational* number, then a maximizer to the problem

$$\max\{-\alpha x_1 + x_2 : -\alpha x_1 + x_2 \leq 0, x_1 \geq 1, x_2 \geq 1, x_1, x_2 \text{ integer}\}$$

does not exist even though the objective function value is bounded from above by zero.

**10.2(a)** Let  $A$ ,  $D$  and  $b$  be any rational data and  $P = P(\mathbf{A}, \mathbf{D}, \mathbf{b})$ ,  $DM$ ,  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  be defined as in (10.7), (10.8), (10.9). Then  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  is a polyhedron in  $\mathbb{R}^{n+p}$ .

*Proof.* We can assume WROG that  $DM \neq \emptyset$  and  $n \geq 1$ . We verify Definition P2 of a polyhedron by constructing a finite generator for  $P_I$ . By Chapter 7.3.3  $P$  has a finite generator that consists of all extreme points  $(\mathbf{x}^i, \mathbf{y}^i)$  for  $i = 1, \dots, s$  and all extreme rays  $(\mathbf{r}^i, \mathbf{t}^i)$  for  $i = 1, \dots, q$  of  $P$ . Since the

data are rational all of  $(\mathbf{x}^i, \mathbf{y}^i)$  and  $(\mathbf{r}^i, \mathbf{t}^i)$  are rational vectors and by scaling  $(\mathbf{r}^i, \mathbf{t}^i)$  appropriately we can assume WROG that  $\mathbf{r}^i \in \mathbb{Z}^n$  and componentwise  $\text{g.c.d.}(r_1^i, \dots, r_n^i) = 1$  for  $i = 1, \dots, q$ . Since  $DM \subseteq P$  we can write every  $(\mathbf{x}, \mathbf{y}) \in DM$  as

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^s \mu_i (\mathbf{x}^i, \mathbf{y}^i) + \sum_{i=1}^q \lambda_i (\mathbf{r}^i, \mathbf{t}^i) \quad \text{where } \mu_i \geq 0, \sum_{i=1}^s \mu_i = 1 \text{ and } \lambda_i \geq 0. \quad (i)$$

Consider the *polytope* in  $\mathbb{R}^{n+p}$  given by

$$XY = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : (\mathbf{x}, \mathbf{y}) \text{ satisfies (i) and } 0 \leq \lambda_i \leq 1 \text{ for } i = 1, \dots, q\}. \quad (10.10)$$

The projection  $X$  of  $XY$  on  $\mathbb{R}^n$  corresponding to the variables  $\mathbf{x}$  when restricted to  $\mathbb{Z}^n$ ,

$$X = \{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{y} \in \mathbb{R}^p \text{ such that } (\mathbf{x}, \mathbf{y}) \in XY\}, \quad (10.11)$$

is a *finite* subset of  $\mathbb{Z}^n$  since  $XY$  is bounded. For each  $\mathbf{x} \in X$  let

$$Y_{\mathbf{x}} = \{\mathbf{y} \in \mathbb{R}^p : (\mathbf{x}, \mathbf{y}) \in XY\}$$

be the corresponding “continuous” part. Every  $\mathbf{y} \in Y_{\mathbf{x}}$  can be written as

$$\mathbf{y} = \sum_{i=1}^s \mu_i \mathbf{y}^i + \sum_{i=1}^q \lambda_i \mathbf{t}^i, \quad (ii)$$

where the scalars  $\mu_i$  and  $\lambda_i$  belong to the set

$$\Lambda_{\mathbf{x}} = \{(\boldsymbol{\mu}, \boldsymbol{\lambda}) \in \mathbb{R}^{s+q} : \mathbf{x} = \sum_{i=1}^s \mu_i \mathbf{x}^i + \sum_{i=1}^q \lambda_i \mathbf{r}^i, \mu_i \geq 0, \sum_{i=1}^s \mu_i = 1, 0 \leq \lambda_i \leq 1\},$$

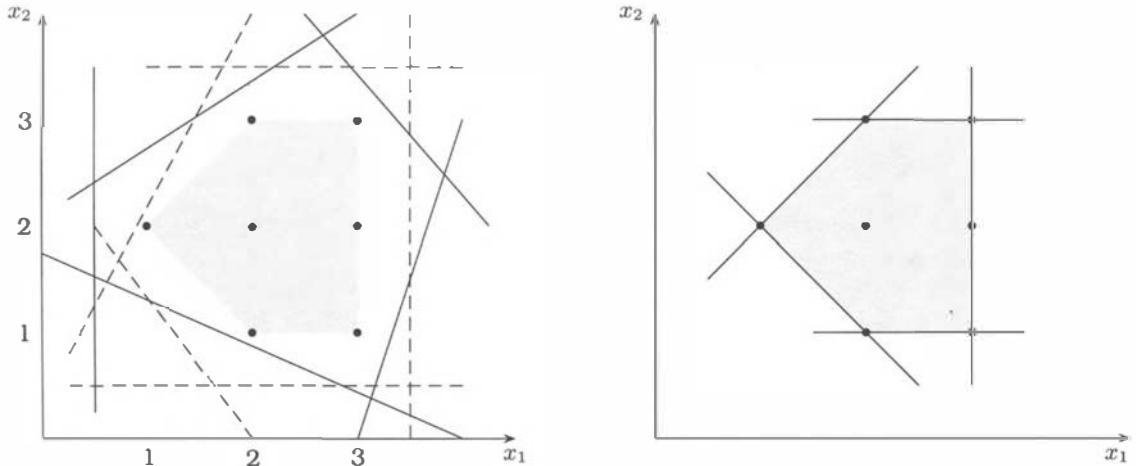
which is a *polytope* in  $\mathbb{R}^{s+q}$ . By Chapter 7.3.3, every  $(\boldsymbol{\mu}, \boldsymbol{\lambda}) \in \Lambda_{\mathbf{x}}$  can be written in turn as

$$(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j (\boldsymbol{\mu}^j, \boldsymbol{\lambda}^j) \quad \text{where } \alpha_j \geq 0, \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j = 1, \quad (iii)$$

where the vectors  $(\boldsymbol{\mu}^j, \boldsymbol{\lambda}^j) \in \mathbb{R}^{s+q}$  for  $j = 1, \dots, L_{\mathbf{x}}$  are the extreme points of  $\Lambda_{\mathbf{x}}$ . Consequently, every  $\mathbf{y} \in Y_{\mathbf{x}}$  is the convex combination of *finitely* many points of  $Y_{\mathbf{x}}$ , namely of those that correspond to the extreme points  $(\boldsymbol{\mu}^j, \boldsymbol{\lambda}^j)$  of  $\Lambda_{\mathbf{x}}$ . This follows because by (ii) and (iii) we have for every  $\mathbf{y} \in Y_{\mathbf{x}}$

$$\mathbf{y} = \sum_{i=1}^s \left( \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j \mu_i^j \right) \mathbf{y}^i + \sum_{i=1}^q \left( \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j \lambda_i^j \right) \mathbf{t}^i = \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j \left( \sum_{i=1}^s \mu_i^j \mathbf{y}^i + \sum_{i=1}^q \lambda_i^j \mathbf{t}^i \right) = \sum_{j=1}^{L_{\mathbf{x}}} \alpha_j \widehat{\mathbf{y}}^j.$$

But  $\widehat{\mathbf{y}}^j \in Y_{\mathbf{x}}$  for  $j = 1, \dots, L_{\mathbf{x}}$  and thus  $E_{\mathbf{x}} = \{\widehat{\mathbf{y}}^j \in \mathbb{R}^p : j = 1, \dots, L_{\mathbf{x}}\}$  is the finite set of points of  $Y_{\mathbf{x}}$  for each  $\mathbf{x} \in X$  as claimed. The union of these finite sets is finite and thus  $E = \{(\mathbf{x}, \widehat{\mathbf{y}}^j) : j = 1, \dots, L_{\mathbf{x}} \text{ and } \mathbf{x} \in X\}$  is a finite set since  $|X| < \infty$ . In some arbitrary indexing denote by  $(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)$



**Fig. 10.3.** Three formulations for a discrete set in  $\mathbb{R}^2$

the  $k^{th}$  element of  $E$ . By construction  $(\bar{x}^k, \bar{y}^k) \in XY$  and thus every  $(x, y) \in XY$  can be written in the form

$$(x, y) = \sum_{k=1}^K \delta_k (\bar{x}^k, \bar{y}^k) \quad \text{for some } \delta_k \geq 0, \sum_{k=1}^K \delta_k = 1, \quad (iv)$$

where  $K = \sum_{x \in X} L_x < \infty$ . By (i) we can write every  $(x, y) \in DM$  as  $(x, y) = (\tilde{x}, \tilde{y}) + \sum_{i=1}^q [\lambda_i](r^i, t^i)$  where  $(\tilde{x}, \tilde{y}) = \sum_{i=1}^s \mu_i(x^i, y^i) + \sum_{i=1}^q (\lambda_i - [\lambda_i])(r^i, t^i)$ . But  $(\tilde{x}, \tilde{y}) \in P$  satisfies  $\tilde{x} \in \mathbb{Z}^n$  since  $x \in \mathbb{Z}^n$ ,  $r^i \in \mathbb{Z}^n$  and  $[\lambda_i] \in \mathbb{Z}$  for  $i = 1, \dots, q$ . Thus  $(\tilde{x}, \tilde{y}) \in XY$ . Hence

$$(x, y) = \sum_{k=1}^K \delta_k (\bar{x}^k, \bar{y}^k) + \sum_{i=1}^q \beta_i (r^i, t^i) \quad (v)$$

for every  $(x, y) \in DM$ , where  $(\bar{x}^k, \bar{y}^k) \in XY \subseteq DM$ ,  $\delta_k \geq 0$ ,  $\sum_{k=1}^K \delta_k = 1$  and  $\beta_i \geq 0$  for  $i = 1, \dots, q$ .  $DM$  possesses thus a finite generator, so does  $P_I = \text{conv}(DM)$  and point 10.2(a) follows. ■

From Weyl's theorem, see point 7.3(h), we have the following consequence for (MIP).

**10.2(b) (Existence of complete formulations)** *If the data  $A$ ,  $D$ ,  $b$  of the problem (MIP) are rational, then there exists an integer number  $0 \leq t < \infty$  and rational matrices  $H$ ,  $G$ ,  $h$  of size  $(t \times n)$ ,  $(t \times p)$  and  $(t \times 1)$ , respectively, such that*

$$P_I(A, D, b) = \{(x, y) \in \mathbb{R}^{n+p} : Hx + Gy \leq h, x \geq 0, y \geq 0\}. \quad (10.12)$$

The linear inequality system (10.13) is a *complete formulation* of the problem (MIP) and thus the *mixed integer linear programming* problem has been reduced to a *linear programming* problem. The simplex algorithm provides a finite solution procedure for any linear program.

**10.2(c) (Existence of finite algorithms)** *If the data  $A$ ,  $D$ ,  $b$  of the problem (MIP) are rational, then*

$$\max\{cx + dy : (x, y) \in DM\} = \{cx + dy : Hx + Gy \leq h, x \geq 0, y \geq 0\}$$

for every  $(\mathbf{c}, \mathbf{d}) \in \mathbb{R}^{n+p}$ , where  $H, G, \mathbf{h}$  are defined in point 10.2(b). Thus there exists a finite algorithm for the resolution of any mixed integer linear program (MIP) with rational data and every such problem either has no feasible solution, or it has an unbounded optimum, or it has a finite optimum solution.

Another implication of point 10.2(a) is the following one. Denote by

$$C_\infty(\mathbf{A}, \mathbf{D}) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{Ax} + \mathbf{Dy} \leq \mathbf{0}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}\}$$

the asymptotic cone of the polyhedron  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  and likewise, by  $C_\infty^I$  the asymptotic cone of the polyhedron  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ . From relation (v) of the proof of point 10.2(a) we have the following observation.

**10.2(d)** For rational  $\mathbf{A}, \mathbf{D}, \mathbf{b}$  the asymptotic cone  $C_\infty^I$  of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  satisfies

$$C_\infty^I = C_\infty(\mathbf{A}, \mathbf{D}) \subseteq \mathbb{R}_+^{n+p}. \quad (10.13)$$

Suppose that  $P = P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  has a facet complexity of  $\phi \geq n + p + 1$ . By point 7.5(b)  $P$  has a finite generator  $(S = \{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^s, \mathbf{y}^s)\}, T = \{(\mathbf{r}^1, \mathbf{t}^1), \dots, (\mathbf{r}^q, \mathbf{t}^q)\})$  with  $\mathbf{r}^i \in \mathbb{Z}^n$  for  $1 \leq i \leq q$ , such that

$$\langle z_j \rangle \leq 4(n+1)(n+p)\phi \text{ for } 1 \leq j \leq n+p \text{ and every } z \in S \cup T.$$

**10.2(e)** (Facet/vertex complexity) If  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  has a facet complexity of  $\phi$ , then the polyhedron  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  has a finite generator  $(S_I, T_I)$  such that  $\langle z_j \rangle \leq 5(n+1)(n+p)\phi$  for all  $1 \leq j \leq n+p$  and  $z \in S_I \cup T_I$ , i.e. a vertex complexity of  $5(n+1)(n+p)^2\phi$ , and  $P_I$  has a facet complexity of  $20(n+1)(n+p)^4\phi$ .

Let us say that a formulation (MIP) has a facet complexity of  $\phi$  if the polyhedron  $P = P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  of (10.7) has a facet complexity of  $\phi$ . By point 10.2(e) (MIP) can be replaced by

$$(\text{MIP}_\phi) \quad \max\{\mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} : (\mathbf{x}, \mathbf{y}) \in P, 0 \leq x_j \leq 2^{5(n+1)(n+p)\phi}, x_j \in \mathbb{Z} \text{ for } 1 \leq j \leq n\}.$$

Since there are only a finite number of possible solutions for the integer variables  $\mathbf{x}$  and since the branch-and-bound algorithm selects only integer variables for branching, it follows that **branch-and-bound** gives a **finite** algorithm when applied to  $(\text{MIP}_\phi)$ .

We get a substantially deeper insight into the tractability of (MIP) by applying Remark 9.20.

**10.2(f)** (Polynomial-time solvability) Let  $P = P(\mathbf{A}, \mathbf{D}, \mathbf{b}) \subseteq \mathbb{R}^{n+p}$  be a formulation of a mixed-integer linear program (MIP) of facet complexity  $\phi$  and  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  be the convexification (10.9) of the discrete mixed set (10.8). The mixed integer linear program (MIP) can be solved in time that is bounded by a polynomial in  $n, p, \phi$  and  $\langle \mathbf{c} \rangle + \langle \mathbf{d} \rangle$  if and only if the polyhedral separation problem for  $P_I$  can be solved in time that is bounded by a polynomial in  $n, p, \phi$  and  $\langle z \rangle$  where  $z \in \mathbb{R}^{n+\phi}$  is any rational vector.

## 10.3 Extremal Characterizations of Ideal Formulations

Given a rational formulation  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  of a mixed integer linear program (MIP) let

$$(F1) \quad H\mathbf{x} + G\mathbf{y} \leq \mathbf{h}, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}$$

be *any* complete formulation of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  like in point 10.2(b). Like in Chapter 6.3 call an inequality *redundant* with respect to (FI) if dropping it does not change the corresponding set of feasible solutions. Given a complete formulation of (MIP), we can always find a minimal and complete formulation of it in a finite number of steps. Call a complete and minimal formulation of (MIP) an **ideal formulation** of (MIP). From Chapter 7.2.2 we know that an ideal formulation of (MIP) is *quasi-unique*.

The task to be done is to characterize ideal formulations of mixed-integer linear programs in a *constructive* manner. To this end define for any row vector  $(\mathbf{f}, \mathbf{g}, f_0) \in \mathbb{R}^{n+p+1}$  the following two concepts.

**Definition VE** An equation  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} = f_0$  is a valid equation for  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  if and only if  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} = f_0$  for all  $(\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ .

**Definition VI** An inequality  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  is a valid<sup>#</sup> inequality for  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  if and only if (i)  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  for all  $(\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  and (ii) there exists  $(\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  such that  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} < f_0$ .

Given an ideal formulation (FI) we test whether or not an inequality is a *valid equation* for  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  by solving two linear programs and thus – like in (7.1) – we can write every ideal formulation of (MIP) as

$$(FIM) \quad H_1\mathbf{x} + G_1\mathbf{y} = h_1, \quad H_2\mathbf{x} + G_2\mathbf{y} \leq h_2, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}$$

where  $H_1, G_1, h_1$  are rational matrices of size  $(t_1 \times n), (t_1 \times p), (t_1 \times 1)$  with  $0 \leq t_1 < \infty$  and  $H_2, G_2, h_2$  are rational matrices of size  $(t_2 \times n), (t_2 \times p), (t_2 \times 1)$  with  $0 \leq t_2 < \infty$ , respectively.

For notational convenience we have included *all* nonnegativity constraints  $\mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$  even though in principle some of the valid equations may imply that  $x_j = 0$  or  $y_k = 0$  for every  $(\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  – since we can detect such valid equations by testing the inequalities  $x_j \geq 0, y_k \geq 0$  we can assume for the analysis that such variables have been deleted from the problem.

To state a constructive characterization of ideal formulations for (MIP) denote like in the proof of point 10.2(a) by  $(\mathbf{x}^i, \mathbf{y}^i)$  for  $i = 1, \dots, s$  all the extreme points and by  $(\mathbf{r}^i, \mathbf{t}^i)$  for  $i = 1, \dots, q$  all the extreme rays of the polyhedron  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  defined in (10.7) where  $\mathbf{r}^i \in \mathbb{Z}^n$  are such that

$$g.c.d.(r_1^i, \dots, r_n^i) = 1 \text{ for } 1 \leq i \leq q.$$

for the components  $r_j^i$  of  $\mathbf{r}^i$ . Define  $XY$  like in (10.10) and let  $(\bar{\mathbf{x}}^i, \bar{\mathbf{y}}^i)$  for  $i = 1, \dots, K$  be the finitely many points of  $XY$  that we have constructed in the proof of point 10.2(a). Thus for every  $(\mathbf{x}, \mathbf{y}) \in P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$

$$(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K \delta_k (\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) + \sum_{i=1}^q \beta_i (\mathbf{r}^i, \mathbf{t}^i), \quad (10.14)$$

where  $\delta_k \geq 0, \sum_{k=1}^K \delta_k = 1$  and  $\beta_i \geq 0$ . The sets

$$S = \{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) \in \mathbb{R}^{n+p} : 1 \leq k \leq K\}, \quad T = \{(\mathbf{r}^i, \mathbf{t}^i) \in \mathbb{R}^{n+p} : 1 \leq i \leq q\}$$

form hence a *finite generator* of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ . All *extreme points* of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  are necessarily among the points  $(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)$  for  $k = 1, \dots, K$  and if they were known *a priori* then it would evidently

suffice to define  $S$  to be the set of the extreme points of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ . Define three matrices

$$\mathbf{W} = \begin{pmatrix} \bar{\mathbf{x}}^1 & \dots & \bar{\mathbf{x}}^K \\ \bar{\mathbf{y}}^1 & \dots & \bar{\mathbf{y}}^K \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} \bar{\mathbf{r}}^1 & \dots & \bar{\mathbf{r}}^q \\ \bar{\mathbf{t}}^1 & \dots & \bar{\mathbf{t}}^q \end{pmatrix}, \quad \widehat{\mathbf{W}} = \begin{pmatrix} \mathbf{W} & \mathbf{U} \\ -\mathbf{e} & \mathbf{0} \end{pmatrix},$$

where  $\mathbf{e} = (1, \dots, 1)$  is a row vector with  $K$  components equal to 1, and so  $\mathbf{W}$ ,  $\mathbf{U}$  are rational matrices of size  $(n+p) \times K$  and  $(n+p) \times q$ , respectively. The cone

$$C_I = \{(\boldsymbol{\pi}, f_0) \in \mathbb{R}^{n+p+1} : \boldsymbol{\pi}\mathbf{W} - \mathbf{e}f_0 \leq \mathbf{0}, \boldsymbol{\pi}\mathbf{U} \leq \mathbf{0}\} \quad (10.15)$$

is the  $f_0$ -polar of  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  – see (7.13). To simplify the notation we have set  $\boldsymbol{\pi} = (\mathbf{f}, \mathbf{g})$  for short and denote  $\mathbf{z} = (\mathbf{x}, \mathbf{y})$  so that  $\boldsymbol{\pi}\mathbf{z} = \mathbf{fx} + \mathbf{gy}$  where  $\boldsymbol{\pi}$  is a row vector and  $\mathbf{z}$  is a column vector of length  $n+p$ . Denote the lineality space of the cone  $C_I$  by

$$L_I = \{(\boldsymbol{\pi}, f_0) \in \mathbb{R}^{n+p+1} : \boldsymbol{\pi}\mathbf{W} - \mathbf{e}f_0 = \mathbf{0}, \boldsymbol{\pi}\mathbf{U} = \mathbf{0}\}.$$

- 10.3(a)** Let  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ . (i)  $\boldsymbol{\pi}\mathbf{z} \leq f_0$  is a valid inequality for  $P_I$  if and only if  $(\boldsymbol{\pi}, f_0) \in C_I$ .  
(ii)  $\boldsymbol{\pi}\mathbf{z} = f_0$  is a valid equation for  $P_I$  if and only if  $(\boldsymbol{\pi}, f_0) \in L_I$ .

Define  $d_I = \text{rank}(\widehat{\mathbf{W}}) - 1$  where  $d_I = -1$  if the matrix is empty and let  $(\mathbf{B}, \mathbf{b}) = (\mathbf{H}_1, \mathbf{G}_1, -\mathbf{h}_1)$  where the rows of  $(\mathbf{B}, \mathbf{b})$  correspond to a basis of  $L_I$ . Let  $L_I^\perp$  be the orthogonal complement of  $L_I$ ,

$$L_I^\perp = \{(\boldsymbol{\pi}, f_0) \in \mathbb{R}^{n+p+1} : \mathbf{B}\boldsymbol{\pi}^T + \mathbf{b}f_0 = \mathbf{0}\},$$

and  $C_I^0 = C_I \cap L_I^\perp$  be the pointed cone

$$C_I^0 = \{(\boldsymbol{\pi}, f_0) \in \mathbb{R}^{n+p+1} : \boldsymbol{\pi}\mathbf{W} - \mathbf{e}f_0 \leq \mathbf{0}, \boldsymbol{\pi}\mathbf{U} \leq \mathbf{0}, \mathbf{B}\boldsymbol{\pi}^T + \mathbf{b}f_0 = \mathbf{0}\}.$$

- 10.3(b)** Let  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ .

- (i)  $\dim P_I = d_I$ ,  $\dim L_I = n+p-d_I$  and  $\text{aff}(P_I) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{H}_1\mathbf{x} + \mathbf{G}_1\mathbf{y} = \mathbf{h}_1\}.$   
(ii) If  $(\boldsymbol{\pi}, f_0)$  is an extreme ray of  $C_I^0$  then  $\boldsymbol{\pi}\mathbf{z} \leq f_0$  defines a facet of  $P_I$ .  
(iii) If  $\boldsymbol{\pi}\mathbf{z} \leq f_0$  defines a facet of  $P_I$  then  $(\boldsymbol{\pi}, f_0) = (\boldsymbol{\pi}^*, f_0^*) + (\boldsymbol{\pi}^+, f_0^+)$  where  $(\boldsymbol{\pi}^*, f_0^*) \in L_I$  and  $(\boldsymbol{\pi}^+, f_0^+)$  is an extreme ray of  $C_I^0$ .

Like in Chapter 7.2.2 we can summarize the preceding as follows.

- 10.3(c)** (Ideal formulations) Let  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  be any rational formulation of a mixed integer linear program (MIP). A formulation (FIM) for (MIP) is ideal if and only if  $\mathbf{H}_1\mathbf{x} + \mathbf{G}_1\mathbf{y} = \mathbf{h}_1$  is a complete system of valid equations of full row rank of the affine hull and  $\mathbf{x} \geq \mathbf{0}$ ,  $\mathbf{y} \geq \mathbf{0}$ ,  $\mathbf{H}_2\mathbf{x} + \mathbf{G}_2\mathbf{y} \leq \mathbf{h}_2$  is a minimal and complete system of facet defining inequalities of the polyhedron  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ .

To obtain an ideal formulation of a mixed integer linear program (MIP) we can proceed as follows:

- Starting from a finite generator of  $P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  we determine  $XY$  and from it the finite set  $X$ .
- For each  $\mathbf{x} \in X$  we determine the extreme points of  $\Lambda_{\mathbf{x}}$  to construct  $\mathbf{W}$  and  $\mathbf{U}$  that define  $C_I$ .
- We determine a basis for the lineality space  $L_I$  of  $C_I$  and a full system of extreme rays of  $C_I^0$ .

The cone  $C_I$  furnishes an **extremal characterization** of an ideal linear description of the polyhedron  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  for the mixed-integer program (MIP). We give next two different necessary and sufficient conditions for a valid<sup>#</sup> inequality for  $P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  to define a facet.

**10.3(d) (Characterization of facets)** Let  $F$  be a nonempty face of  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$ ,  $(\mathbf{H}_1 \mathbf{G}_1 \mathbf{h}_1)$  be a matrix of full rank and of size  $t_1 \times (n+p+1)$  with  $0 \leq t_1 < \infty$  such that  $\text{aff}(P_I) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+p} : \mathbf{H}_1 \mathbf{x} + \mathbf{G}_1 \mathbf{y} = \mathbf{h}_1\}$  and let  $d_I = \dim P_I = n+p-t_1$ . Then the following statements are equivalent:

- (i)  $F$  is a facet of  $P_I$ , i.e.  $\dim F = d_I - 1$ .
- (ii) There exist a valid<sup>#</sup> inequality  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  for  $P_I$  such that  $F = \{(\mathbf{x}, \mathbf{y}) \in P_I : \mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} = f_0\}$  and  $d_I$  affinely independent points  $(\mathbf{x}^i, \mathbf{y}^i) \in P_I$  satisfying  $\mathbf{f}\mathbf{x}^i + \mathbf{g}\mathbf{y}^i = f_0$  for  $i = 1, \dots, d_I$ .
- (iii) There exists a valid<sup>#</sup> inequality  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  for  $P_I$  such that  $F = \{(\mathbf{x}, \mathbf{y}) \in P_I : \mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} = f_0\}$  satisfying the property: if  $\mathbf{f}'\mathbf{x} + \mathbf{g}'\mathbf{y} \leq f'_0$  is any valid inequality for  $P_I$  such that  $F \subseteq \{(\mathbf{x}, \mathbf{y}) \in P_I : \mathbf{f}'\mathbf{x} + \mathbf{g}'\mathbf{y} = f'_0\}$  then there exist a scalar  $\alpha \geq 0$  and a vector  $\boldsymbol{\lambda} \in \mathbb{R}^{t_1}$  such that

$$(\mathbf{f}', \mathbf{g}') = \alpha(\mathbf{f}, \mathbf{g}) + \boldsymbol{\lambda}(\mathbf{H}_1, \mathbf{G}_1), \quad f'_0 \geq \alpha f_0 + \boldsymbol{\lambda} \mathbf{h}_1. \quad (10.16)$$

Point 10.3(b)(ii) gives a *direct method* for verifying or falsifying that a valid<sup>#</sup> inequality  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  for  $P_I = P_I(\mathbf{A}, \mathbf{D}, \mathbf{b})$  defines a facet if the dimension  $d_I$  of  $P_I$  is known: all one has to do is to produce a list of  $d_I$  affinely independent points in  $P_I$  or, if  $f_0 \neq 0$  equivalently, a list of  $d_I$  linearly independent points of  $P_I$  that satisfy the inequality as an equation. To decide whether or not an inequality defines a facet of  $P_I$  one determines the rank of the matrix given by this list of points.

Point 10.3(b)(iii) gives an *indirect method* for verifying or falsifying that a valid<sup>#</sup> inequality  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} \leq f_0$  for  $P_I$  defines a facet: here we need to know first of all the matrix that defines the affine hull of  $P_I$ , which may, however, be empty. The indirect proof method then proceeds by assuming that the inequality does not define a facet of  $P_I$ . Consequently, we have the existence of some valid inequality  $\mathbf{f}'\mathbf{x} + \mathbf{g}'\mathbf{y} \leq f'_0$  satisfying the stated property of point 10.3(d) (iii). Utilizing the knowledge about the discrete mixed set  $DM$ , i.e. the *structure* of the underlying problem, one then verifies that (10.16) holds. The way this is done consists of constructing points  $(\mathbf{x}, \mathbf{y})$  of  $P_I$  that satisfy  $\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{y} = f_0$  and thus by implication  $\mathbf{f}'\mathbf{x} + \mathbf{g}'\mathbf{y} = f'_0$  as well. Since we know all of the coefficients  $(\mathbf{f}, \mathbf{g}, f_0)$  explicitly we can choose suitable points  $(\mathbf{x}, \mathbf{y}) \in P_I$  and then determine successively all components of  $(\mathbf{f}', \mathbf{g}', f'_0)$  by taking differences in order to verify (10.16).

Point 10.3(b)(iii) also shows that ideal formulations are *essentially unique*: if  $t_1 = 0$ , i.e. the dimension of  $P_I$  is *full*, then we have uniqueness of the system of linear inequalities that define the facets of  $P_I$  up to *multiplication* by a positive scalars, i.e. up to the scaling of the coefficients. If  $t_1 > 0$ , then we have in essence the same *modulo* linear combinations of the equations that define the affine hull of  $P_I$ .

## 10.4 Polyhedra with the Integrality Property

Recall that one of the motivations for the approach we have taken to solving (MIP) is the following observation: if the relaxed linear program ( $\text{MIP}_{LP}$ ) produces an optimal solution  $(\mathbf{x}, \mathbf{y}) \in P(\mathbf{A}, \mathbf{D}, \mathbf{b})$  such that  $\mathbf{x} \in \mathbb{Z}^n$ , then we have *a posteriori* an optimal solution for the problem (MIP). The question that comes to one's mind is: when does a rational formulation guarantee such an outcome?

As usual such questions can be made precise in more than one way. One way is to "fix" the data  $\mathbf{A}, \mathbf{D}, \mathbf{b}$  and to ask for a guarantee that the optimum solution  $(\mathbf{x}, \mathbf{y})$  of ( $\text{MIP}_{LP}$ ) – if it exists –

has the property that  $x \in \mathbb{Z}^n$  no matter what objective function  $cx + dy$  is maximized. We shall say that problem (MIP) has the **integrality property** whenever this is the case. The answer to the question is clear: (MIP) has the integrality property if and only if  $P(A, D, b)$  is an *ideal* formulation for (MIP).

**10.4(a)** Let  $P = P(A, D, b)$  be a formulation of facet complexity  $\phi$  of a mixed integer linear program (MIP), let  $C_\infty$  be the asymptotic cone of  $P$  and denote by  $z_{LP}^{MIP}$  the objective function value of  $(\text{MIP}_{LP})$ . The formulation  $P$  is ideal if and only if for every  $c \in \mathbb{Z}^n$  and every  $d \in \mathbb{Z}^p$  such that  $-\infty < z_{LP}^{MIP} < +\infty$  we have  $cx \in \mathbb{Z}$  for every optimal solution  $(x, y)$  of  $(\text{MIP}_{LP})$ .

If  $D$  is nonvoid, i.e. if there are flow variables in the problem (MIP), then point 10.4(a) does not tell us much. But if  $D$  is void, then we have a pure integer problem and we can analyze e.g. the dual linear program to decide whether or not the (primal) formulation is ideal. So the criterion of point 10.4(a) is worth knowing and in certain cases it makes the analysis considerably easier.

A second way to make the question concerning the integrality property of (MIP) precise is to restrict it further: we now ask for a characterization of matrices  $A$  and  $D$  such that the optimum solution  $(x, y)$  of  $(\text{MIP}_{LP})$  – provided it exists – satisfies  $x \in \mathbb{Z}^n$  no matter what right-hand side vector  $b$  and objective function  $cx + dy$  are used in (MIP).

In this context it does not make sense to permit nonintegral rational data in the matrix  $A$ . To see this let  $I \subseteq \{1, \dots, n\}$  be any subset such that the columns  $a_j$  of  $A$  are linearly independent for  $j \in I$ . Then setting e.g.  $b = \frac{1}{2} \sum_{j \in I} a_j$  we get a rational right-hand side vector for which  $P(A, D, b)$  has an extreme point  $(x^*, y^*)$  with  $x^* \notin \mathbb{Z}^n$ . Consequently, for some objective function  $cx + dy$  the solution to  $(\text{MIP}_{LP})$  will not solve the problem (MIP).

So assume that the data of  $A$  are integer. If  $D$  is void then a complete answer is known.

**Definition TU** A matrix  $A$  is called *totally unimodular* if and only if every square submatrix of  $A$  has a determinant equal to 0 or  $\pm 1$ .

In particular, all elements of  $A$  must equal 0 or  $\pm 1$ . We drop the matrix  $D$  from the definitions of the polyhedra if  $D$  is void.

**10.4(b)** Let  $A$  be any  $m \times n$  matrix of integers. Then  $P(A, b) = P_I(A, b)$  for all  $b \in \mathbb{Z}^m$  if and only if  $A$  is totally unimodular.

To give point 10.4(b) a more general form let  $Q(A, b^1, b^2, d^1, d^2) = \{x \in \mathbb{R}^n : b^1 \leq Ax \leq b^2, d^1 \leq x \leq d^2\}$  and  $Q_I(A, b^1, b^2, d^1, d^2) = \text{conv}(Q(A, b^1, b^2, d^1, d^2) \cap \mathbb{Z}^n)$ , where  $b^i \in \mathbb{Z}^m$ ,  $d^i \in \mathbb{Z}^n$  for  $i = 1, 2$ .

**10.4(c)** An  $m \times n$  matrix  $A$  of integers is totally unimodular if and only if

$$Q(A, b^1, b^2, d^1, d^2) = Q_I(A, b^1, b^2, d^1, d^2)$$

for all  $b^i \in \mathbb{Z}^m$  and  $d^i \in \mathbb{Z}^n$  where  $i = 1, 2$ .

Totally unimodular matrices give rise to “easy” formulations of pure integer programs and there are other classes of matrices with this property.

**Definition PI** Let  $A$  be an  $m \times n$  matrix of zeros and ones,  $e^T = (1, \dots, 1) \in \mathbb{R}^m$  have  $m$  components equal to 1 and  $P(A, b)$ ,  $P_I(A, b)$  be defined as in (10.18).

**(i)**  $A$  is called *perfect* if and only if  $P(A, e) = P_I(A, e)$ .

**(ii)**  $A$  is called *ideal* if and only if  $P(-A, -e) = P_I(-A, -e)$ .

- (iii)  $A$  is called balanced if and only if  $P(A', e') = P_I(A', e')$  for all submatrices  $A'$  of  $A$  and compatibly dimensioned subvectors  $e'$  of  $e$ .

By point 10.4(b) every totally unimodular zero-one matrix  $A$  is perfect, ideal and balanced, but the converse is not true. Moreover, every balanced matrix is both perfect and ideal. The class of zero-one matrices that are *both* perfect and ideal contains the class of balanced matrices properly, see Exercise 10.5. It is an open problem to characterize matrices that are both perfect and ideal.

Like in the case of totally unimodular matrices one knows a lot about these matrices; in particular, their characterizations by way of *forbidden* submatrices are known, i.e. by those matrices which are in a sense the “smallest” matrices for which the defining integrality property is lost.

## 10.5 Exercises

---

### Exercise 10.1

Show that the rank  $r(H_1, G_1) = t_1$  in an ideal formulation (FIM). Show that every inequality  $H_2x + G_2y \leq h_2$  of (FIM) is a valid<sup>#</sup> inequality for  $P_I(A, D, b)$ .

---

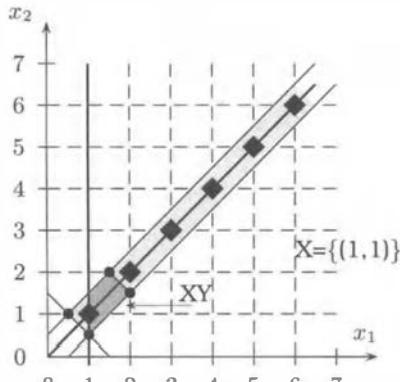
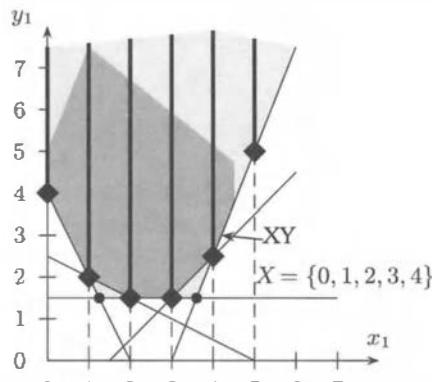
First we notice that  $t_1 \leq n+p$  since otherwise the equation system has either redundant equalities, which contradicts the assumption that (FIM) is ideal, or, it is infeasible, which means that (MIP) is infeasible. But if  $t_1 \leq n+p$  and  $r(H_1, G_1) < t_1$  again this means that some of the rows of the equation system either are linearly dependent or they yield infeasibility.

Consider now an inequality  $h_2^i x + g_2^i y \leq h_2^i$  which is clearly satisfied by all points in  $P_I(A, D, b)$ , i.e. condition (i) for the inequality to be valid<sup>#</sup> is satisfied. Suppose that condition (ii) is not satisfied, i.e. that there exists no  $(x, y) \in P_I(A, D, b)$  such that  $h_2^i x + g_2^i y < h_2^i$ . It follows that  $h_2^i x + g_2^i y = h_2^i$  for all  $(x, y) \in P_I(A, D, b)$  and thus by definition the inequality is implied by the system  $H_1x + G_1y = h_1$ , which contradicts the minimality of the ideal formulation.

---

### Exercise 10.2

- (i) Use the constructions of the last two sections to show algebraically that an ideal formulation for  $DM = \{x \in \mathbb{R}^2 : x_1 \geq 0, x_2 \geq 0, x_1 \text{ and } x_2 \text{ integer}, -2x_1 + 2x_2 \leq 1, 2x_1 - 2x_2 \leq 1 \text{ and } -2x_1 - 2x_2 \leq -3\}$  is given by  $x_1 - x_2 = 0, -x_1 \leq -1$ . (Hint: Use the double description algorithm.)
- (ii) Apply the same technique to find an ideal formulation for  $DM = \{(x_1, y_1) \in \mathbb{R}^2 : x_1 \geq 0 \text{ and integer}, y_1 \geq 0, -2x_1 - y_1 \leq -4, 5x_1 - 2y_1 \leq 15, -2y_1 \leq -3\}$ .
- (iii) Describe a method to find a linear description of the polytope  $XY$  given by (10.10).
- (iv) Using the double description algorithm calculate the linear description of the polytopes  $XY$  given by (10.10) for the examples of (MIP) of parts (i) and (ii).

**Fig. 10.4.** Geometry for 10.2(i)**Fig. 10.5.** Geometry for 10.2(ii)

**(i)** First we calculate the set  $XY$ . Using the DDA we calculate the extreme points and extreme rays of the polyhedron

$$P = \{x \in \mathbb{R}^2 : x_1 \geq 0, x_2 \geq 0, -2x_1 + 2x_2 \leq 1, 2x_1 - 2x_2 \leq 1, -2x_1 - 2x_2 \leq -3\}.$$

We find that  $P$  has two extreme points  $x^1 = (\frac{1}{2}, 1)$  and  $x^2 = (1, \frac{1}{2})$ , and one extreme ray  $r^1 = (1, 1)$ . Therefore we have that

$$XY = \{x \in \mathbb{R}^2 : x = (\frac{1}{2}\mu_1 + \mu_2 + \lambda, \mu_1 + \frac{1}{2}\mu_2 + \lambda), \mu_1 \geq 0, \mu_2 \geq 0, \mu_1 + \mu_2 = 1, 0 \leq \lambda \leq 1\}$$

which we can equivalently write as follows

$$XY = \{x \in \mathbb{R}^2 : x = (1 + \lambda - \frac{\mu}{2}, \frac{1}{2} + \lambda + \frac{\mu}{2}), 0 \leq \mu, \lambda \leq 1\}.$$

Next we calculate the set  $X = \{x \in \mathbb{Z}^n : x \in XY\}$ . From  $XY$  we know that if  $x \in XY$  then  $\frac{1}{2}e \leq x \leq 2e$ , where  $e$  is the vector in  $\mathbb{R}^2$  with two ones. So there are only four candidates for the set  $X$ , namely,  $(1, 1)$ ,  $(2, 1)$ ,  $(1, 2)$ ,  $(2, 2)$ . For each of them we solve the system of equations  $x_1 = 1 + \lambda - \frac{\mu}{2}$  and  $x_2 = \frac{1}{2} + \lambda + \frac{\mu}{2}$  to find the following values for  $(\lambda, \mu)$  respectively:  $(\frac{1}{4}, \frac{1}{2})$ ,  $(\frac{3}{4}, -\frac{1}{2})$ ,  $(\frac{3}{4}, \frac{3}{2})$ ,  $(\frac{5}{4}, \frac{1}{2})$ . Thus only the point  $(1, 1)$  is feasible and thus  $X = \{(1, 1)\}$  in this case.

Since we have a pure integer set we do not have to find values  $\hat{y}$  and thus there is no need to calculate the extreme points of  $A_x$ . Now we can write down the cone  $C_I$  as follows

$$C_I = \{(\pi_1, \pi_2, f_0) : \pi_1 + \pi_2 - f_0 \leq 0, \pi_1 - \pi_2 \leq 0\}$$

for which we calculate (using DDA) the extreme rays  $(1, -1, 0)$ ,  $(-1, 1, 0)$ ,  $(0, 0, 1)$ ,  $(0, -1, -1)$ , corresponding to the inequalities  $x_1 - x_2 \leq 0$ ,  $-x_1 + x_2 \leq 0$ ,  $0 \leq 1$ , and  $-x_2 \leq -1$  respectively. These inequalities give the following ideal formulation for  $P$ ,  $P = \{x \in \mathbb{R}^2 : x_1 - x_2 = 0, -x_2 \leq -1\}$ . Notice that because of symmetry one can replace the ray  $(0, -1, -1)$  by  $(-1, 0, -1)$  to get in turn the inequality  $-x_1 \leq -1$  instead of the  $-x_2 \leq -1$ .

**(ii)** We apply the same procedure as in part (i) with the difference that here we have a mixed integer set. Let

$$P = \{(x, y) \in \mathbb{R}^2 : x \geq 0, y \geq 0, -2x - y \leq -4, 5x - 2y \leq 15, -2y \leq -3\}.$$

Using DDA we calculate the extreme rays and extreme points of  $P$  and find that  $P$  has 3 extreme points,  $(\frac{5}{4}, \frac{3}{2})$ ,  $(\frac{18}{5}, \frac{3}{2})$ ,  $(0, 4)$ , and two extreme rays  $(0, 1)$  and  $(1, \frac{5}{2})$ . (N.B.: Notice that the extreme rays must be scaled such that the greatest common divisor among the components of the integer vector is 1. In our case the integer vector has only one component and thus if it is nonzero as in the second ray it has to be 1.) Now we can write the set

$$\begin{aligned} XY = \{(x, y) \in \mathbb{R}^2 : (x, y) = & (\frac{5}{4}\mu_1 + \frac{18}{5}\mu_2 + \lambda_2, \frac{3}{2}\mu_1 + \frac{3}{2}\mu_2 + 4\mu_3 + \lambda_1 + \frac{5}{2}\lambda_2), \\ & \mu_1 + \mu_2 + \mu_3 = 1, \mu_1, \mu_2, \mu_3 \geq 0, 0 \leq \lambda_1, \lambda_2 \leq 1\} \end{aligned}$$

and calculate the set  $X = \{x \in \mathbb{Z} : \exists y \text{ such that } (x, y) \in XY\}$ . From the description of  $XY$  we know that  $x \leq \frac{18}{5} + 1 = \frac{23}{5}$  and thus  $X \subseteq \{0, 1, 2, 3, 4\}$ . One verifies by e.g. drawing a picture that in fact  $X = \{0, 1, 2, 3, 4\}$ . A precise way to find the set  $X$  will be discussed in part (iii). Now we have to calculate the extreme points of the sets

$$\Lambda_x = \{\boldsymbol{\mu}, \boldsymbol{\lambda} \in \mathbb{R}^{3+2} : x = \frac{5}{4}\mu_1 + \frac{18}{5}\mu_2 + \lambda_2, \mu_1 + \mu_2 + \mu_3 = 1, \mu_1, \mu_2, \mu_3 \geq 0, 0 \leq \lambda_1, \lambda_2 \leq 1\}.$$

Using DDA we calculate:

$x$	extreme points of $\Lambda_x$
0	$(0, 0, 1, 0, 0), (0, 0, 1, 1, 0)$
1	$(0, 5/18, 13/18, 0, 0), (4/5, 0, 1/5, 0, 0), (0, 5/18, 13/18, 1, 0)$ $(4/5, 0, 1/5, 1, 0), (0, 0, 1, 0, 1), (0, 0, 1, 1, 1)$
2	$(0, 5/9, 4/9, 0, 0), (32/47, 15/47, 0, 0, 0), (0, 5/18, 13/18, 0, 1)$ $(0, 5/9, 4/9, 1, 0), (32/47, 15/47, 0, 1, 0), (4/5, 0, 1/5, 0, 1)$ $(1, 0, 0, 0, 3/4), (0, 5/18, 13/18, 1, 1), (4/5, 0, 1/5, 1, 1)$ $(1, 0, 0, 1, 3/4)$
3	$(0, 5/6, 1/6, 0, 0), (12/47, 35/47, 0, 0, 0), (0, 5/9, 4/9, 0, 1)$ $(0, 5/6, 1/6, 1, 0), (12/47, 35/47, 0, 1, 0), (32/47, 15/47, 0, 0, 1)$ $(0, 5/9, 4/9, 1, 1), (32/47, 15/47, 0, 1, 1)$
4	$(0, 5/6, 1/6, 0, 1), (0, 1, 0, 0, 2/5), (12/47, 35/47, 0, 0, 1)$ $(0, 5/6, 1/6, 1, 1), (0, 1, 0, 1, 2/5), (12/47, 35/47, 0, 1, 1)$

Now we calculate the values of

$$\hat{y} = \frac{3}{2}\mu_1 + \frac{3}{2}\mu_2 + 4\mu_3 + \lambda_1 + \frac{5}{2}\lambda_2$$

for each  $(\mu_1, \mu_2, \mu_3, \lambda_1, \lambda_2)$  given in the table above. Having done so, we can write down the matrix  $W$

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 5 & \frac{67}{18} & 2 & \frac{85}{18} & 3 & 5 & \frac{15}{2} & \frac{31}{9} & \frac{3}{2} & \frac{67}{18} & \frac{31}{9} & \frac{5}{2} & \frac{9}{2} & \frac{27}{8} & \frac{130}{18} & \frac{11}{2} & \frac{35}{8} & \frac{23}{12} & \frac{3}{2} & \frac{107}{18} & \frac{35}{12} & \frac{5}{2} & 4 & \frac{125}{18} & 5 & \frac{53}{12} & \frac{5}{2} & 4 & \frac{65}{12} & \frac{7}{2} & 5 \end{pmatrix}$$

and compute the extreme rays of the cone  $C_I = \{(\boldsymbol{\pi}, f_0) \in \mathbb{R}^3 : \boldsymbol{\pi}\mathbf{W} - e f_0 \leq \mathbf{0}, \boldsymbol{\pi}\mathbf{U} \leq \mathbf{0}\}$  where  $\mathbf{U} = \begin{pmatrix} 0 & 1 \\ 1 & \frac{5}{2} \end{pmatrix}$ . Using DDA we find that the extreme rays of  $C_I$  are  $(0, 0, 1)$ ,  $(2, -2, 3)$ ,  $(-1, 0, 0)$ ,  $(-1, -2, -5)$ ,  $(5, -2, 15)$ ,  $(0, -2, -3)$ ,  $(-2, -1, -4)$ , corresponding to the inequalities  $0 \leq 1$ ,  $2x - 2y \leq 3$ ,  $-x \leq 0$ ,  $-x - 2y \leq -5$ ,  $5x - 2y \leq 15$ ,  $-2y \leq -3$ ,  $-2x - y \leq -4$ , respectively. Thus an ideal description of  $P$  is

$$P = \{(x, y) : 2x - 2y \leq 3, -x \leq 0, -x - 2y \leq -5, 5x - 2y \leq 15, -2y \leq -3, -2x - y \leq -4\}.$$

**(iii)** The difference between the set  $XY$  and the polyhedron  $P$  is that  $XY$  is bounded, by restricting the values  $\lambda_i$  to be between 0 and 1. Therefore, to get a description of  $XY$  it suffices to find the convex hull of the points  $(\mathbf{x}^i, \mathbf{y}^i) + \sum_{j=1}^q \lambda_j (\mathbf{r}^j, \mathbf{t}^j)$  for all  $1 \leq i \leq s$ , where we consider all possible combinations of 0-1 values for  $\lambda_j$ . Here  $(\mathbf{x}^i, \mathbf{y}^i)$  for  $i = 1, \dots, s$  are the extreme points of  $P$  and  $(\mathbf{r}^j, \mathbf{t}^j)$  for  $j = 1, \dots, q$  are its extreme rays. One can use DDA to get a linear description of the convex hull of the points. Having the linear description of  $XY$ , we can calculate the set  $X$  by projecting out the  $y_k$  variables.

**(iv)** For the polyhedron of part (i) we have two extreme points  $\mathbf{x}^1 = (\frac{1}{2}, 1)$  and  $\mathbf{x}^2 = (1, \frac{1}{2})$ , and one extreme ray  $\mathbf{r}^1 = (1, 1)$ . Therefore applying the method described in part (iii) we have to find the convex hull of the following points:  $(1/2, 1)$ ,  $(3/2, 2)$ ,  $(1, 1/2)$ ,  $(2, 3/2)$ . Using DDA we find that the linear description of the set  $XY$  is

$$XY = \{(x_1, x_2) : -2x_1 - 2x_2 \leq -3, -2x_1 + 2x_2 \leq 1, 2x_1 - 2x_2 \leq 1, 2x_1 + 2x_2 \leq 7\}$$

Similarly for the polyhedron of part (ii) we have 3 extreme points,  $(\frac{5}{4}, \frac{3}{2})$ ,  $(\frac{18}{5}, \frac{3}{2})$ ,  $(0, 4)$ , and two extreme rays  $(0, 1)$  and  $(1, \frac{5}{2})$ . Thus we have to find the convex hull of the following points:

$$(5/4, 3/2), (5/4, 5/2), (9/4, 4), (9/4, 5), (18/5, 3/2), (18/5, 5/2), (23/5, 4), (23/5, 5), (0, 4), (0, 5), (1, 13/2), (1, 15/2).$$

Using DDA we find that the linear description of the convex hull of these points and thus of  $XY$  is

$$XY = \{(x, y) : -2x - y \leq -4, -2y \leq -3, -x \leq 0, -5x + 2y \leq 10, 5x - 2y \leq 15, 5x \leq 23, 25x + 36y \leq 295\}.$$

### \*Exercise 10.3

- (i) Using the direct method prove that every inequality of the formulation  $(F_2)$  of the example of Chapter 10.2 defines a facet of its convex hull.

- (ii) Do the same using the indirect method of proof.
  - (iii) Prove that the integrality requirement can be dropped, i.e. that  $(F_2)$  is an ideal formulation of this simple problem.
  - (iv) Prove that formulation  $(F_1)$  is worse than  $(F_2)$ .
- 

Let  $P$  be the polytope

$$P = \{x \in \mathbb{R}^{n+1} : x_j \leq x_{n+1} \text{ for } j = 1, \dots, n, x_{n+1} \leq 1, x \geq 0\}.$$

**(i)** Let  $\mathbf{u}^i \in \mathbb{R}^{n+1}$  be the  $i$ -th unit vector. Since the  $n+2$  affinely independent vectors  $\mathbf{u}^i + \mathbf{u}^{n+1}$ , for  $i = 1, \dots, n$ ,  $\mathbf{u}^{n+1}$  and  $\mathbf{0}$  are in  $P$ , the polytope is full dimensional, i.e.  $\dim P = n+1$ . First we show that the nonnegativity inequalities are facet defining. Consider the inequality  $x_k \geq 0$  where  $1 \leq k \leq n$ . The points  $\mathbf{u}^{n+1}$ ,  $\mathbf{u}^i + \mathbf{u}^{n+1}$ ,  $i = 1, \dots, n+1$ ,  $i \neq k$ , and the zero vector lie on the face  $F_k = \{x \in P : x_k = 0\}$  and thus  $\dim F = n = \dim P - 1$ . To prove that the inequality  $x_k \leq x_{n+1}$  for some  $1 \leq k \leq n$  defines a facet we note that the vectors  $\mathbf{u}^i + \mathbf{u}^k + \mathbf{u}^{n+1}$ , for  $i = 1, \dots, n$ ,  $i \neq k$ ,  $\mathbf{u}^k + \mathbf{u}^{n+1}$  lie on the face  $F_k = \{x \in P : x_k = x_{n+1}\}$ . The matrix formed with rows  $\mathbf{u}^i + \mathbf{u}^k + \mathbf{u}^{n+1}$ , for  $i = 1, \dots, k-1$ ,  $\mathbf{u}^k + \mathbf{u}^{n+1}$ ,  $\mathbf{u}^j + \mathbf{u}^k + \mathbf{u}^{n+1}$  for  $j = k+1, \dots, n$  has a rank of  $n$  and thus the  $n$  vectors together with the zero vector which is also in  $F$  are  $n+1$  affinely independent points in  $F$ . It follows that  $\dim F_k = n = \dim P - 1$  and thus the inequality  $x_k \leq x_{n+1}$  is facet defining. For the inequality  $x_{n+1} \leq 1$  define  $F = \{x \in P : x_{n+1} = 1\}$ . The vectors  $\mathbf{u}^i + \mathbf{u}^{n+1}$ ,  $i = 1, \dots, n$  and  $\mathbf{u}^{n+1}$  lie on  $F$  and are affinely independent and thus  $x_{n+1} \leq 1$  is facet defining. Note that  $x_j \leq 1$  for  $j = 1, \dots, n$  is not facet defining since it is the sum of  $x_j \leq x_{n+1}$  and  $x_{n+1} \leq 1$ .

**(ii)** For the nonnegativity inequality  $x_k \geq 0$  the indirect proof proceeds as follows. Suppose that  $F_k = \{x \in P : x_k = 0\}$  is not a facet, and let  $b\mathbf{x} \leq b_0$  be a facet defining inequality such that  $F_b = \{x \in P : b\mathbf{x} = b_0\} \supset F_k$ . Since  $\mathbf{0} \in F_k \subset F_b$ , we have  $b_0 = 0$ . Since the vectors  $\mathbf{u}^i + \mathbf{u}^{n+1}$ ,  $1 \leq i \leq n+1$ ,  $i \neq k$  are in  $F_k$  and thus in  $F_b$ , it follows that  $b_i + b_{n+1} = 0$  for all  $1 \leq i \leq n+1$ ,  $i \neq k$ , and since  $\mathbf{u}^{n+1} \in F_k \subset F_b$  we have  $b_{n+1} = 0$ . Hence,  $b = \alpha\mathbf{u}^k$  and  $\alpha < 0$  since  $\mathbf{u}^k + \mathbf{u}^{n+1} \in P$  which contradicts the assumption that  $F_k$  is not facet.

To prove that  $x_j \leq x_{n+1}$  is facet defining we proceed as follows. Suppose that  $F_j = \{x \in P : x_j - x_{n+1} = 0\}$  is not facet and let  $b\mathbf{x} \leq b_0$  be such that  $F_b = \{x \in P : b\mathbf{x} = b_0\} \supset F_j$ . Since  $\mathbf{0} \in F_j \subset F_b$  we have that  $b_0 = 0$ . Since  $\mathbf{u}^i + \mathbf{u}^j + \mathbf{u}^{n+1}$ ,  $j \neq i \neq n+1$  is in  $F_j \subset F_b$  we have  $b_i + b_j + b_{n+1} = 0$  for all  $1 \leq i \leq n$ ,  $i \neq j$ . Since  $\mathbf{u}^j + \mathbf{u}^{n+1} \in F_j \subset F_b$  we have  $b_j + b_{n+1} = 0$  and thus  $b_i = 0$  for all  $1 \leq i \neq j \leq n$ . Thus  $x_j \leq x_{n+1}$  is a scalar multiple of  $b\mathbf{x} \leq b_0$  which contradicts the assumption that  $F_j$  is not a facet of  $P$ .

Finally to show that  $x_{n+1} \leq 1$  defines a facet, suppose that  $F = \{x \in P : x_{n+1} = 1\}$  is not a facet and let  $b\mathbf{x} \leq b_0$  be such that  $F_b = \{x \in P : b\mathbf{x} = b_0\} \supset F$ . Since  $\mathbf{u}^{n+1} \in F \subset F_b$  we have that  $b_{n+1} = b_0$ . Since  $\mathbf{u}^i + \mathbf{u}^{n+1} \in F \subset F_b$  we have  $b_i + b_{n+1} = b_0$  but since  $b_{n+1} = b_0$ ,  $b_i = 0$  for all  $1 \leq i \leq n$ . So again  $b\mathbf{x} \leq b_0$  is a multiple of  $x_{n+1} \leq 1$  which contradicts the assumption that  $F$  is not a facet.

**(iii)** It suffices to show that the polytope has only integral extreme points. From point 7.2(b)  $x^0$  is an extreme point of a polyhedron in  $\mathbb{R}^n$  if and only if it is feasible and the matrix of the constraints it satisfies at equality has rank  $n$ . The constraint matrix in our case has  $2n+1$  rows and the

dimension of  $P$  is  $n + 1$ . Let  $\mathbf{x}^0$  be an extreme point of  $P$ . If  $x_{n+1}^0 = 0$  then  $\mathbf{x}^0 = \mathbf{0}$ . Clearly  $\mathbf{x}^0$  is an extreme point and is integral. We next prove that in all other cases  $x_{n+1}^0 = 1$ . First observe that for each  $1 \leq j \leq n$  at most one of the inequalities  $x_j \geq 0$  and  $x_j \leq x_{n+1}$  can be satisfied at equality, since  $x_{n+1} > 0$ , and thus at most  $n$  of the  $2n$  inequalities  $x_j \geq 0$ ,  $x_j \leq x_{n+1}$ ,  $1 \leq j \leq n$  can be satisfied at equality by some feasible point  $\mathbf{x}$ . Suppose that  $0 < x_{n+1}^0 < 1$ . Then from the feasibility of  $\mathbf{x}^0$  and the inequalities  $x_j \leq x_{n+1}$  we have  $x_j < 1$  for all  $1 \leq j \leq n$ . So there are at most  $n$  inequalities satisfied at equality which means that the rank of the corresponding matrix is at most  $n$  contradicting the assumption that  $\mathbf{x}^0$  is an extreme point of  $P$ . Thus  $x_{n+1}^0 = 1$  in every extreme point of  $P$  other than 0. Now suppose that there exists  $1 \leq k \leq n$  such that  $0 < x_k < 1$ . Then none of the inequalities  $x_k \leq x_{n+1}$  and  $x_k \geq 0$  is satisfied at equality, and thus there are at most  $n$  inequalities ( $n - 1$  corresponding to variables  $x_j$ ,  $1 \leq j \leq n$ ,  $j \neq k$ , and one corresponding to the constraint  $x_{n+1} \leq 1$ ) satisfied at equality which like before gives a contradiction. Thus every extreme point of  $P$  is integral, and the description is ideal.

**(iv)** The point  $x_1 = 1$ ,  $x_j = 0$  for  $j = 2, \dots, n$  and  $x_{n+1} = \frac{1}{K}$  is feasible for the linear relaxation of  $(F_1)$ , but not for the linear relaxation of  $(F_2)$ , which proves the point.

---

### Exercise 10.4

Consider again the Berlin airlift model of Chapter 10.1 and the relations (10.3), ..., (10.6).

(i) Let  $\gamma_j^i = \alpha^{j+1-i}(\lfloor \delta_{i+1}/(\alpha - 1) \rfloor - \delta_i)$  if  $\alpha\delta_i \geq (\alpha - 1)\lfloor \delta_{i+1}/(\alpha - 1) \rfloor > (\alpha - 1)\delta_i$ ,  $\gamma_j^i = 0$  otherwise. Show that every integer solution to (10.3) satisfies the inequalities

$$(\alpha - 1) \sum_{\substack{k=1 \\ k \neq j}}^{i-1} \alpha^{i-1-k} x_k + (\alpha - 1 + \phi_j^i) \alpha^{i-1-j} x_j + x_i \leq \delta_i + \alpha^{i-1-j} \lfloor \gamma_j^i \rfloor \phi_j^i$$

for  $1 \leq j \leq i - 2$ ,  $3 \leq i \leq T - 1$  where  $\phi_j^i = \lceil \gamma_j^i \rceil - \gamma_j^i$  and  $\delta_i = \sum_{j=1}^i \alpha^{i-j} d_j$  for  $1 \leq i \leq T$ .

(ii) Show that  $386x_4^1 + 19x_4^2 + x_4^3 \leq 2826$  is an inequality that is satisfied by all integer solutions to the Berlin airlift model. Show that inequalities (10.6), the original equations and the additional inequality given here furnish an ideal description of the Berlin airlift problem.

---

(i) Suppose first that  $0 \leq x_j \leq \lfloor \gamma_j^i \rfloor$ . Then we write the inequality as follows

$$(\alpha - 1) \sum_{k=1}^{i-1} \alpha^{i-1-k} x_k + x_i \leq \delta_i + \alpha^{i-1-j} (\lceil \gamma_j^i \rceil - \gamma_j^i) (\lfloor \gamma_j^i \rfloor - x_j)$$

and thus the validity follows from inequality (10.4), since  $\alpha^{i-1-j} (\lceil \gamma_j^i \rceil - \gamma_j^i) (\lfloor \gamma_j^i \rfloor - x_j) \geq 0$ . Suppose

now that  $x_j \geq \lceil \gamma_j^i \rceil$ . We have

$$\begin{aligned} & \max\{(\alpha - 1) \sum_{\substack{k=1 \\ k \neq j}}^{i-1} \alpha^{i-1-k} x_k + x_i : \text{\textbf{x} satisfies (10.4) and (10.5), } x_j = \lambda\} \\ & \leq \max\{(\alpha - 1) \sum_{\substack{k=1 \\ k \neq j}}^{i-1} \alpha^{i-1-k} x_k + x_i : \sum_{\substack{k=1 \\ k \neq j}}^i \alpha^{i-k} x_k \leq \lfloor \frac{\delta_{i+1}}{\alpha - 1} \rfloor - \alpha^{i-j} \lambda\}. \end{aligned}$$

But then we have from the nonnegativity of the term in the summation

$$(\alpha - 1) \sum_{\substack{k=1 \\ k \neq j}}^{i-1} \alpha^{i-1-k} x_k + x_i \leq \alpha \sum_{\substack{k=1 \\ k \neq j}}^{i-1} \alpha^{i-1-k} x_k + x_i \leq \lfloor \frac{\delta_{i+1}}{\alpha - 1} \rfloor - \alpha^{i-j} \lambda,$$

Thus it suffices to show that

$$\left\lfloor \frac{\delta_{i+1}}{\alpha - 1} \right\rfloor - \alpha^{i-j} \lambda \leq \delta_i + \alpha^{i-1-j} \lfloor \gamma_j^i \rfloor \phi_j^i - \alpha^{i-1-j} (\alpha - 1 + \phi_j^i) \lambda,$$

for all integer  $\lambda \geq \lceil \gamma_j^i \rceil$ . In the case when  $\phi_j^i = 0$  the inequality reduces to

$$\left\lfloor \frac{\delta_{i+1}}{\alpha - 1} \right\rfloor - \alpha^{i-j} \lambda \leq \delta_i - \alpha^{i-j} \lambda + \alpha^{i-1-j} \lambda$$

which gives  $\gamma_j^i \leq \lambda = x_j$  and the validity follows. If  $\phi_j^i \neq 0$ , then  $\gamma_j^i$  is fractional,  $\lceil \gamma_j^i \rceil = \lfloor \gamma_j^i \rfloor + 1$ , and  $0 < \phi_j^i < 1$ . Then the inequality reduces to  $\gamma_j^i \leq \lfloor \gamma_j^i \rfloor \phi_j^i + \lambda(1 - \phi_j^i)$  which after substituting  $\lfloor \gamma_j^i \rfloor = \lceil \gamma_j^i \rceil - 1$ , gives  $\lceil \gamma_j^i \rceil \leq \lambda = x_j$  and thus the proof of validity is complete.

**(ii)** For the Berlin airlift model, we write the inequality of part (i) for  $i = 3, j = 1$ . We have  $\alpha = 20$ ,  $\delta_1 = 30$ ,  $\delta_2 = 150$ ,  $\delta_3 = 2890$  and  $\delta_4 = 55560$ . Thus  $\gamma_3^1 = \frac{1}{20} (\lfloor \frac{\delta_4}{19} \rfloor - \delta_3) = \frac{67}{10}$ , since  $20\delta_3 \geq 19 \lfloor \frac{\delta_4}{19} \rfloor > 19\delta_3$ . Then  $\phi_3^1 = \frac{3}{10}$  and the right-hand side of the inequality is  $\delta_3 + 20^{3-1-1} \lfloor \frac{67}{10} \rfloor \frac{3}{10} = 2826$ . Now we have  $(19 + \frac{3}{10})20x_1 + 19x_2 + x_3 \leq 2826$  which gives  $386x_1 + 19x_2 + x_3 \leq 2826$ .

We use the DDA program to calculate the extreme points of the polyhedron defined by the following system of equalities and inequalities

$$x_1 + \frac{1}{20}x_5 = 30, \quad -x_1 + x_2 - x_5 + \frac{1}{20}x_6 = -450, \quad -x_2 + x_3 - x_6 + \frac{1}{20}x_7 = -210 - x_3 + x_4 - x_7 + \frac{1}{20}x_8 = -240,$$

$$x_1 \leq 7, \quad 20x_1 + x_2 \leq 146, \quad 400x_1 + 20x_2 + x_3 \leq 2924, \quad 386x_1 + 19x_2 + x_3 \leq 2826,$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0, \quad x_6 \geq 0, \quad x_7 \geq 0, \quad x_8 \geq 0.$$

We find that the polyhedron is in fact a polytope with the following twenty two extreme points

$$\begin{aligned} & (0, 0, 0, 0, 600, 3000, 55800, 1111200), \quad (0, 0, 0, 55560, 600, 3000, 55800, 0), \\ & (0, 0, 2790, 0, 600, 3000, 0, 51000), \quad (0, 0, 2790, 2550, 600, 3000, 0, 0), \quad (0, 134, 244, 0, 600, 320, 0, 80), \\ & (0, 146, 0, 0, 600, 80, 320, 1600), \quad (0, 146, 0, 80, 600, 80, 320, 0), \quad (6, 0, 510, 0, 480, 720, 0, 5400), \\ & (6, 0, 510, 270, 480, 720, 0, 0), \quad (6, 14, 244, 0, 480, 440, 0, 80), \quad (7, 0, 0, 0, 460, 340, 2600, 47200), \\ & (7, 0, 0, 2360, 460, 340, 2600, 0), \quad (7, 0, 124, 0, 460, 340, 120, 80), \quad (7, 6, 0, 0, 460, 220, 320, 1600), \\ & (7, 6, 0, 80, 460, 220, 320, 0), \quad (0, 134, 244, 4, 600, 320, 0, 0), \quad (0, 146, 4, 0, 600, 80, 240, 80), \\ & (6, 14, 244, 4, 480, 440, 0, 0), \quad (7, 0, 124, 4, 460, 340, 120, 0), \quad (7, 6, 4, 0, 460, 220, 240, 80), \\ & (0, 146, 4, 4, 600, 80, 240, 0), \quad (7, 6, 4, 4, 460, 220, 240, 0) \end{aligned}$$

which are all integer, and thus the description is complete. We leave it to the reader to verify that the description is also minimal, i.e., that an ideal description of the convex hull of the example problem has been obtained.

---

### Exercise 10.5

- (i) Let  $A$  be the  $(m+n) \times (nm)$  constraint matrix of the transportation problem corresponding to the constraints (1.1) and (1.2) of Chapter 1. Show that  $A$  is totally unimodular.
  - (ii) Let  $A = (a_{ij}^i)_{i=1,\dots,m; j=1,\dots,n}$  satisfy (1) that  $a_{ij}^i \in \{0, \pm 1\}$ , (2) that every column of  $A$  has at most two nonzero entries, and (3) that every column of  $A$  containing two nonzeros contains a  $+1$  and a  $-1$  entry. Prove that  $A$  is totally unimodular.
  - (iii) Let  $A$  be defined like in (ii) and assume that it satisfies (1) and (2), but not (3). Give examples of such matrices that are not totally unimodular.
- 

**(i)** The constraint matrix  $A$  of the transportation problem is a 0-1 matrix with exactly two nonzero elements in each column both of which are 1. Moreover, we observe that the rows of the matrix can be partitioned into two sets,  $R_1$  and  $R_2$  say, one corresponding to constraints (1.1) and the other to constraints (1.2); see also Exercise 2.3.

We prove by induction that this matrix is totally unimodular. Trivially, the  $1 \times 1$  submatrices have determinants either zero or one. For the  $2 \times 2$  submatrices we have that either (a) all four elements are 1 in which case the determinant is zero, or, (b) at least one element is zero in which case the determinant is plus or minus the product of two elements and thus its value is in  $\{0, \pm 1\}$ . Assume now that all  $k \times k$  submatrices of  $A$  have determinants  $\{0, \pm 1\}$ , and consider a  $(k+1) \times (k+1)$  submatrix  $A_{k+1}$  of  $A$ . There are three possibilities: (a)  $A_{k+1}$  has a zero column, (b)  $A_{k+1}$  has a column, say  $t$ , with exactly one nonzero element, say  $a_t^i = 1$ , and (c) all columns of  $A_{k+1}$  have precisely two nonzero elements, both equal to 1. Since in case (a) the matrix has a zero column its determinant is zero. In case (b) we develop the determinant with respect to column  $t$  to get  $\det A_{k+1} = \pm a_t^i A_k$  where  $A_k$  is the  $k \times k$  submatrix resulting from the deletion of column  $t$  and row  $i$  from  $A_{k+1}$ . From the induction hypothesis  $\det A_k \in \{0, \pm 1\}$ , and thus since  $a_t^i = 1$  we have  $\det A_{k+1} \in \{0, \pm 1\}$ . In case (c) we claim that the rows of the matrix are linearly dependent and thus the determinant is zero. To prove that the rows of the matrix are linearly dependent we notice that since each column of the matrix  $A_{k+1}$  has exactly two ones, then one of these is in a row in  $R_1$  and the other in a row in  $R_2$ . But then adding up the rows of  $A_{k+1}$  that are in  $R_1$  we obtain a vector which is also the sum of the rows of  $A_{k+1}$  that are in  $R_2$ , i.e. the rows of  $A_{k+1}$  are linearly dependent and the proof is complete.

**(ii)** We prove by induction that every  $k \times k$  submatrix  $A_k$  of  $A$  has a determinant of  $\pm 1$  or 0. For  $k = 1$  the assertion is true since all elements of  $A$  are in  $\{0, \pm 1\}$ . For  $k = 2$ , we have that either (a) all four elements are nonzero in which case the determinant is zero since one column is a multiple of the other, or, (b) at least one element is zero in which case the determinant is plus or minus the product of two elements and thus its value is in  $\{0, \pm 1\}$ . Suppose now that every  $k \times k$  submatrix of  $A$  has a determinant of  $\pm 1$  or 0, and consider a  $(k+1) \times (k+1)$  matrix

$A_{k+1}$ . There are three possibilities: (a)  $A_{k+1}$  has a zero column, (b)  $A_{k+1}$  has a column, say  $t$ , with exactly one nonzero element, say  $a_t^i$ , and (c) all columns of  $A_{k+1}$  have precisely two nonzero elements, one +1 and one -1. Since in case (a) the matrix has a zero column and in case (c) its rows add up to the zero vector, the determinant in both these cases is zero. In case (c) we develop the determinant with respect to column  $t$  to get  $\det A_{k+1} = \pm a_t^i \det A_k$  where  $A_k$  is the  $k \times k$  submatrix resulting from the deletion of column  $t$  and row  $i$  from  $A_{k+1}$ . From the induction hypothesis  $\det A_k \in \{0, \pm 1\}$ , and thus since  $a_t^i = \pm 1$  we have  $\det A_{k+1} \in \{0, \pm 1\}$  and the proof is complete.

(iii) Since condition (1) is satisfied, all  $1 \times 1$  matrices are totally unimodular. However, for  $2 \times 2$  matrices one gets already a counterexample. Consider e.g. the matrix  $A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$  which satisfies conditions (1) and (2) and it has a determinant of 2.

---

**Exercise 10.6** (i) Show that the following  $6 \times 5$  zero-one matrix  $A$  is not balanced. (ii) Using e.g. the double description algorithm verify that  $A$  is both perfect and ideal.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$


---

(i) To show that  $A$  is not balanced it suffices to find a submatrix  $A'$  such that the polyhedron  $P(A', e')$  has a fractional extreme point. Such a submatrix is the following matrix

$$A' = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

One verifies that indeed the fractional point  $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$  is an extreme point of the polytope  $P = \{x \in \mathbb{R}^3 : x_1 + x_3 \leq 1, x_1 + x_2 \leq 1, x_2 + x_3 \leq 1, x \geq 0\}$  and thus the matrix  $A$  is not balanced.

(ii) Using the double description algorithm to find the extreme points of the polytope  $P = \{x \in \mathbb{R}^5 : Ax \leq e, x \geq 0\}$  we get that  $P$  has the following extreme points: 0,  $u^i$  for  $i = 1, \dots, 5$ ,  $u^2 + u^5$ , and  $u^1 + u^4$ , where  $u^i$  is the  $i$ -th unit vector in  $\mathbb{R}^5$ . Since all extreme points of  $P(A, e)$  are integral the matrix  $A$  is perfect.

Similarly, we find that the polyhedron  $P(-A, -e) = \{x \in \mathbb{R}^5 : -Ax \leq -e, x \geq 0\}$  has the following extreme points:  $u^2 + u^5$ ,  $u^1 + u^4$ ,  $u^2 + u^3 + u^4$ ,  $u^1 + u^3 + u^5$  and the unit vectors  $u^i$ , for  $i = 1, \dots, 5$  as extreme directions. Since all extreme points of  $P(-A, -e)$  are integral,  $A$  is ideal.

**\*Exercise 10.7**

Given integers  $M_0 = 0 < M_1 < \dots < M_n$  where  $n \geq 1$  model the requirement that  $x$  be a discrete-valued variable that assumes the values  $M_0, M_1$ , etc. or  $M_n$  in two different ways using  $n$  auxiliary zero-one variables. Compare the formulations that you obtained, like we did in the example of Chapter 10.2. Are your formulations (locally) ideal? Which one of the formulations do you prefer in a branch-and-bound or branch-and-cut context?

Let  $\delta_\ell \in \{0, 1\}$  for  $\ell = 1, \dots, n$ . Then

$$(F_1^*) \quad x = \sum_{\ell=1}^n M_\ell \delta_\ell, \quad \sum_{\ell=1}^n \delta_\ell \leq 1, \quad \delta_\ell \geq 0 \text{ and integer for } \ell = 1, \dots, n$$

expresses the discrete-value requirement on  $x$ . Let  $m_\ell = M_\ell - M_{\ell-1}$  and  $\xi_\ell \in \{0, 1\}$  for  $\ell = 1, \dots, n$ . Then

$$(F_2^*) \quad x = \sum_{\ell=1}^n m_\ell \xi_\ell, \quad 0 \leq \xi_n \leq \xi_{n-1} \leq \dots \leq \xi_2 \leq \xi_1 \leq 1 \text{ and integer for } \ell = 1, \dots, n$$

also formulates the problem correctly. To compare the two formulations we consider the polytopes

$$P_1 = \{(\boldsymbol{\delta}, x) \in \mathbb{R}^{n+1} : x = \sum_{\ell=1}^n M_\ell \delta_\ell, \quad \sum_{\ell=1}^n \delta_\ell \leq 1, \quad \delta_\ell \geq 0\} \text{ and}$$

$$P_2 = \{(\boldsymbol{\xi}, x) \in \mathbb{R}^{n+1} : x = \sum_{\ell=1}^n m_\ell \xi_\ell, \quad 0 \leq \xi_n \leq \xi_{n-1} \leq \dots \leq \xi_2 \leq \xi_1 \leq 1\}$$

corresponding to the linear programming relaxations of the two mixed-integer formulations. From

$$x = \sum_{\ell=1}^n M_\ell \delta_\ell = \sum_{\ell=1}^n m_\ell \sum_{j=\ell}^n \delta_j$$

we find that the transformation  $\xi_\ell = \sum_{j=\ell}^n \delta_j$  for  $1 \leq \ell \leq n$  maps any  $(\boldsymbol{\delta}, x) \in P_1$  into a point  $(\boldsymbol{\xi}, x) \in P_2$ . On the other hand, inverting the transformation, we find  $\delta_\ell = \xi_\ell - \xi_{\ell+1}$  for  $1 \leq \ell \leq n-1$  and  $\delta_n = \xi_n$ . It follows that for every  $(\boldsymbol{\xi}, x) \in P_2$  we find that under this mapping  $(\boldsymbol{\delta}, x) \in P_1$ , i.e., there exists a nonsingular linear transformation  $T$  such that  $P_1 = TP_2$ . Consequently  $F_1^*$  and  $F_2^*$  are locally, i.e., in the absence of any other constraints, equally “good” formulations of the discrete-value requirement. Consider now the extreme points of  $P_1$ . Since there are besides the nonnegativity conditions on  $\delta_\ell$  only two constraints and  $x$  is a free variable, every nonzero extreme point of  $P_1$  is of the form  $(\mathbf{u}_i, M_i)$  where  $\mathbf{u}_i \in \mathbb{R}^n$  is the  $i$ -th unit vector for  $i = 1, \dots, n$ . Consequently, in the absence of other constraints, both formulations are ideal formulations of the problem. If used in a branch-and-bound or branch-and-cut context, we would prefer to use  $F_1^*$  because branching on any  $\delta_\ell$  forces all other  $\delta_k$  to zero, while this is not necessarily the case

for the variables  $\xi_\ell$ . For more on analytical comparisons of different formulations see Padberg and T-Y Sung, “An analytical comparison of different formulations of the traveling salesman problem”, *Mathematical Programming*, 52 (1991), 315–357, and “An analytic symmetrization of max flow-min cut”, *Discrete Mathematics*, 165/166 (1997), 531–545, also by Padberg and Sung.

---

### \*Exercise 10.8

Let  $\phi(x)$  be any nonlinear function over some finite interval  $[a_0, a_u]$ . Given a partitioning  $a_0 < a_1 < a_2 < \dots < a_k = a_u$  of  $[a_0, a_u]$  we approximate  $\phi(x)$  by a piecewise linear function  $\hat{\phi}(x)$ , see Figure 10.6, using the function values  $b_\ell = \phi(a_\ell)$  at the points  $a_\ell$  for  $0 \leq \ell \leq k$ .

(i) Write  $x = a_0 + y_1 + \dots + y_k$  and require that each  $y_\ell$  is a continuous variable satisfying

- (a)  $0 \leq y_\ell \leq a_\ell - a_{\ell-1}$  for  $1 \leq \ell \leq k$  and
- (b) either  $y_i = a_i - a_{i-1}$  for  $1 \leq i \leq \ell$  or  $y_{\ell+1} = 0$  for  $1 \leq \ell \leq k-1$ .

Formulate the approximation problem as a mixed zero-one program (Model I).

(ii) Write  $x = a_0 \xi_0 + a_1 \xi_1 + \dots + a_k \xi_k$  and require that the continuous variables  $\xi_\ell$  satisfy

- (c)  $\sum_{\ell=0}^k \xi_\ell = 1$ ,  $\xi_\ell \geq 0$  for  $0 \leq \ell \leq k$ , and
- (d) at most two consecutive  $\xi_\ell$  and  $\xi_{\ell+1}$ , say, are positive.

Formulate the approximation problem as a mixed zero-one program (Model II).

(iii) Show that Model I is locally ideal, i.e., that in the absence of other constraints the linear programming relaxation of the formulation has mixed zero-one extreme points only.

(iv) Show that Model II is “worse” than Model I, i.e., that its linear programming relaxation is a polytope containing that one of Model I properly.

(v) Modify Model II so that it becomes an ideal formulation of the approximation problem.

(vi) Discuss how to use the above in the more general context of piecewise linear approximation of separable nonlinear functions of  $n$  variables, i.e., functions of the form  $\phi(x_1, \dots, x_n) = \sum_{j=1}^n \phi_j(x_j)$  where each  $\phi_j(x_j)$  is a nonlinear function of a single variable  $x_j$ .

---

(i) From (a) it follows that (b) can be replaced by the requirement

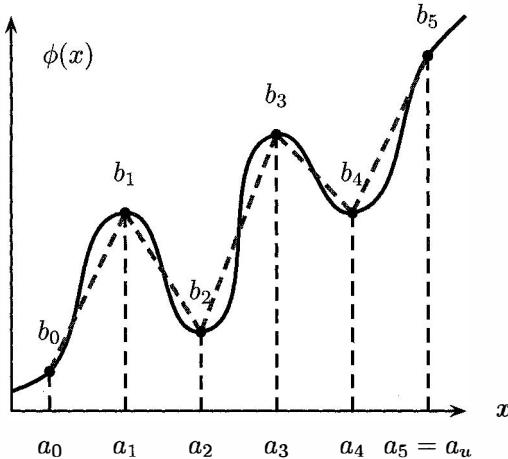
$$(b') \quad \text{either } y_i \geq a_i - a_{i-1} \text{ for } 1 \leq i \leq \ell \text{ or } y_{\ell+1} \leq 0 \text{ for } 1 \leq \ell \leq k-1.$$

We introduce zero-one variables  $z_\ell$  and consider the mixed zero-one model

$$x = a_0 + \sum_{\ell=1}^k y_\ell, \quad \hat{\phi}(x) = b_0 + \sum_{\ell=1}^k \frac{b_\ell - b_{\ell-1}}{a_\ell - a_{\ell-1}} y_\ell, \quad (1)$$

$$y_1 \leq a_1 - a_0, \quad y_k \geq 0, \quad (2)$$

$$y_\ell \geq (a_\ell - a_{\ell-1}) z_\ell, \quad y_{\ell+1} \leq (a_{\ell+1} - a_\ell) z_\ell \text{ for } 1 \leq \ell \leq k-1, \quad (3)$$



**Fig. 10.6.** Piecewise linear approximation

where  $z_\ell \in \{0, 1\}$  for  $1 \leq \ell \leq k - 1$  are the “new” 0-1 variables. For  $k = 1$  there is no need for a zero-one variable and (1), (2) describe the linear approximation correctly. For  $k = 2$  the correctness follows by examining the two cases where  $z_1 = 0$  and  $z_1 = 1$ , respectively. The correctness of the mixed zero-one model (1), ..., (3) for the piecewise linear approximation of a nonlinear function follows by induction on  $k$ . Model I has  $k$  real variables and  $k - 1$  zero-one variables.

From (3) every solution to (2) and (3) satisfies automatically  $1 \geq z_1 \geq z_2 \geq \dots \geq z_{k-1} \geq 0$ , thus the upper and lower bounds on the 0-1 variables are not required in the formulation. In a computer model, however, we would declare the variables  $z_\ell$  to be “binary” variables rather than general “integer” variables.

(ii) To formulate (c) and (d) as the set of solutions to a mixed zero-one program we introduce 0-1 variables  $\eta_\ell$  for  $0 \leq \ell \leq k - 1$  and consider the model

$$x = \sum_{\ell=0}^k a_\ell \xi_\ell, \quad \hat{\phi}(x) = \sum_{\ell=0}^k b_\ell \xi_\ell, \quad (4)$$

$$0 \leq \xi_0 \leq \eta_0, \quad 0 \leq \xi_\ell \leq \eta_{\ell-1} + \eta_\ell \quad \text{for } 1 \leq \ell \leq k - 1, \quad 0 \leq \xi_k \leq \eta_{k-1}, \quad (5)$$

$$\sum_{\ell=0}^k \xi_\ell = 1, \quad \sum_{\ell=0}^{k-1} \eta_\ell = 1, \quad (6)$$

$$\eta_\ell \geq 0 \quad \text{for } 1 \leq \ell \leq k - 2, \quad (7)$$

where  $\eta_\ell \in \{0, 1\}$  for  $0 \leq \ell \leq k - 1$  are the “new” 0-1 variables. The nonnegativity of  $\eta_0$  and  $\eta_{k-1}$  is implied by (5). For  $k = 1$  the formulation (4), ..., (7) of the problem at hand is evidently correct. The correctness of Model II for  $k \geq 1$  follows inductively. Model II has  $k + 1$  real variables and  $k$  zero-one variables.

(iii) Denote the linear programming (LP) relaxation of Model I by

$$F_{LP}^I = \{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{2k-1} : (\mathbf{y}, \mathbf{z}) \text{ satisfies (2) and (3)}\}. \quad (8)$$

We scale the continuous variables of Model I by introducing new variables

$$y'_\ell = y_\ell / (a_\ell - a_{\ell-1}) \text{ for } 1 \leq \ell \leq k. \quad (9)$$

The constraint set defining  $F_{LP}^I$  can thus be written as

$$y'_1 \leq 1, \quad y'_k \geq 0, \quad y'_\ell \geq z_\ell, \quad y'_{\ell+1} \leq z_\ell \text{ for } 1 \leq \ell \leq k-1. \quad (10)$$

The constraint matrix given by (10) is totally unimodular and hence by Cramer's rule every extreme point of the polytope given by (10) has all components equal to zero or one, i.e., Model I is locally ideal.

(iv) In the following we **assume** that  $k \geq 3$ , because for  $k \leq 2$  either model is locally ideal. To compare the two models we use the equations (6) to eliminate  $\xi_0$  and  $\eta_0$  from the formulation of Model II. Using the variable transformation

$$y_\ell = (a_\ell - a_{\ell-1}) \sum_{j=\ell}^k \xi_j \text{ for } 1 \leq \ell \leq k$$

and its inverse mapping that we calculate to be

$$\xi_j = \frac{y_j}{a_j - a_{j-1}} - \frac{y_{j+1}}{a_{j+1} - a_j} \text{ for } 1 \leq j \leq k-1, \quad \xi_k = \frac{1}{a_k - a_{k-1}} y_k,$$

we obtain the following equivalent formulation of Model II :

$$x = a_0 + \sum_{\ell=1}^k y_\ell, \quad \hat{\phi}(x) = b_0 + \sum_{\ell=1}^k \frac{b_\ell - b_{\ell-1}}{a_\ell - a_{\ell-1}} y_\ell, \quad (11)$$

$$y_1 \leq a_1 - a_0, \quad y_1 \geq (a_1 - a_0) \sum_{\ell=1}^{k-1} \eta_\ell, \quad (12)$$

$$(a_\ell - a_{\ell-1}) y_{\ell+1} \leq (a_{\ell+1} - a_\ell) y_\ell \text{ for } 1 \leq \ell \leq k-1, \quad (13)$$

$$\frac{y_1}{a_1 - a_0} - \frac{y_2}{a_2 - a_1} \leq 1 - \sum_{\ell=2}^{k-1} \eta_\ell, \quad \frac{y_\ell}{a_\ell - a_{\ell-1}} - \frac{y_{\ell+1}}{a_{\ell+1} - a_\ell} \leq \eta_{\ell-1} + \eta_\ell \text{ for } 2 \leq \ell \leq k-1, \quad (14)$$

$$y_k \geq 0, \quad y_k \leq (a_k - a_{k-1}) \eta_{k-1}, \quad (15)$$

$$\eta_\ell \geq 0 \text{ for } 1 \leq \ell \leq k-2, \quad (16)$$

where  $\eta_\ell \in \{0, 1\}$  for  $1 \leq \ell \leq k-1$ . Note that (12) implies that  $\sum_{\ell=1}^{k-1} \eta_\ell \leq 1$  and thus  $\sum_{\ell=j}^{k-1} \eta_\ell \leq 1$  for all  $1 \leq j \leq k-1$  and feasible 0-1 values  $\eta_\ell$ ,  $1 \leq \ell \leq k-1$ . Using the variable substitution

$$z_j = \sum_{\ell=j}^{k-1} \eta_\ell \text{ for } 1 \leq j \leq k-1,$$

which is *integrality preserving* because its inverse is given by

$$\eta_j = z_j - z_{j+1} \text{ for } 1 \leq j \leq k-2, \quad \eta_{k-1} = z_{k-1},$$

the above constraints (12), ..., (16) can be written equivalently as follows:

$$y_1 \leq a_1 - a_0, \quad y_1 \geq (a_1 - a_0)z_1, \quad (17)$$

$$(a_\ell - a_{\ell-1})y_{\ell+1} \leq (a_{\ell+1} - a_\ell)y_\ell \text{ for } 1 \leq \ell \leq k-1, \quad (18)$$

$$\frac{y_1}{a_1 - a_0} - \frac{y_2}{a_2 - a_1} \leq 1 - z_2, \quad \frac{y_\ell}{a_\ell - a_{\ell-1}} - \frac{y_{\ell+1}}{a_{\ell+1} - a_\ell} \leq z_{\ell-1} - z_{\ell+1} \text{ for } 2 \leq \ell \leq k-1, \quad (19)$$

$$y_k \geq 0, \quad y_k \leq (a_k - a_{k-1})z_{k-1}, \quad (20)$$

$$z_\ell - z_{\ell+1} \geq 0 \text{ for } 1 \leq \ell \leq k-2, \quad (21)$$

where for  $\ell = k-1$  we simply let  $z_k = 0$  in (19) and the integer variables  $z_\ell$  are 0-1 valued for  $1 \leq \ell \leq k-1$ . It follows that the (equivalently) changed Model II has now the same variable set as Model I and we are in the position to *compare* the two formulations in the context of a linear programming based approach to the solution of the corresponding mixed-integer programming problem. Note that like in Model I the constraints (17), (20) and (21) imply that every feasible solution to (17), ..., (21) automatically satisfies  $1 \geq z_1 \geq z_2 \geq \dots \geq z_{k-1} \geq 0$ . We denote the LP relaxation of the (equivalently) changed Model II by

$$F_{LP}^{II} = \{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{2k-1} : (\mathbf{y}, \mathbf{z}) \text{ satisfies (17), ..., (21)}\}, \quad (22)$$

which like  $F_{LP}^I$  is a polytope in  $\mathbb{R}^{2k-1}$ . Let  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^I$ , i.e.,  $(\mathbf{y}, \mathbf{z})$  satisfies (2) and (3). Then  $(\mathbf{y}, \mathbf{z})$  satisfies (17) and (20) trivially. From (3) and  $a_\ell - a_{\ell-1} > 0$  for all  $1 \leq \ell \leq k$  we calculate

$$(a_{\ell+1} - a_\ell)y_\ell \geq (a_{\ell+1} - a_\ell)(a_\ell - a_{\ell-1})z_\ell \geq (a_\ell - a_{\ell-1})y_{\ell+1}$$

for  $1 \leq \ell \leq k-1$  and thus (18) is satisfied. From (2) and (3) we have  $y_1 \leq a_1 - a_0$  and  $y_2 \geq (a_2 - a_1)z_2$  and thus the first relation of (19) follows. Again from (3) we have  $y_\ell \leq (a_\ell - a_{\ell-1})z_{\ell-1}$  and  $y_{\ell+1} \geq (a_{\ell+1} - a_\ell)z_{\ell+1}$  for all  $2 \leq \ell \leq k-1$ , where  $z_k = 0$ , and thus combining the two inequalities we see that (19) is satisfied. Every  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^I$  satisfies  $1 \geq z_1 \geq z_2 \geq \dots \geq z_{k-1} \geq 0$  and thus (21) is satisfied as well. Consequently,  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^{II}$  and thus  $F_{LP}^I \subseteq F_{LP}^{II}$ . Let  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^{II}$  be such that  $\mathbf{z} \in \{0, 1\}^{k-1}$ . From (17), ..., (21) it follows that  $y_i = a_i - a_{i-1}$  for  $i = 1, \dots, h$ ,  $0 \leq y_{h+1} \leq a_{h+1} - a_h$ ,  $y_i = 0$  for  $i = h+2, \dots, k$ ,  $z_i = 1$  for  $i = 1, \dots, h$ ,  $z_i = 0$  for  $i = h+1, \dots, k-1$  where  $0 \leq h \leq k-1$ . From (2) and (3) thus  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^I$ . Now consider  $(\mathbf{y}, \mathbf{z})$  given by  $y_1 = (a_1 - a_0)/2$ ,  $y_j = 0$  for  $2 \leq j \leq k$ ,  $z_1 = z_2 = 1/2$ , and  $z_j = 0$  for  $3 \leq j \leq k-1$ . It follows that  $(\mathbf{y}, \mathbf{z})$  satisfies (17), ..., (21), i.e.,  $(\mathbf{y}, \mathbf{z}) \in F_{LP}^{II}$ . But  $(\mathbf{y}, \mathbf{z})$  violates the constraint  $y_2 \geq (a_2 - a_1)z_2$  of Model I and thus  $(\mathbf{y}, \mathbf{z}) \notin F_{LP}^I$ . Since  $F_{LP}^{II}$  is a polytope, it follows that  $F_{LP}^{II}$  has extreme points  $(\mathbf{y}, \mathbf{z})$  with  $\mathbf{z} \notin \{0, 1\}^{k-1}$ . Thus  $F_{LP}^{II}$  has extreme points with fractional components for  $\mathbf{z}$  and indeed it has many such extreme points. It is not overly difficult to characterize all of them, which we leave as another good exercise.

(v) By reversing the various transformations that we have used to analyze Model II one obtains from Model I the following Model III for the piecewise linear approximation problem:

$$x = \sum_{\ell=0}^k a_\ell \xi_\ell, \quad \hat{\phi}(x) = \sum_{\ell=0}^k b_\ell \xi_\ell, \quad (23)$$

$$\sum_{\ell=0}^k \xi_\ell = 1, \quad \sum_{\ell=0}^{k-1} \eta_\ell = 1, \quad (24)$$

$$\sum_{j=\ell}^{k-1} \eta_j \geq \sum_{j=\ell+1}^k \xi_j \geq \sum_{j=\ell+1}^{k-1} \eta_j \text{ for } 1 \leq \ell \leq k-2, \quad (25)$$

$$0 \leq \xi_0 \leq \eta_0, \quad 0 \leq \xi_k \leq \eta_{k-1}, \quad (26)$$

where  $\eta_\ell \in \{0, 1\}$  for  $0 \leq \ell \leq k-1$ . Model III has at first sight little resemblance to the original Model II except that the same set of variables is used. More precisely, let

$$\begin{aligned} P_{LP} &= \{(\xi, \eta) \in \mathbb{R}^{2k+1} : (\xi, \eta) \text{ satisfies (5), (6), (7)}\}, \\ P_{LP}^\# &= \{(\xi, \eta) \in \mathbb{R}^{2k+1} : (\xi, \eta) \text{ satisfies (24), (25), (26)}\}, \end{aligned}$$

be the linear programming relaxations of Model II and III. It follows by construction and from part (iv) that

$$P_{LP}^\# \subset P_{LP} \text{ and } P_{LP}^\# = \text{conv}(P_{LP} \cap (\mathbb{R}^{k+1} \times \mathbb{Z}^k)).$$

Model III shares *locally* the property of Model I of having all its extreme points  $(\xi, \eta)$  satisfy  $\eta \in \{0, 1\}^k$ . Model III can be used in lieu of Model I, but Model II should definitely be abandoned despite its popularity in the textbooks. Model II just happens to be a poor formulation for the piecewise linear approximation problem when linear programming methods are used.

**(vi)** If there are  $n$  variables of a separable nonlinear function, we can use the piecewise linear approximation for each one of the functions to write down a mixed zero-one formulation for the entire problem. Using e.g. Model I one gets an ideal formulation for the overall problem –provided there are no other constraints constraining the variables  $x_1, \dots, x_n$ . In case there are such constraints then the resulting linear programming relaxation can, of course, not be expected to be an ideal formulation, i.e., it will typically have extreme points where the zero-one variables assume fractional values.

### \*Exercise 10.9

- (i) Consider the capital budgeting problem of Exercise 6.3. Suppose that the lending rate  $r_1$  and the borrowing rate  $r_2$  for money are different ( $r_1 \leq r_2$ ). Formulate the problem as a mixed zero-one programming problem with variables  $y_i$  for money lent and  $z_i$  for money borrowed in period  $i \in \{1, \dots, T\}$ .
- (ii) Give a number example to show that the linear programming relaxation of the model obtained in part (i) has extreme points with fractional values for the zero-one variables.
- (iii) Using the notation of Exercise 6.3 define recursively for  $i = 1, \dots, T$  and  $S \subseteq N = \{1, \dots, n\}$

$$\beta_i(S) = \max\{0, -\sum_{k=1}^i (1+r_1)^{i-k} (s_k + \sum_{j \in S} a_{kj}) + (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \beta_k(S)\}.$$

Show that  $\beta_i(S)$  is the minimum borrowing need in period  $i$  if the projects with  $j \in S$  are accepted and those in  $N - S$  are rejected.

- (iv) In the case of a single project, i.e., if  $n = 1$ , show that it is optimal to approve the project if and only if

$$c_1 + \sum_{i=1}^T (1+r_1)^{T-i} a_{i1} > (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} (\beta_i(1) - \beta_i),$$

where  $\beta_i$  is the “unavoidable” borrowing need ( $S = \emptyset$ ) and  $\beta_i(1)$  the firm’s borrowing need resulting from the approval of the project, i.e., that it is optimal to accept a project if and only if its net present value exceeds the discounted lost interest due to the changed borrowing needs.

---

- (i) Using the same notation as in Exercise 6.3 we get the following mixed zero-one linear program:

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j + y_T - z_T \\ \text{s.t.} \quad & - \sum_{j=1}^n a_{1j} x_j + y_1 - z_1 \leq s_1 \\ & - \sum_{j=1}^n a_{ij} x_j - (1+r_1)y_{i-1} + (1+r_2)z_{i-1} + y_i - z_i \leq s_i \quad \text{for } i = 2, \dots, T \\ & x_j \in \{0, 1\} \quad \text{for } j = 1, \dots, n \\ & y_i, z_i \geq 0 \quad \text{for } i = 1, \dots, T. \end{aligned}$$

The objective function models the financial position of the firm as of the (beginning of) year  $T$ , the “horizon” year, as it results from project selection/rejection and the activities of borrowing or lending funds. The constraints assure that these activities can be realized by the financial means generated by the existing operations of the firm. Taking  $r_1 = r_2$ , i.e., assuming perfect capital market conditions, we retrieve the model of Exercise 6.3.

- (ii) Consider the problem of accepting/rejecting a single project with the data

$$a_{11} = -4, a_{21} = -2, a_{31} = 6, c_1 = 2, s_1 = 1, s_2 = 0, s_3 = 0,$$

the lending rate of  $r_1 = 0.10$  and the borrowing rate  $r_2 = 0.20$ . We get the capital budgeting problem:

$$\begin{aligned} \max \quad & 2x_1 + y_3 - z_3 \\ \text{s.t.} \quad & 4x_1 + y_1 - z_1 \leq 1 \\ & 2x_1 - 1.1y_1 + 1.2z_1 + y_2 - z_2 \leq 0 \\ & -6x_1 - 1.1y_2 + 1.2z_2 + y_3 - z_3 \leq 0 \\ & x_1 \in \{0, 1\} \\ & y_i, z_i \geq 0 \quad \text{for } i = 1, 2, 3 \end{aligned}$$

Solving the relaxed linear program where  $x_1 \in \{0, 1\}$  is replaced by  $0 \leq x_1 \leq 1$  we obtain the optimal solution

$$x_1 = 0.25, y_1 = y_2 = z_1 = z_3 = 0, y_3 = 0.9, z_2 = 0.5$$

with an objective function value of 1.4. The solution indicates that the firm should seek a 25% “partnership” in the project – assuming, of course, that a “partner” taking the remaining 75% can be found. In most investment situations this will not be possible because of the indivisibility of the projects. Thus the imperfection of the capital markets leads to a “true” mixed-integer problem that – contrary to the case of a perfect capital market, see Exercise 6.3 – typically *can not* be solved by the associated linear programming relaxation.

In the example it is easy to find the optimal answer. Setting  $x_1 = 0$  we find  $y_1 = 1, y_2 = 1.1, y_3 = 1.21, z_1 = z_2 = z_3 = 0$  to be an optimal solution with an objective function value of 1.21 MUs (monetary units). On the other hand, setting  $x_1 = 1$  we find the optimal solution  $y_1 = y_2 = y_3 = 0, z_1 = 3, z_2 = 5.6, z_3 = 0.72$  with an objective function value of 1.28 MUs. Thus to maximize its financial position at the beginning of year 3 the firm should accept rather than reject the project.

It is instructive to see what the traditional project selection rules<sup>1</sup> yield in the case of this number example: Applying the *NPV* rule with  $k = r_2$  and  $k = r_1$ , respectively, we find  $NPV(r_2) = -4 - 2/1.2 + (6 + 2)/1.44 = -0.1111$  which means that the project is rejected, whereas  $NPV(r_1) = 0.7933$  indicates that the project is accepted. The internal rate of return is calculated to be  $r \approx 0.186$ . Since  $r_1 < r < r_2$ , the project is accepted if we use the lending rate as the cut-off rate whereas it is rejected if the borrowing rate is used as the cut-off rate. Finally, calculating the profitability index using  $r_2$  as the discount rate  $PI = ((6 + 2)/1.44)/(4 + 2/1.2) = 0.9804$  whereas the same calculation for  $r_1$  yields  $PI = 1.1364$ . The outcome depends thus upon which rate of interest is used as the “cost of capital”. The question which one of the two rates to use is a subject that has been treated in the financial literature. It is ill-posed because the optimal decision rule depends not only on both rates  $r_1$  and  $r_2$  and the cash flows associated with the project, but also on the investment funds available in periods  $1, \dots, T$ ; see part (iv) below.

(iii) We have to show that  $z_\ell = \beta_\ell(S)$  are optimal solutions to the linear programs

$$\begin{array}{ll} \min & z_\ell \\ \text{s.t.} & -y_1 + z_1 \geq -s_1 - \sum_{j \in S} a_{1j} \\ & (1 + r_1)y_{i-1} - (1 + r_2)z_{i-1} - y_i + z_i \geq -s_i - \sum_{j \in S} a_{ij} \quad \text{for } i = 2, \dots, T \\ & y_i, z_i \geq 0 \quad \text{for } i = 1, \dots, T. \end{array}$$

---

<sup>1</sup>The traditional project selection methods are the *NPV*, the *IRR* and the *PI* rules:

- (1) *Net Present Value (NPV) Rule:* The formula is  $NPV = \sum_{t=1}^N \frac{F_t}{(1+k)^t} - C$  where  $F_t$  is the net cash flow in period  $t$ ,  $k$  is the “cost of capital”,  $C$  is the initial cost of the project and  $N$  is the project’s expected life. The project is accepted if its NPV is positive. If two projects are mutually exclusive the one with the higher NPV is chosen.
- (2) *Internal Rate of Return (IRR) Rule:* IRR is the interest rate  $r$  which equates the net present value of the future inflows/outflows associated with the project to the initial cost, i.e.,  $\sum_{t=1}^N \frac{F_t}{(1+r)^t} - C = 0$ . The project is accepted if its  $r > k$ . If two projects are mutually exclusive the one with higher  $r$  is chosen.
- (3) *Profitability Index (PI) Rule:* PI is the relative profitability of the project, or the present value of the benefits per dollar of cost, i.e.,  $PI = \frac{PV(\text{benefits})}{\text{cost}}$ . The project is accepted if its  $PI > 1$ . If two projects are mutually exclusive the one with higher PI is chosen.

where  $\ell = 1, \dots, T$ . Using  $\beta_i(S)$  we define

$$\lambda_i(S) = \max\{0, \sum_{k=1}^i (1+r_1)^{i-k} (s_k + \sum_{j \in S} a_{kj}) - (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \beta_k(S)\}.$$

and claim that  $z_i = \beta_i(S)$  and  $y_i = \lambda_i(S)$  for  $i = 1, \dots, T$  are feasible for the linear program. Note that

$$-y_i + z_i = -\lambda_i(S) + \beta_i(S) = -\sum_{k=1}^i (1+r_1)^{i-k} (s_k + \sum_{j \in S} a_{kj}) + (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \beta_k(S)$$

for all  $1 \leq i \leq T$ . By definition  $z_i \geq 0$ ,  $y_i \geq 0$  and the first constraint is satisfied. Suppose inductively that constraints  $1, \dots, i$  are satisfied. We calculate for constraint  $i+1$

$$\begin{aligned} (1+r_1)y_i - (1+r_2)z_i - y_{i+1} + z_{i+1} &= (1+r_1)(y_i - z_i) - (r_2 - r_1)z_i - y_{i+1} + z_{i+1} = \\ (1+r_1) \left( \sum_{k=1}^i (1+r_1)^{i-k} (s_k + \sum_{j \in S} a_{kj}) - (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \beta_k(S) \right) - (r_2 - r_1)\beta_i(S) \\ - \sum_{k=1}^{i+1} (1+r_1)^{i+1-k} (s_k + \sum_{j \in S} a_{kj}) + (r_2 - r_1) \sum_{k=1}^i (1+r_1)^{i-k} \beta_k(S) &= -s_{i+1} - \sum_{j \in S} a_{i+1j} \end{aligned}$$

and thus feasibility of the solution follows. Note that  $z_1 \geq \max\{0, -s_1 - \sum_{j \in S} a_{1j}\}$  and thus from the feasibility of the solution it follows that  $z_1 = \beta_1(S)$  is an optimal solution if  $\ell = 1$ . Suppose inductively that  $z_i = \beta_i(S)$  is an optimal solution for  $i = 1, \dots, \ell$ . The dual of the corresponding linear program for  $i = \ell + 1$ , augmented by the redundant constraints  $z_i \geq \beta_i(S)$  for  $i = 1, \dots, \ell$ , is

$$\begin{aligned} \max \quad & \sum_{i=1}^T (-s_i - \sum_{j \in S} a_{ij}) u_i + \sum_{k=1}^{\ell} \beta_k(S) v_k \\ \text{s.t.} \quad & -u_i + (1+r_1)u_{i+1} \leq 0 \quad \text{for } i = 1, \dots, T \\ & u_i - (1+r_2)u_{i+1} + v_i = 0 \quad \text{for } i = 1, \dots, \ell \\ & u_{\ell+1} - (1+r_2)u_{\ell+2} \leq 1 \\ & u_i - (1+r_2)u_{i+1} \leq 0 \quad \text{for } i = \ell + 2, \dots, T \end{aligned}$$

where  $u_i \geq 0$  for  $i = 1, \dots, T$ ,  $u_i = 0$  for  $i > T$  and  $v_i \geq 0$  for  $i = 1, \dots, \ell$ . The solution  $u_i = 0$  for  $i = 1, \dots, T$  and  $v_i = 0$  for  $i = 1, \dots, \ell$  is feasible to this linear program and so is the solution  $u_k = (1+r_1)^{\ell+1-k}$  for  $k = 1, \dots, \ell + 1$ ,  $u_k = 0$  for  $k = \ell + 2, \dots, T$  and  $v_i = (r_2 - r_1)(1+r_1)^{\ell-i}$  for  $i = 1, \dots, \ell$ . Evaluating the objective function, it follows that the dual linear program has a feasible solution with objective function value greater than or equal  $\beta_{\ell+1}(S)$  and hence by weak duality, the assertion follows.

**(iv)** For notational simplicity we drop the subscript 1 and write  $x$ ,  $c$  and  $a_i$  for  $i = 1, \dots, T$ . By construction it follows that the inequalities  $z_i \geq \beta_i + \Delta_i x$  are valid inequalities for the mixed-integer program, where  $\Delta_i = \beta_i(1) - \beta_i$  and  $i = 1, \dots, T$ . Consider the linear program

$$\begin{aligned}
\max \quad & cx + y_T - z_T \\
\text{s.t.} \quad & -a_1x + y_1 - z_1 \leq s_1 \\
& -a_i x - (1+r_1)y_{i-1} + (1+r_2)z_{i-1} + y_i - z_i \leq s_i \quad \text{for } i = 2, \dots, T \\
& \Delta_i x - z_i \leq -\beta_i \quad \text{for } i = 1, \dots, T \\
& 0 \leq x \leq 1, \quad y_i \geq 0 \quad \text{for } i = 1, \dots, T.
\end{aligned}$$

We define a solution  $(x, \mathbf{y}, \mathbf{z})$  as follows: For  $i = 1, \dots, T$  let  $z_i = \beta_i + \Delta_i x$  and

$$\begin{aligned}
y_i &= \beta_i + \sum_{k=1}^i (1+r_1)^{i-k} s_k - (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \beta_k \\
&\quad + \left\{ \Delta_i + \sum_{k=1}^i (1+r_1)^{i-k} a_k - (r_2 - r_1) \sum_{k=1}^{i-1} (1+r_1)^{i-1-k} \Delta_k \right\} x \\
x &= \begin{cases} 1 & \text{if } c + \sum_{i=1}^T (1+r_1)^{T-i} a_i - (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} \Delta_i > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

It follows that  $z_i \geq 0$  and by the definitions of  $\beta_i$  and  $\beta_i(1)$  that  $y_i(x=0) \geq 0$  and  $y_i(x=1) \geq 0$  for all  $i$ . Consequently  $y_i \geq 0$  for all  $i$  and  $0 \leq x \leq 1$ . By calculation it follows that the  $2T$  first constraints are all satisfied at equality. The solution  $(x, \mathbf{y}, \mathbf{z})$  is thus feasible and has an objective function value of

$$\begin{aligned}
cx + y_T - z_T &= \max\{0, c + \sum_{i=1}^T (1+r_1)^{T-i} a_i - (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} \Delta_i\} \\
&\quad + \sum_{k=1}^T (1+r_1)^{T-k} s_k - (r_2 - r_1) \sum_{k=1}^{T-1} (1+r_1)^{T-1-k} \beta_k.
\end{aligned}$$

To prove the optimality of the solution we consider the dual linear program which is

$$\begin{aligned}
\min \quad & \sum_{i=1}^T s_i u_i - \sum_{i=1}^T \beta_i v_i + w \\
\text{s.t.} \quad & -\sum_{i=1}^T a_i u_i + \sum_{i=1}^T \Delta_i v_i + w \geq c \\
& u_i - (1+r_1)u_{i+1} \geq 0 \quad \text{for } i = 1, \dots, T-1 \\
& u_T \geq 1 \\
& -u_i + (1+r_2)u_{i+1} - v_i = 0 \quad \text{for } i = 1, \dots, T-1 \\
& u_T - v_T = -1 \\
& w \geq 0, \quad u_i, v_i \geq 0 \quad \text{for } i = 1, \dots, T.
\end{aligned}$$

We define  $u_i = (1+r_1)^{T-i}$  for  $i = 1, \dots, T$ ,  $v_i = (r_2 - r_1)(1+r_1)^{T-i-1}$  for  $i = 1, \dots, T-1$  and  $v_T = 0$ . It follows that the first constraint reduces to

$$w \geq c + \sum_{i=1}^T (1+r_1)^{T-i} a_i - (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} \Delta_i$$

and the other constraints are all satisfied. Consequently, setting  $w$  equal to the maximum of zero and the right-hand of the last inequality we have a feasible solution to the dual linear program with objective function value

$$\begin{aligned} \sum_{i=1}^T s_i u_i - \sum_{i=1}^T \beta_i v_i + w &= \sum_{i=1}^T (1+r_1)^{T-i} s_i - (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} \beta_i \\ &\quad + \max\{0, c + \sum_{i=1}^T (1+r_1)^{T-i} a_i - (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{T-1-i} \Delta_i\} \end{aligned}$$

which is the same as the one of the primal solution obtained above. Thus by weak duality the decision rule is optimal. Write the “horizon value”  $c$  of the project as

$$c = \sum_{k=T+1}^{\infty} (1+r_1)^{T-k} a_k$$

where  $a_{T+1}, a_{T+2}, \dots$  are all future cash flows associated with the project. Then the optimal decision for project approval can be restated as follows:

$$NPV(r_1) = \sum_{i=1}^{\infty} (1+r_1)^{-i+1} a_i > (r_2 - r_1) \sum_{i=1}^{T-1} (1+r_1)^{-i} (\beta_i(1) - \beta_i).$$

The left-hand side of the inequality is the net present value of all net cash flows associated with the project evaluated at the lending rate, while the right-hand side of the inequality can be interpreted as a “hurdle” that has to be applied to the traditional net present value rule; for more see Padberg and Wilczak, “Optimal project selection when borrowing and lending rates differ”, *Mathematical and Computer Modelling*, 29 (1999), 63–78.

---

### \*Exercise 10.10

You are given a “big” rectangular box  $B_0$  with sides of length  $D_X$ ,  $D_Y$  and  $D_Z$  units, respectively. As shown in Figure 10.7 some corner of  $B_0$  is selected to fix a (three-dimensional) coordinate system with  $X$ ,  $Y$  and  $Z$  axes as the frame of reference. In addition to  $B_0$  you have  $n$  “small” boxes  $B_1, \dots, B_n$  having sides of length  $L_{1i}$ ,  $L_{2i}$  and  $L_{3i}$  units, respectively, where  $i = 1, \dots, n$ . The problem is to pack e.g. as many of the small boxes into  $B_0$  as possible under the restriction that the faces of every small box packed into  $B_0$  are parallel to the faces of  $B_0$  (orthogonal placement). Other than that there are no restrictions on the placement of the small boxes, but the model should permit the user to eventually prescribe the packing and positioning in  $B_0$  of certain small boxes. Also desirable is to have the model allow for static balancing, i.e., the center of gravity of the packed box  $B_0$  should fall near to the geometric center of gravity of  $B_0$ . Formulate the orthogonal packing problem as a mixed-integer linear program.

---

Like in the case of the big box  $B_0$ , for each small box  $B_i$  select a corner of  $B_i$  to fix a coordinate system with axes 1, 2 and 3 of lengths  $L_{1i}$ ,  $L_{2i}$  and  $L_{3i}$ , respectively. In the following “box” refers usually to a “small” box.

Let

$$\delta_{\alpha i}^H = 1 \text{ if axis } \alpha \text{ of box } i \text{ is parallel to the } H \text{ axis, } \delta_{\alpha i}^H = 0 \text{ otherwise,}$$

where  $\alpha \in \{1, 2, 3\}$ ,  $H \in \{X, Y, Z\}$  and  $i = 1, \dots, n$ . The orthogonal packing requirement is modeled by the *orthogonal placement* constraints:

$$\sum_H \delta_{1i}^H \leq 1, \quad \sum_H \delta_{2i}^H = \sum_H \delta_{1i}^H, \quad \sum_\alpha \delta_{\alpha i}^H = \sum_H \delta_{1i}^H, \quad (1)$$

where  $H \in \{X, Y, Z\}$ ,  $\alpha \in \{1, 2, 3\}$  and  $i = 1, \dots, n$ . Note that  $\sum_H \delta_{1i}^H = 0$  means that box  $B_i$  is not put into  $B_0$ . To *position* the small box  $B_i$  in the big one denote by

$x_i^X, x_i^Y, x_i^Z$  = the coordinates of the center of gravity of  $B_i$  in the  $XYZ$  coordinate system,

where  $i = 1, \dots, n$ . The  $x_i^X, x_i^Y, x_i^Z$  variables are continuous ("flow") variables that must satisfy *domain* constraints and *non-intersection* constraints.

Let  $\ell_{\alpha i} = \frac{1}{2}L_{\alpha i}$  for  $\alpha \in \{1, 2, 3\}$  and  $i = 1, \dots, n$ . The *domain* constraints are

$$\sum_\alpha \ell_{\alpha i} \delta_{\alpha i}^H \leq x_i^H \leq \sum_\alpha (D_H - \ell_{\alpha i}) \delta_{\alpha i}^H, \quad (2)$$

where  $H \in \{X, Y, Z\}$  and  $i = 1, \dots, n$ . These constraints ensure that the center of gravity of  $B_i$  (if put into the big box) is chosen so that the entire box fits into  $B_0$ .

The space occupied by two boxes  $B_i$  and  $B_j$  that are put into  $B_0$  is disjoint if –depending upon the rotations of  $B_i$  and  $B_j$ , respectively– the coordinate  $x_i^H$  is "far enough" (to the left or to the right) from the coordinate  $x_j^H$  for one direction  $H \in \{X, Y, Z\}$ . Say that box  $B_i$  "precedes" or is positioned "to the left" of  $B_j$  along the  $H$  axis if  $x_i^H + \sum_\alpha \ell_{\alpha i} \delta_{\alpha i}^H + \sum_\alpha \ell_{\alpha j} \delta_{\alpha j}^H \leq x_j^H$ . To express the condition that the space occupied by any two boxes put into  $B_0$  is disjoint, introduce the zero-one variables

$$\lambda_{ij}^H = 1 \text{ if box } B_i \text{ precedes } B_j \text{ along the } H \text{ axis, } \lambda_{ij}^H = 0 \text{ otherwise,}$$

where  $1 \leq i \neq j \leq n$  and  $H \in \{X, Y, Z\}$ . For  $1 \leq i < j \leq n$  and  $H \in \{X, Y, Z\}$  the *non-intersection* constraints are :

$$D_H \lambda_{ji}^H + \sum_\alpha \ell_{\alpha i} \delta_{\alpha i}^H - \sum_\alpha (D_H - \ell_{\alpha j}) \delta_{\alpha j}^H \leq x_i^H - x_j^H \leq \sum_\alpha (D_H - \ell_{\alpha i}) \delta_{\alpha i}^H - \sum_\alpha \ell_{\alpha j} \delta_{\alpha j}^H - D_H \lambda_{ij}^H, \quad (3)$$

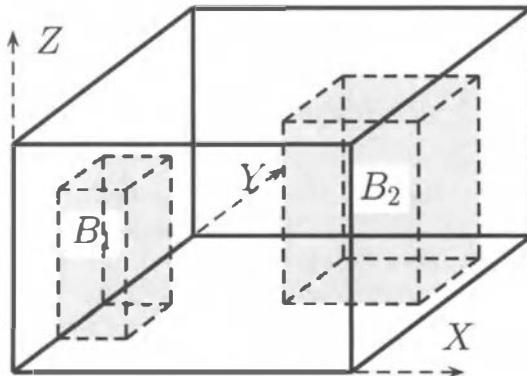
$$\sum_H (\lambda_{ij}^H + \lambda_{ji}^H) \leq \sum_H \delta_{1i}^H, \quad \sum_H (\lambda_{ij}^H + \lambda_{ji}^H) \leq \sum_H \delta_{1j}^H, \quad (4)$$

$$\sum_H \delta_{1i}^H + \sum_H \delta_{1j}^H \leq 1 + \sum_H (\lambda_{ij}^H + \lambda_{ji}^H). \quad (5)$$

The logical conditions (4) model the fact that if a box is placed to the left or to the right of another box in  $B_0$ , then both boxes must be in the  $B_0$ . The conditions (5) express the fact that if two boxes are put into  $B_0$ , then along at least one of the coordinate axes the center of gravity of one box must be to the left or to the right of the other box.

To justify the non-intersection constraints (3) note that the domain constraints (2) imply

$$\sum_\alpha \ell_{\alpha i} \delta_{\alpha i}^H - \sum_\alpha (D_H - \ell_{\alpha j}) \delta_{\alpha j}^H \leq x_i^H - x_j^H \leq \sum_\alpha (D_H - \ell_{\alpha i}) \delta_{\alpha i}^H - \sum_\alpha \ell_{\alpha j} \delta_{\alpha j}^H,$$



**Fig. 10.7.** Packing small boxes into  $B_0$

where  $H \in \{X, Y, Z\}$ . Consequently, if  $\lambda_{ij}^H = 0$  or  $\lambda_{ji}^H = 0$ , then the corresponding non-intersection constraint is redundant. If  $\lambda_{ij}^H = 1$  or  $\lambda_{ji}^H = 1$ , then by (4)  $\sum_\alpha \delta_{\alpha i}^H = \sum_\alpha \delta_{\alpha j}^H = 1$  in either case. Hence, the non-intersection constraints express exactly what we want.

Let  $w_i \geq 0$  be a measure of the value that accrues if box  $B_i$  is put into the big box  $B_0$ . To maximize the total value of the boxes put into  $B_0$  the objective function is

$$\text{maximize} \quad \sum_{i=1}^n w_i \left( \sum_H \delta_{1i}^H \right). \quad (6)$$

The decision to include or not a small box into  $B_0$  is part of the mixed-integer programming model. Typically, the weight reflects some economic value of packing a small box into the big one, as might be the case e.g. with the cargo of a spacecraft or of a ship container. If  $w_i = 1$  for all  $i = 1, \dots, n$  then the model seeks to include the largest number of small boxes that fit into the big one.

The model has  $3n$  flow variables,  $3n$  zero-one variables  $\delta$  for orthogonal placement and a total of  $3n(n - 1)$  zero-one variables  $\lambda$  for the “left-right” positioning of the small boxes in the big box. It has  $4n$  equations and  $n$  inequalities for the orthogonality requirement,  $6n$  domain constraints,  $3n(n - 1)$  non-intersection constraints and  $3n(n - 1)/2$  logical constraints.

The model permits the user to prescribe the packing and the positioning of certain small boxes in the big one. This is done by fixing variables. E.g. if axis  $\alpha$  of box  $B_i$  is to be aligned with an “up” direction, then  $\delta_{\alpha i}^Z = 1$  expresses this requirement (provided that the  $Z$  axis corresponds to the “up” direction of  $B_0$ ). Given a solution with  $\delta_{\alpha i}^Z = 1$ , a condition like “This side up” can then be taken into consideration in a post-processor step.

The model permits the user also to incorporate static balancing constraints. Let  $m_i$  be the mass of box  $B_i$  which we may assume to be uniformly distributed over  $B_i$ . More precisely, we assume that the geometric center of gravity of box  $B_i$  equals (or is sufficiently close to) the center of gravity of the mass of  $B_i$ . Otherwise, the center of gravity of the mass of box  $B_i$  is a linear translation of the geometric center of gravity that has to be accounted for. To balance the cargo in  $B_0$ , the coordinates of the center of gravity of the total mass loaded into  $B_0$  must fall between lower and upper limits  $lo^H$ ,  $up^H$ , which are typically in the vicinity of the geometric center of

gravity of  $B_0$ . This gives six constraints of the form

$$\ell o^H \sum_{i=1}^n m_i \left( \sum_H \delta_{1i}^H \right) \leq \sum_{i=1}^n m_i (x_i^H + \Delta_i^H) \leq up^H \sum_{i=1}^n m_i \left( \sum_H \delta_{1i}^H \right)$$

where  $\Delta_i^H$  is the difference of the coordinates of the two centers of gravity for each box and coordinate direction  $H \in \{X, Y, Z\}$  ( $\Delta_i^H = 0$  in most cases). For more on this problem see Fasano, "Cargo analytical integration in space engineering: a three-dimensional model", in Ciriani *et al* (eds), *Operational Research in Industry*, MacMillan, 1999, and Padberg, "Packing small boxes into a big box", to appear in *Mathematical Methods of Operations Research*, 52 (2000).

## A. Short-Term Financial Management

One of the tasks of a financial officer of a large corporation involves the management of *cash* and related financial instruments on a short-term basis so as to produce revenues for the corporation that otherwise would go to the banks. By investing excess cash into marketable securities (MS) of various kinds revenue can be generated, selling such short-term investments prior to their maturities can alleviate financial stress. Payments on accounts payables typically carry *terms of credit* such as a discount of  $x\%$  if paid within a certain (limited) number of days, the full amount otherwise within a longer time period. Conventional wisdom has it that discount should be taken at any cost, but the short-term cost of capital to the firm due to other reasons may well exceed the benefits to be derived from the discount. Thus the financial officer must decide on what portion of the outstanding payables discounts should be taken. Other possibilities to alleviate financial stress consists of borrowing against e.g. an *open line of credit* that most companies have with their respective banks. In times of excess cash the question of debt retirement must be weighed in. All of this is to be done with a reasonably short- to medium-term time perspective and thus we have to consider several time periods. While the immediate present, e.g. the next two weeks, require a detailed planning, the decisions to be taken two to three months from now impact the present, but in a less direct way. To capture their impact it suffices to aggregate the future periods into time periods consisting of several days or weeks, with the corresponding financial data being approximated by their totals for the respective time periods. Your job is to determine the optimal cash management decisions for your firm using a linear programming model similar to that of Y.Orgler, "An Unequal Period Model for Cash Management Decisions", *Management Science*, 16(1969) B77-B92. The essentials of the problem have **not** changed since that time and we recommend that you get a copy of that article to do this case.

Suppose that you have chosen a four month time horizon, divided unequally into four periods of 10, 20, 30 and 60 days, respectively; i.e. period 1 has 10 days, period 2 has 20 days, etc.

The portfolio of marketable securities held by your firm by the beginning of the first period includes 5 different sets of securities with face values \$100, \$75, \$750, \$600 and \$900. The first four securities mature in periods one to four, respectively, while the last one matures beyond the horizon. All marketable securities can be sold prior to maturity at a discount, i.e. at a loss for your firm. Furthermore, in period 2 an *additional* amount of \$100 in marketable securities matures (not included in \$75) which, however, has been "earmarked" for other purposes. The only short term financing source (other than the sale of marketable securities from the initial portfolio) is represented by a \$850 open line of credit. Under the agreement with the bank, loans can be obtained at the beginning of any period and are due after one year at a monthly interest rate of 0.7 percent. Early repayments are not permitted. The costs of taking a loan that are relevant for measuring the performance of the cash management decisions are the  $F_j$ ,  $j=1,2,3,4$ , in Table A.1.

The payment decisions to be considered by you correspond to accounts payable with 2 percent 10 days, net 30 days terms of credit or for short, 2-10/N-30. All obligations prior to period 1 have been met so that liability applies only within the time horizon of the four months and the firm does not want to postpone any payments beyond the horizon. Predicted purchases of periods 1,2,3,4, total \$400, \$650, \$1,400 and \$2,300, respectively. It is assumed that all bills are received in the first half of the respective periods and that payments are made at the beginning of the

**Table A.1.** Cost and revenue coefficients

<u>Payments</u>	<u>Securities</u>		<u>Line of credit</u>
	<u>Purchases</u>	<u>Sales</u>	
$C_{12} = 0.0204$	$D_{21} = 0.0010$	$E_{21} = 0.0020$	$F_1 = 0.0280$
$C_{13} = 0.0000$	$D_{31} = 0.0040$	$E_{31} = 0.0050$	$F_2 = 0.0257$
$C_{22} = 0.0204$	$D_{41} = 0.0080$	$E_{41} = 0.0100$	$F_3 = 0.0210$
$C_{23} = 0.0000$	$D_{51} = 0.0160$	$E_{51} = 0.0200$	$F_4 = 0.0140$
$C_{33} = 0.0204$	$D_{32} = 0.0025$	$E_{32} = 0.0037$	
$C_{34} = 0.0000$	$D_{42} = 0.0070$	$E_{42} = 0.0087$	
$C_{44} = 0.0204$	$D_{52} = 0.0150$	$E_{52} = 0.0190$	
	$D_{43} = 0.0040$	$E_{43} = 0.0050$	
	$D_{53} = 0.0120$	$E_{53} = 0.0150$	
	$D_{54} = 0.0080$	$E_{54} = 0.0100$	

**Table A.2.** Input data-requirement vector (in thousand \$)

<u>Accounts Payable</u>	<u>Marketable Securities</u>	<u>Net Fixed Cash Flows</u>
	<u>in Initial Portfolio</u>	
$L_1 = \$400$	$S_1 = \$100$	$N_1 = \$ - 1,000$
$L_2 = \$650$	$S_2 = \$75$	$N_2 = \$ - 1,500$
$L_3 = \$1,400$	$S_3 = \$750$	$N_3 = \$2,000$
$L_4 = \$2,300$	$S_4 = \$600$	$N_4 = \$4,500$
	$S_5 = \$900$	
<u>Line of Credit:</u> $R=\$850$	<u>Minimal Cash Balance:</u> $M_j=\$0$ for $j=1,2,3,4$	
<u>Beginning Cash Balance:</u> $B_0=\$100$	<u>Average Daily Cash Balance:</u> $A=\$100$	

periods. (See the coefficients  $C_{ij}$  in Table A.1 for more detail: Any portion of the “bundle” of bills in period 1 can be paid either at the beginning of period 2 with a discount of 2%, i.e.  $C_{12} = 0.0204$ , or it can be paid at face value with no discount in period 3, i.e.  $C_{13} = 0.0$ . It remains to be decided upon what part of this bundle of bills to pay in period 2 or in period 3 etc.)

The costs and revenues associated with the transactions in marketable securities are displayed in the  $D_{ij}$  and  $E_{ij}$  columns of Table A.1. For instance, investing \$100 in the first period into marketable securities that mature in period 4 yields a revenue of  $\$D_{41} \cdot 100$  or 80 cents, whereas obtaining \$100 by selling marketable securities that mature beyond the horizon already in period 2 costs  $\$E_{52} \cdot 100$  or \$1.90 in terms of lost yield and transaction costs.

Finally, suppose that the initial cash balance is \$100 and that the net fixed (or exogenous) cash flows are expected to equal  $-\$1,000$ ,  $-\$1,500$ ,  $\$2,000$ , and  $\$4,500$  in periods 1,2,3 and 4, respectively. The minimum cash balance requirement is \$0 for all four periods. Also you wish to incorporate a requirement that the average daily cash balance be at least \$100.

**Table A.3.** Rule-based balanced cash budget

	Period 1	Period 2	Period 3	Period 4
Cash Balance BoP	100.00	0.00	0.00	0.00
Total Receipts	100.00	2,299.90 <sup>1)</sup>	2,000.00	4,500.00
Total Cash Available	200.00	2,299.90	2,000.00	4,500.00
Total Disbursements	1,000.00	2,349.90 <sup>2)</sup>	1,554.76 <sup>3)</sup>	2,254.00 <sup>4)</sup>
Cash Balance EoP	(800.00)	(50.00)	445.24	2,246.00
Minimum Cash Balance	0.00	0.00	0.00	0.00
Excess (Shortage)	(800.00)	(50.00)	445.24	2,246.00
Invest	0.00	0.00	445.24	2,046.00 <sup>5)</sup>
Borrow	800.00	50.00	0.00	0.00

1) From selling marketable securities: 75.00+747.23+594.78+882.90.

2) Net fixed cash flow of \$1,500 plus 849.90 against Accounts Payable.

3) Liability  $L_3$  with discount plus outstanding amounts on liabilities  $L_1$  and  $L_2$ .

4) Liability  $L_4$  with discount.

5) To maintain an average daily cash balance of \$100.

Follow meticulously the following suggestions to organize your work and answer all the question with supporting numerical analysis.

1. Please use the notation (symbols, etc.) of Orgler's paper cited above. Write down explicitly the definitions of all variables, the objective function and the constraints of a linear programming formulation of the above problem.
2. Discuss several possibilities to "balance" the cash budget in periods 1,2,3 and 4 in terms of their respective cost, i.e. find at least one feasible solution to the linear program "by hand"; see the addendum for advice.
3. How does a minimum cash balance requirement of \$100 in all four periods change your formulation obtained under question 1?
4. Solve the linear programming problem using any standard linear programming package with and without an average daily cash balance requirement. Solve the linear program with a minimum cash balance requirement of \$100 in all four periods. How much does the "earmarking" of \$100 of marketable securities in period 2 cost your firm? Summarize your findings in words and discuss briefly the dual solution to your linear program from a managerial point of view.
5. Making suitable assumptions about liabilities (all liabilities incurred on the first day of a period, discount goes from the 2nd day to the number of days specified) how does your formulation change if your terms of credit are 1-10/N-60 in lieu of the ones mentioned in the text? How do the objective function coefficients change? What else must change in the linear programming formulation? What happens if you assume that 50% of your bills in each period have the terms of credit 1-10/N-60 and the remaining 50% the terms of credit 2-10/N-30?

**Addendum:** The following serves as an example for “balancing the cash budget” in the cash management task. Suppose your company has adopted the following cash budgeting rules:

1. Take all discounts (if at all possible).
2. Use your line of credit fully.
3. Delay selling marketable securities as long as possible, but sell if necessary to get the discount.

With terms of credit of 2-10/N-30 this results in the cash budget shown in Table A.3. The objective function value of this cash budget is \$64.62 where the revenue =  $91.308 + 21.711 = 113.019$  and the cost =  $23.685 + 24.714 = 48.399$ , which is not optimal. BoP stands for “beginning of the period” and EoP likewise.

## A.1 Solution to the Cash Management Case

1. Following Orgler's notation we define the following variables

$x_{gj}$	: amount paid in period $j$ for liabilities incurred in period $g$
$y_{ij}$	: amount invested in period $j$ in securities maturing in period $i$
$z_{ij}$	: amount sold in period $j$ from securities maturing in period $i$
$w_g$	: amount borrowed in period $g$
$b_j$	: cash balance in time period $j$

For completeness, we repeat the needed notation from that paper:

$A$	: average cash balance required over all periods
$C_{gj}$	: net return per dollar allocated to payment $x_{gj}$
$D_{ij}$	: net return from investment in marketable securities $y_{ij}$
$E_{ij}$	: lost return from sold securities $z_{ij}$
$F_g$	: cost per dollar borrowed in period $g$
$L_g$	: total amount of liability incurred in period $g$
$N_j$	: fixed net cash flows (other receipts less other payments in period $j$ )
$R$	: total amount of short-term financing available
$S_i$	: total maturity value of initial securities maturing in period $i$
$a_{gj}$	= $1 + C_{gj}$
$d_{ij}$	= $1 + D_{ij}$
$e_{ij}$	= $1 + E_{ij}$

For our problem the various constraints are as follows.

**Payments:** The general form of the constraints for the payments is

$$\sum_{j=g}^4 a_{gj} x_{gj} = L_g \quad \text{for } g = 1, \dots, 4.$$

E.g. for  $g = 1$  we know that the liabilities incurred in period 1 can be paid in period 2 with a discount of 2% or in period 3 at face value (see the first column of Table A.1). Therefore we have the constraint  $1.0204x_{12} + x_{13} = 400$ .

**Line of credit:** The total amount of short term financing over the four periods should not exceed the available line of credit, i.e.,

$$w_1 + w_2 + w_3 + w_4 \leq 850.$$

**Securities sales:** The general form of these constraints is

$$\sum_{j=1}^{i-1} e_{ij} z_{ij} \leq S_i \quad \text{for } i = 2, \dots, 5.$$

E.g. for  $i = 2$  we know from Table A.1 that the securities maturing in period 2 can be sold in period 1 at a discount, yielding  $\frac{1}{e_{21}}$  of the face value. Thus the corresponding constraint is  $1.002z_{21} \leq 75$ .

**Cash balance:** The average balance requirement is imposed by the following constraint

$$10b_1 + 20b_2 + 30b_3 + 60b_4 \geq 12000.$$

**Cash flows:** The cash-flows constraint is in general form:

$$b_g - b_{g-1} = \sum_{i=g+1}^5 z_{ig} + w_g - \sum_{j=1}^g x_{gj} - \sum_{i=g+1}^5 y_{ig} + S_g - \sum_{j=1}^{g-1} e_{gj} z_{gj} - \sum_{j=1}^{g-1} d_{gj} y_{gj} + N_g$$

which e.g. for  $g = 1$  gives the following constraint

$$b_1 - 100 = z_{21} + z_{31} + z_{41} + z_{51} + w_1 - y_{21} - y_{31} - y_{41} - y_{51} + 100 - 1000.$$

**Objective function:** The objective is to maximize the net revenue over all periods, i.e.,

$$\max \sum_{j=1}^4 C_{gj} x_{gj} + \sum_{j=1}^4 \sum_{i=j+1}^5 (D_{ij} y_{ij} - E_{ij} z_{ij}) - \sum_{g=1}^4 F_g w_g.$$

All the variables in our model are nonnegative. The linear programming problem (in CPLEX lp format) is

maximize

$$\begin{aligned} & 0.0204 x_{12} + 0.0204 x_{22} + 0.0204 x_{33} + 0.0204 x_{44} \\ & + 0.001 y_{21} + 0.004 y_{31} + 0.0025 y_{32} + 0.008 y_{41} + 0.007 y_{42} \\ & + 0.004 y_{43} + 0.016 y_{51} + 0.015 y_{52} + 0.012 y_{53} + 0.008 y_{54} \\ & - 0.002 z_{21} - 0.005 z_{31} - 0.0037 z_{32} - 0.01 z_{41} - 0.0087 z_{42} \\ & - 0.005 z_{43} - 0.02 z_{51} - 0.019 z_{52} - 0.015 z_{53} - 0.01 z_{54} \\ & - 0.028 w_1 - 0.0257 w_2 - 0.021 w_3 - 0.014 w_4 \end{aligned}$$

subject to

$$(1) \quad 1.0204 x_{12} + x_{13} = 400$$

```

(2) 1.0204 x22 + x23 = 650
(3) 1.0204 x33 + x34 = 1400
(4) 1.0204 x44 = 2300
(5) w1 + w2 + w3 + w4 <= 850
(6) 1.002 z21 <= 75
(7) 1.005 z31 + 1.0037 z32 <= 750
(8) 1.01 z41 + 1.0087 z42 + 1.005 z43 <= 600
(9) 1.02 z51 + 1.019 z52 + 1.015 z53 + 1.01 z54 <= 900
(10) 10 b1 + 20 b2 + 30 b3 + 60 b4 >= 12000
(11) b1 - z21 - z31 - z41 - z51 + y21 + y31 + y41 + y51 - w1 = -800
(12) b2 - b1 + x12 + x22 - z32 - z42 - z52 + y32 + y42 + y52 - w2
    - 1.001 y21 + 1.002 z21 = -1425
(13) b3 - b2 + x13 + x23 + x33 - z43 - z53 + y43 + y53 - w3
    + 1.005 z31 + 1.0037 z32 - 1.004 y31 - 1.0025 y32 = 2750
(14) b4 - b3 + x34 + x44 + y54 - z54 - w4 + 1.01 z41 + 1.0087 z42
    + 1.005 z43 - 1.008 y41 - 1.007 y42 - 1.004 y43 = 5100
end

```

**2.** Any feasible solution to the linear program given above “balances” the budget. The solution suggested in the Addendum (page 383) corresponds to the following solution in our variables:  $x_{12} = 392$ ,  $x_{22} = 457.9$ ,  $x_{23} = 182.76$ ,  $y_{53} = 445.24$ ,  $y_{54} = 2046$ ,  $z_{32} = 747.23$ ,  $z_{42} = 594.78$ ,  $z_{52} = 882.9$ ,  $w_1 = 800$ ,  $w_2 = 50$  while all other variables are zero. One verifies easily that for this solution the objective value is indeed 64.62. It should be noted that there are more than one solution corresponding to the one provided in the addendum. In particular, the variables  $x_{12}$ ,  $x_{22}$ ,  $x_{13}$ ,  $x_{23}$ , should satisfy the following constraints  $x_{12} + x_{22} = 849.9$ ,  $1.0204x_{12} + x_{13} = 400$  and  $1.0204x_{22} + x_{23} = 650$ . Since variables  $x_{12}$  and  $x_{22}$  have the same coefficient in the objective function, it suffices to set  $x_{13} = 0$  to get a feasible solution. Also the variables  $y_{43}$  and  $y_{53}$  should satisfy  $y_{43} + y_{53} = 445.24$ . Since  $y_{53}$  has a bigger coefficient in the objective function and we are maximizing, we set  $y_{43} = 0$ .

It is interesting to see what the **minimum** net return over all periods turns out to be. Suppose you borrow the full amount of credit in period 1 (since it is more expensive), sell all securities in the initial portfolio at cost in period 1, and make all payments in periods where there is no discount. All these are irrational decisions, but, remember, we want to minimize the net return! These decisions correspond to the following variable settings:  $w_1 = 850$ ,  $x_{13} = 400$ ,  $x_{23} = 650$ ,  $x_{34} = 1400$ ,  $x_{44} = 2254.02$ ,  $z_{21} = 74.85$ ,  $z_{31} = 746.27$ ,  $z_{41} = 594.06$ ,  $z_{51} = 882.35$ , and, of course, all  $y$  variables are set to zero. This yields  $b_1 = 2347.53$ ,  $b_2 = 847.53$ ,  $b_3 = 1797.53$ , and  $b_4 = 2643.51$  and the net return is  $-5.29$ . Thus the worst case scenario is that the company will lose 5.29 thousand dollars. Of course, we can calculate the corresponding balanced cash budget under this scenario by simply minimizing the objective function of the above linear program and then interpret the numerical output – which is precisely what we did.

**3.** To model the requirement for a minimum cash balance of \$100 for each period, we add the following constraints to our problem:

$$b_j \geq 100 \quad \text{for } j = 1, \dots, 4.$$

Notice that in this case the cash balance constraint of part 1 is implied by these new constraints and thus may be deleted from the linear program.

**Table A.4.** Optimal solutions for LP1, LP2, and LP3

Objective	65.70	67.27	63.89
$x_{12}$	211.64	392.00	392.00
$x_{22}$	637.01	457.49	357.39
$x_{13}$	184.04		
$x_{23}$		183.17	285.32
$x_{33}$	1372.01	1372.01	1372.01
$x_{44}$	2254.02	2254.02	2254.02
$y_{21}$		82.35	
$y_{43}$	443.95	444.81	342.67
$y_{54}$	2604.44	2692.58	2590.02
$z_{32}$	747.24	747.24	747.24
$z_{41}$	594.06		17.65
$z_{42}$		594.83	577.16
$z_{51}$	882.35	882.35	882.35
$w_2$	850.00	850.00	850.00
$b_1$	676.41		100.00
$b_2$			100.00
$b_3$			100.00
$b_4$	87.26		100.00

4. We use CPLEX to solve the linear programming problems described in the previous parts. We refer to the problem of part 1 as LP1. LP2 is the LP1 after dropping the cash balance constraint (10). LP3 is the model of part 3. In Table A.4 we summarize our findings.

The optimal solution for LP1, i.e., the model of part 1, generates a net revenue of 65.70 thousand dollars. One can break down the objective function to its components as follows. The net revenue is broken down to a total revenue of 113.90 thousand dollars (payments and investments) and a total cost of 48.19 thousand dollars (sale of securities prior to maturity and use of credit line). 91.29 thousand dollars of the total revenue are generated by payments ( $x_{gj}$  variables) and 22.61 thousand dollars from investments in marketable securities ( $y_{ij}$  variables). 26.34 thousand dollars of the total cost are incurred from selling marketable securities prior to their maturity ( $z_{ij}$  variables), and 21.85 thousand dollars are incurred from the use of the credit line ( $w_g$  variables). Table A.5 shows the balanced cash budget for LP1 annotated like in the Addendum.

From the run LP2 we find that the requirement for an average daily cash balance of \$100 costs the company  $67.27 - 65.70 = 1.57$  thousand dollars for the 6-month period, while from the run of LP3 we see that the requirement for an end of period cash balance of \$100 for each period costs the company  $67.27 - 63.89 = 3.38$  thousand dollars for the same period.

**Table A.5.** Optimal balanced cash-budget

	Period 1	Period 2	Period 3	Period 4
Cash Balance BoP	100.00	676.41	0.00	0.00
Total Receipts	1,576.41 <sup>1)</sup>	822.24 <sup>3)</sup>	2,000.00	4,945.72 <sup>6)</sup>
Total Cash Available	1,676.41	1,498.65	2,000.00	4,945.72
Total Disbursements	1,000.00	2,348.65 <sup>4)</sup>	1,556.05 <sup>5)</sup>	2,254.02 <sup>4)</sup>
Cash Balance EoP	676.41	(850.00)	443.95	2,691.70
Minimum Cash Balance	0.00	0.00	0.00	0.00
Excess (Shortage)	676.41 <sup>2)</sup>	(850.00)	443.95	2,691.70 <sup>2)</sup>
Invest	0.00	0.00	443.95	2,604.44
Borrow	0.00	850.00	0.00	0.00

<sup>1)</sup> From selling marketable securities (MS):  $z_{41} = 594.06$  and  $z_{51} = 882.35$  early.

<sup>2)</sup> To maintain an average daily balance of \$100.

<sup>3)</sup> \$75 from maturing MS and  $z_{32} = 747.24$  from selling MS early.

<sup>4)</sup> Net fixed cash flow plus \$848.65 against Accounts Payable.

<sup>5)</sup> Payments  $x_{13} = 184.04$  plus  $x_{33} = 1,372.01$  against Accounts Payable.

<sup>6)</sup> Net fixed cash flow \$4,500 plus maturing MS from investment in Period 3 of \$445.72.

The optimal **dual variables** for the run LP1 in the order of the above constraints (1), ..., (14) are as follows (we round to four digits after the point):  $-0.0120, -0.0120, 0.0082, 0.0122, 0.0070, 0.0000, 0.0168, 0.0158, 0.0137, -0.0001, 0.0340, 0.0327, 0.0120, 0.0080$ . The values of these variables of the linear programming problem can be used for guiding several managerial decisions. For the payment constraints (constraints (1)-(4) in the model of part 1) the dual variables are unrestricted in sign. A positive value indicates an increase in the net revenue for an increase of payments in the corresponding period. Therefore, the net return would increase with payments on purchases scheduled in the last two periods. The dual variables of these constraints can be used for scheduling purchases and to evaluate managerial decisions such as the “stretching” of payments, i.e., the postponement of payments beyond their due date as given by the terms of credit. Such behavior on the part of the company can of course, lead to a loss of goodwill on the part of the suppliers, but it is a managerial choice anyway. More precisely, in the current scenario stretching or postponing \$100 worth of payments on liabilities due in period 2 to period 3 can be expected to save the company about  $-100 \times (-0.0120) + 100 \times 0.0082 = 2.02$  thousands of dollars.

For the borrowing constraint (constraint (5) in the model of part 1) the dual variable is positive only when all available credit line has been exhausted. Then this dual variable suggests the interest that we are willing to pay to get extra credit given equal conditions otherwise. In the case of the model of part 1 this dual variable is  $\delta = 0.007$ . This indicates that the current arrangement with the bank is favorable for the company, i.e., the company may be willing to pay up to 0.7% per 120 days (or roughly 0.175% per month) more in interest to the bank, since any additional dollar obtained from the bank given the current conditions increases the profit from cash management by 0.7 thousand dollars in a certain range.

The dual variables of the securities sales constraints (constraints (6)-(9) in the model of part 1) equal  $\gamma_1 = 0.0000$ ,  $\gamma_2 = 0.0168$ ,  $\gamma_3 = 0.0158$ ,  $\gamma_4 = 0.0137$  in the run LP1 and give the increase in the net return if the corresponding amount of maturing securities increases. However, the amount  $S_i$  of maturing marketable securities affect the corresponding cash flow constraints as well. "Earmarking" of \$100 worth of marketable securities in a particular period means that both the right-hand side of the corresponding security-sales constraints and the right-hand side of the corresponding cash flow constraints will increase by \$100. Thus, the earmarking of \$100 of marketable securities maturing in period 2 costs the company roughly  $100(0.0000 + 0.0327) = 3.27$  thousand dollars in the current scenario.

The dual variable of the average cash balance constraint (constraint (10) in the model of part 1) is nonpositive indicating that increasing the average cash balance will decrease the net return since it will reduce the investment opportunities or increase financing costs.

Finally, the dual variables of the cash flow constraints (constraints (11)-(14) in the model of part 1) give the increase in the net return resulting from an increase in the net cash flows. The values can be used to evaluate the effects of changes in the terms of trade credit on accounts receivable or a change in the method of collecting receivables.

**5.** Changing the terms of credit from 2-10/N-30 to 1-10/N-60 has the following implications to the model of part 1:

- Since payments for liabilities incurred up to 60 days before are allowed, we need to introduce variables  $x_{14}$  and  $x_{24}$ .
- The objective function coefficients of the variables  $x_{12}$ ,  $x_{22}$ ,  $x_{33}$ , and  $x_{44}$  have to change to  $1/(1 - 0.01) - 1 = 0.0101$ , and their coefficients  $a_{gj}$  have to change to  $1/(1 - 0.01) = 1.0101$ .

The LP of part 1 will therefore become:

```

maximize
  0.0101 x12 + 0.0101 x22 + 0.0101 x33 + 0.0101 x44
  + 0.001 y21 + 0.004 y31 + 0.0025 y32 + 0.008 y41 + 0.007 y42
  + 0.004 y43 + 0.016 y51 + 0.015 y52 + 0.012 y53 + 0.008 y54
  - 0.002 z21 - 0.005 z31 - 0.0037 z32 - 0.01 z41 - 0.0087 z42
  - 0.005 z43 - 0.02 z51 - 0.019 z52 - 0.015 z53 - 0.01 z54
  - 0.028 w1 - 0.0257 w2 - 0.021 w3 - 0.014 w4
subject to
(1) 1.0101 x12 + x13 + x14 = 400
(2) 1.0101 x22 + x23 + x24 = 650
(3) 1.0101 x33 + x34 = 1400
(4) 1.0101 x44 = 2300
(5) w1 + w2 + w3 + w4 <= 850
(6) 1.002 z21 <= 75
(7) 1.005 z31 + 1.0037 z32 <= 750
(8) 1.01 z41 + 1.0087 z42 + 1.005 z43 <= 600
(9) 1.02 z51 + 1.019 z52 + 1.015 z53 + 1.01 z54 <= 900
(10) 10 b1 + 20 b2 + 30 b3 + 60 b4 >= 12000
(11) b1 - z21 - z31 - z41 - z51 + y21 + y31 + y41 + y51 - w1 = -800
(12) b2 - b1 + x12 + x22 - z32 - z42 - z52 + y32 + y42 + y52 - w2
     - 1.001 y21 + 1.002 z21 = -1425

```

```
(13) b3 - b2 + x13 + x23 + x33 - z43 - z53 + y43 + y53 - w3
     + 1.005 z31 + 1.0037 z32 - 1.004 y31 - 1.0025 y32 = 2750
(14) b4 - b3 + x14 + x24 + x34 + x44 + y54 - z54 - w4 + 1.01 z41
     + 1.0087 z42 + 1.005 z43 - 1.008 y41 - 1.007 y42
     - 1.004 y43 = 5100
```

end

Using CPLEX to solve this LP we find that the maximum net return with the new credit terms is 26.69 thousand dollars. Thus, changing the credit terms from 2-10/N-30 to 1-10/N-60 results to a loss of  $65.70 - 26.69 = 39.01$  thousand dollars. This is so, mainly because the discount for early payments is halved for 2% to 1%. The optimal solution is the following:  $x_{33} = 1386.00$ ,  $x_{44} = 2277.00$ ,  $y_{43} = 614.00$ ,  $y_{54} = 1789.45$ ,  $z_{31} = 523.59$ ,  $z_{32} = 222.97$ ,  $z_{41} = 594.06$ ,  $z_{51} = 882.35$ ,  $w_2 = 2.03$ ,  $x_{14} = 400.00$ ,  $x_{24} = 650.00$ ,  $b_1 = 1200.00$ .

If we have two types of payments, 2-10/N-30 and 1-10/N-60, we have to introduce a third index  $h$  to our payment variables. Let  $x_{hgj}$  be the amount paid in period  $j$  for liabilities incurred in period  $g$  with the  $h$ -th type of payment. In our case,  $h$  takes the values 1 and 2, indicating terms of credit 2-10/N-30 and 1-10/N-60, respectively. The other changes that have to be made to the model of part 1 are clear. First, we have two groups of constraints (1)-(4), one for each type of payments. Since 50% of the liability  $L_g$  is paid with the first type of payment, and the rest with the second, we have  $L_{1g} = L_{2g} = L_g/2$ . Second, we replace  $x_{gj}$  by  $x_{1gj} + x_{2gj}$  wherever both  $x_{1gj}$  and  $x_{2gj}$  are properly defined. The linear program is as follows:

maximize

$$\begin{aligned} & 0.0204 x_{112} + 0.0204 x_{122} + 0.0204 x_{133} + 0.0204 x_{144} \\ & + 0.0101 x_{212} + 0.0101 x_{222} + 0.0101 x_{233} + 0.0101 x_{244} \\ & + 0.001 y_{21} + 0.004 y_{31} + 0.0025 y_{32} + 0.008 y_{41} + 0.007 y_{42} \\ & + 0.004 y_{43} + 0.016 y_{51} + 0.015 y_{52} + 0.012 y_{53} + 0.008 y_{54} \\ & - 0.002 z_{21} - 0.005 z_{31} - 0.0037 z_{32} - 0.01 z_{41} - 0.0087 z_{42} \\ & - 0.005 z_{43} - 0.02 z_{51} - 0.019 z_{52} - 0.015 z_{53} - 0.01 z_{54} \\ & - 0.028 w_1 - 0.0257 w_2 - 0.021 w_3 - 0.014 w_4 \end{aligned}$$

subject to

- (1)  $1.0204 x_{112} + x_{113} = 200$
- (2)  $1.0204 x_{122} + x_{123} = 325$
- (3)  $1.0204 x_{133} + x_{134} = 700$
- (4)  $1.0204 x_{144} = 1150$
- (1a)  $1.0101 x_{212} + x_{213} + x_{214} = 200$
- (2a)  $1.0101 x_{222} + x_{223} + x_{224} = 325$
- (3a)  $1.0101 x_{233} + x_{234} = 700$
- (4a)  $1.0101 x_{244} = 1150$
- (5)  $w_1 + w_2 + w_3 + w_4 \leq 850$
- (6)  $1.002 z_{21} \leq 75$
- (7)  $1.005 z_{31} + 1.0037 z_{32} \leq 750$
- (8)  $1.01 z_{41} + 1.0087 z_{42} + 1.005 z_{43} \leq 600$
- (9)  $1.02 z_{51} + 1.019 z_{52} + 1.015 z_{53} + 1.01 z_{54} \leq 900$
- (10)  $10 b_1 + 20 b_2 + 30 b_3 + 60 b_4 \geq 12000$
- (11)  $b_1 - z_{21} - z_{31} - z_{41} - z_{51} + y_{21} + y_{31} + y_{41} + y_{51} - w_1 = -800$
- (12)  $b_2 - b_1 + x_{112} + x_{212} + x_{122} + x_{222} - z_{32} - z_{42} - z_{52} + y_{32}$

$$\begin{aligned}
 & + y_{42} + y_{52} - w_2 - 1.001 y_{21} + 1.002 z_{21} = -1425 \\
 (13) \quad b_3 - b_2 + x_{113} + x_{213} + x_{123} + x_{223} + x_{133} + x_{233} - z_{43} - z_{53} \\
 & + y_{43} + y_{53} - w_3 + 1.005 z_{31} + 1.0037 z_{32} - 1.004 y_{31} \\
 & - 1.0025 y_{32} = 2750 \\
 (14) \quad b_4 - b_3 + x_{214} + x_{224} + x_{134} + x_{234} + x_{144} + x_{244} + y_{54} - z_{54} \\
 & - w_4 + 1.01 z_{41} + 1.0087 z_{42} + 1.005 z_{43} - 1.008 y_{41} \\
 & - 1.007 y_{42} - 1.004 y_{43} = 5100
 \end{aligned}$$

end

Using CPLEX we find that the maximum net return in this case is 46.82 thousand dollars. Thus, changing the credit terms from 2-10/N-30 to 1-10/N-60 for 50% of the bills and 2-10/N-30 for the other 50%, costs  $65.70 - 46.82 = 18.82$  thousand dollars. The optimal solution is:  $x_{112} = 196.00$ ,  $x_{122} = 318.50$ ,  $x_{133} = 686.01$ ,  $x_{144} = 1127.01$ ,  $x_{233} = 693.00$ ,  $x_{244} = 1138.50$ ,  $y_{43} = 620.99$ ,  $y_{54} = 2332.97$ ,  $z_{31} = 523.59$ ,  $z_{32} = 222.97$ ,  $z_{41} = 594.06$ ,  $z_{51} = 882.35$ ,  $w_2 = 516.53$ ,  $x_{214} = 200.00$ ,  $x_{224} = 325.00$ ,  $b_1 = 1200.00$ .

## B. Operations Management in a Refinery

A refinery produces and sells primarily oils, fuels of various grades and quality and side products such as e.g. asphalt. To produce these items from the raw inputs – crude oils of different varieties – energy in the form of hot steam and electricity is required to set the production process and the necessary chemical reactions into motion. Rather than buying the required steam and electricity entirely from an outside supplier – an electricity company, for instance – refineries produce the necessary steam themselves by burning some of the fuel they produce. Burning fuel to produce heat for steam generation reduces the amount of sellable output, but it leaves enough hot steam to be used e.g. for the production of electricity. Indeed, “hot steam” in a refinery is not a homogeneous commodity: when it is produced hot steam is generally “high pressure” steam that has to be “slowed down” to be of use in some parts of the production process. Slowing down steam is done by passing the steam e.g. through turbines, a process from which electricity can be produced as a side product. The production of electricity on its own premises reduces in turn the amount that the refinery has to buy from a supplier. Moreover, in case surplus electricity is generated by the refinery, it becomes a sellable product that can generate revenues just like the refinery’s primary output products oils, fuels, etc.

Thus an integrated refinery complex – i.e. one that has its own burners, boilers and the like for steam generation, turbogenerators, turbines and the like to produce the different kinds of hot steam required, motors for electricity generation, etc – is faced with the problem of trading off consumption of some of its own products against the cost of purchasing the electricity it needs from an outside supplier. Through the generation of electricity an integrated refinery achieves at the same time a higher degree of overall usage of raw energy, and thus of *energy conservation* in the large sense of the word, because of the productive use of residual energy in the form of hot steam that would be lost otherwise.

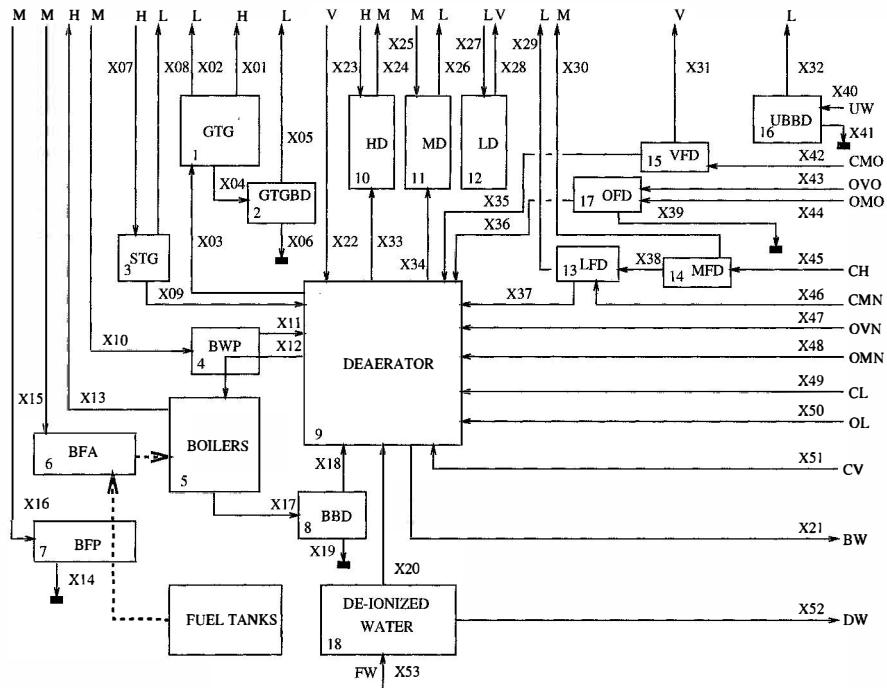
In what follows we describe the steam production, distribution and usage system in a typical refinery and the job at hand is to formulate and solve the problem of optimizing the refinery’s *daily* operations related to steam by linear programming.

The term “production unit” refers in the following to the various devices of the production process of the refinery such as the distillation column, the cracking production unit, the reformer, etc. See also Figure 1.1 of Chapter 1.2.2 on this point and Tables B.1 and B.2 below where the production units are labeled U01,...,U27. Their respective proper functions are immaterial for the problem that we are addressing.

### B.1 Steam Production in a Refinery

The steam production, distribution and usage system in a refinery can be described like in Figure B.1 where you find a flow-chart of the main steam production system.

High pressure steam is produced in the boilers (BOILERS) and the gas turbogenerators (GTG). In the boilers high pressure steam is produced by heating water. The necessary heat is produced by burning fuel oil stored in tanks (FUEL TANKS) which is dispersed by steam in a device called *fuel atomizer* (BFA) after it has been preheated in the fuel preheater (BFP). The water is *de-ionized* water that has gone through a device, called the deaerator (DEAERATOR), and then has been



**Fig. B.1.** The main steam production area

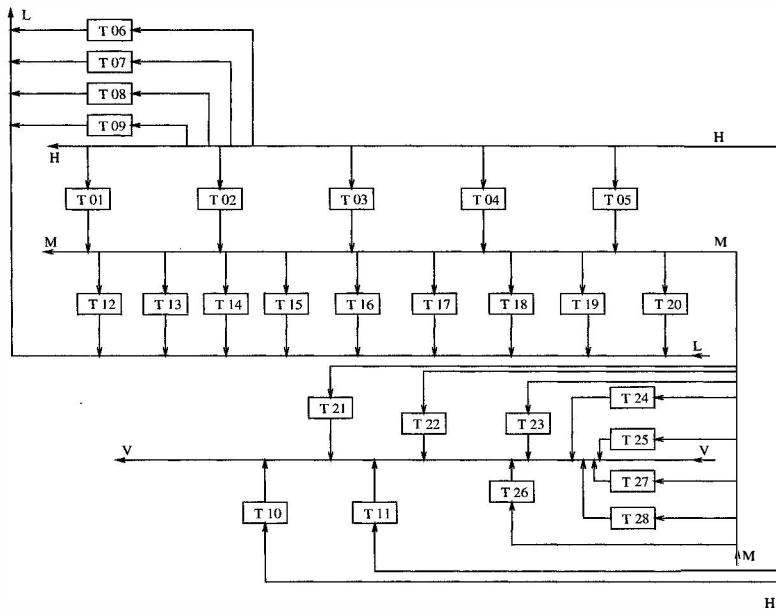
preheated in the device (BWP) to reach the appropriate temperature for usage in the boilers. The de-ionized water (FW) is stored in tanks (DE-IONIZED WATER) and it is available in unlimited quantity at zero cost to the refinery that we consider. Besides steam the boilers yield also some water called “blowdown” which is sent to the blowdown flash drum (BBD). The blowdown is separated into water that returns to the deaerator and some condensate that is drained off. In Figure B.1 and elsewhere in this appendix small solid boxes indicate the drain-off which absorbs the “slack”.

The gas turbogenerators (GTG) get their water from the deaerator. They produce both high and low pressure steam in boilers that burn *fuel gas* instead of fuel oil, while low pressure steam is produced from the blowdown of the boilers of the gas turbogenerators in the device labeled (GTGBD).

Fuel gas is produced in several production units during normal operations.

Electricity is produced by reducing high pressure steam to medium pressure steam. In the steam turbogenerators (STG) high pressure steam is used to produce low pressure steam and electrical power while some water goes to the deaerator.

The rest of the steam is produced in four different ways: by condensate flashing, by “enthalpy” reduction, in the production units and in the turbines. In the medium (MFD), low (LFD) and very low (VFD) condensate flash drums, high and medium pressure condensates from the production units undergo pressure reduction for the production of medium, low and very low pressure steam, respectively. The high (HD), medium (MD) and low (LD) desuperheaters reduce the enthalpy of



**Fig. B.2.** Arrangement of the turbines

incoming high, medium and low pressure steam to produce medium, low and very low steam respectively.

Steam at all levels of pressure is produced and/or consumed in the production units. Several of the production units are equipped with boilers of their own. Low pressure steam is produced in a flash drum labeled UBBD in Figure B.1 from the blowdown of boilers (UW) that are in those production units. Oily condensates that are returned from some of the production units are processed through a condensate flash drum labeled OFD in Figure B.1. Any excess of very low pressure steam is released into the environment.

Finally, the turbines get steam from one level of pressure and produce steam of a lower level while they transform the steam energy into electric power. The refinery has a total of 28 turbines that are labeled T01,...,T28 and that reside in various production units. Their arrangement in the steam circuits is shown in Figure B.2.

The physical layout of the refinery includes four circuits in the form of "pipes" that transport the four different types of steam necessary for the operation of the plant. While transporting steam in the pipes entails some loss of energy, these losses are negligible for the refinery that we consider. The four different circuits feed the required steam into the production units U01,...,U27 and the turbines T01,...,T28. Like in Figure B.2 the production units are connected to the various circuits *in parallel* so that if a turbine or a production unit is "down" the respective circuits are not blocked. In Figures B.1 and B.2 the arcs labeled H, M, L, V indicate the outflow and inflow of high (H), medium (M), low (L) and very low (V) pressure steam from the parts of the refinery that produce the steam. The physical layout of the plant also includes circuits for water and various "condensates" that are generated by the production units. Some boiler water (BW) and some de-ionized water (DW) flows to the production units. Some water (UW) returns from boilers

located in the production units to the blowdown flash drum UBBD and is used to produce more low pressure steam. Sour water (SW) is discharged from some of the production units into the environment at no cost to the refinery. Steam, water and condensates that are generated by the production units are fed into the respective circuits. Due to technological differences the production units U01,...,U09 are referred to as “old” units, while U10,...,U27 are the “new” units. Old production units give some condensates that are different from the ones output by the new ones. In Figure B.1 this difference is indicated by the third letter “O” or “N” respectively, e.g. for the medium pressure and low pressure oily condensates. Wherever the third letter is missing new and old production units can feed the respective inputs. Dotted lines in Figure B.1 indicate fuel transport. In Figure B.3 we show the inflow and outflow of the relevant quantities of steam, water and condensates for production unit 11, the vacuum distillation column, as an example. Not shown in Figure B.3 is the input of electricity and fuel oil for unit 11 as well as all functions of U11 that are irrelevant for the steam balance. The steam, water and condensates production/consumption in all production units is summarized in Table B.1. A positive entry in the table means the total inflow or consumption per hour of the respective quantities given in the 15 columns of the table, a negative entry means total outflow or production per hour if the respective production unit is operating. Consumption and production of the quantities are “pro-rated” linearly if the production unit operates at reduced capacity so that they are zero if it does not operate at all. The labels of the respective columns are explained at the bottom of the table. The units of the amounts of flow given in the table are metric tons per hour or Mt/h.

There are three types of fuels that matter in the context of steam production: fuel gas, fuel oil 1 and fuel oil 2. Fuel gas is produced in several production units of the refinery and is consumed in furnaces of some of the production units and in the gas turbogenerators. Since fuel gas does not have a constant composition, it cannot be sold and thus it must be consumed in the refinery or be burned in the flare. There are four types of furnaces in the production units, those that burn only fuel gas (G), those that can burn fuel gas or fuel oil 1 (GO1), those that can burn fuel gas or fuel oil 2 (GO2) and finally those that burn only fuel oil 1 (O1). The fuel oil used for the main boilers (BOILERS) is fuel oil 1 and needs preheating, while fuel oil 1 used in the production units does not require preheating. The fuel tanks are sufficiently large to handle all operating situations and are replenished automatically. The electricity is produced in the refinery by the gas turbogenerators, the steam turbogenerators and the turbines. Additional electricity is bought from an outside electricity company if necessary. The required steam and the produced electrical power as well as the production unit where the turbine resides are given for the turbines in Table B.3. Whenever a production unit operates at less than 100% of its capacity its turbines are shut off. Table B.2 gives the produced/consumed fuels in the production units as well as their electricity requirements. The same convention for the signs as in Table B.1 applies to Tables B.2 and B.3. Figure B.4 is intended to give you an approximate idea of how energy – in the form of steam, water, fuel, condensates and electricity – “flows” in a refinery. EL stands for electricity generated by the refinery and LO for the electricity bought from an outside supplier. The flow of only some of the condensates is shown in the figure.

## B.2 The Optimization Problem

To “solve the steam balance” in a refinery means to satisfy all the requirements steam in the refinery at minimum cost. Since the costs are additive and linear and all the constraints implied by the requirements are linear, i.e. from an *engineering perspective* we may realistically assume

**Table B.1.** Steam, water and condensates production/consumption per hour in the production units

Unit	H	M	L	V	BW	UW	SW	DW	CH	CM	CL	CV	OM	OL	OV
U01		3.50			4.6		-7.60								
U03		5.80		1.5			-5.60								
U04		16.80								-9.3					
U06		1.80					-1.60								
U07			6.10									-6.1			
U08		5.10		0.2									-12.6		-0.2
U09				1.8			-1.80								
U10	4.4			1.2	7.5							-10.2			
U11	3.1	-12.40	-5.50		39.0	-1.40	-20.60			-0.5	-0.50				
U12	11.5	2.60							-11.5						
U13	4.4	-26.90	2.52		26.0	-1.00				-1.3	-2.40				
U14		0.30	0.10								-0.10				
U15		2.80	52.00		4.3			0.5		-3.1	-55.70			-0.30	
U16					3.8										
U17	1.2	-18.60	-2.90		24.0				-1.4		-1.40				
U18			20.10							-20.10					
U19	0.2	-19.80	0.20		24.5	-1.00	-3.47	0.4		-0.3	-0.20				
U20	-1.30	-19.70	1.20		28.5	-1.28	-5.90				-1.10				
U21	-30.0	10.10	2.82		33.5	-1.30	-29.10				-1.82				
U22							-3.80								
U23			5.40		0.5									-5.90	
U24	4.0		2.80		0.7			1.1	-4.7					-2.80	
U25		5.00										-4.8			
U26	-8.0	-0.13				-0.65		15.6							
U27	7.1	10.90	16.33	2.5					-3.3	-2.1	-7.70	-3.4	-0.6	-8.63	-3.1

H: High pressure steam

M: Medium pressure steam

L: Low pressure steam

V: Very low pressure steam

BW: Boiler water from deaerator

SW: Sour water

UW: Unit boiler blowdown

DW: De-ionized water

CH: High pressure condensate

CM: Medium pressure condensate

CL: Low pressure condensate

CV: Very low pressure condensate

OM: Medium pressure oily condensate

OL: Low pressure oily condensate

OV: Very low pressure oily condensate

**Table B.2.** Fuel/electricity requirements per hour

Unit	FG	FO1	FO2	OFG	OO1	LO
U01		3.60				0.80
U02						0.115
U03		6.40				1.245
U04				1.08		0.82
U05				3.77		
U06		1.36		0.96		0.395
U07	-5.000					0.05
U08	-3.925			0.01		0.22
U10				0.25		0.105
U11			5.65			1.62
U12		1.73				1.27
U13			9.22	0.35		4.35
U14					0.89	1.13
U15	-17.103					0.925
U16						0.093
U17				0.73		0.685
U18						0.205
U19			2.59			1.14
U20			2.03			2.975
U21			2.34			6.33
U22						0.002
U23						0.14
U24	-0.856					0.72
U25		1.59				0.565
U26	-1.578			1.14		0.545
U27				1.02		9.555

FG: Fuel gas

FO1: Fuel gas or fuel oil 1

FO2: Fuel gas or fuel oil 2

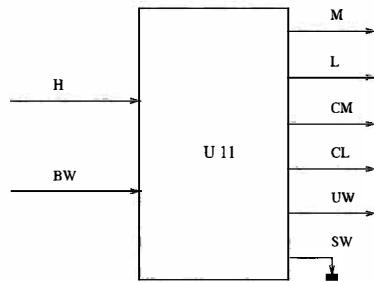
OFG: Exclusively fuel gas

OO1: Exclusively fuel oil 1

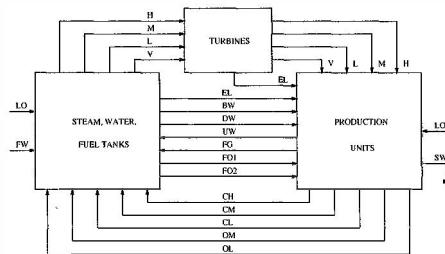
LO: Required electric power

**Table B.3.** Turbines specifications per hour

Turbine Unit	Steam Load (Mt/h)	Load (MW)
<b>H to M</b>		
T01	U19	9.50 0.372
T02	U11	3.30 0.116
T03	U21	9.00 0.280
T04	U26	4.20 0.111
T05	U16	37.00 0.045
<b>H to L</b>		
T06	U19	4.60 0.278
T07	U11	6.60 0.306
T08	U11	4.60 0.204
T09	U20	0.50 0.013
<b>H to V</b>		
T10	U22	8.00 0.460
T11	U22	6.10 0.145
<b>M to L</b>		
T12	U15	7.30 0.164
T13	U13	7.30 0.144
T14	U13	4.80 0.092
T15	U13	3.30 0.052
T16	U20	2.70 0.036
T17	U21	3.30 0.044
T18	U20	2.40 0.030
T19	U21	1.30 0.015
T20	U13	2.10 0.024
<b>M to V</b>		
T21	U12	2.00 0.184
T22	U16	4.70 0.250
T23	U16	2.20 0.100
T24	U16	3.80 0.150
T25	U16	0.90 0.015
T26	U12	0.90 0.013
T27	U22	1.00 0.011
T28	U16	2.00 0.015



**Fig. B.3.** Inflow/outflow for production unit U11



**Fig. B.4.** Major flows of steam, water, fuel, condensates and electricity

that they are, linear programming is an adequate way to formulate the problem and thus, to solve the *daily steam balance*. The **objective function** is the minimization of the operating cost that consists of the cost of purchasing electric energy and the cost of consuming fuels that could be sold. The **variables** of the problem are of three types: the amounts of steam, water and condensates, the amounts of fuel and electricity and 55 zero-one variables that indicate which production units or turbines are operating and which are not. The variables corresponding to the production units can assume fractional values as well to reflect the situation when a production unit is operating at reduced capacity. However from a modeling perspective we assume at first that they are zero-one valued as well. The zero-one variables for the production units are not needed when the refinery operates normally, but their presence in the linear programming model is convenient since it makes it easy to change the program e.g. when a production unit is down or has to be taken down for maintenance. This is of particular importance since this way the linear program can be used on an hourly basis to determine the *valve settings* for the various circuits automatically. All the quantities are on an hourly basis. The units for the amounts of flow are metric tons per hour (Mt/h), the ones for enthalpy are thousands of calories per kilogram (kcal/kg), while the ones for the amounts of power are in millions of Watts (MW).

The **constraints** of the problem are of three general types: mass balance constraints, energy balance constraints and technological constraints. The flow conservation constraints for the mass balance state that total inflow equals total outflow for each one of the “boxes” of Figure B.1. The energy conservation constraints for the energy balance use the enthalpy numbers of Table B.4 to convert the various flows to make them comparable and additive. They are “generalized” flow conservation constraints and the necessary enthalpy numbers are computed from engineering handbooks and formulas. From a theoretical thermodynamics point of view we are making an

important assumption, namely that the quality of the steam and condensates is the same at any point of the respective pipes that carry them. This assumption permits us to work with *constant enthalpy* data, while in theory due to the “flow” in the pipes the flow parameters change and thereby the respective enthalpies. According to the engineers of the refinery where this model was developed the assumption we are making is defendable. In a theoretically more satisfying approach, enthalpy is modeled as being variable as well which leads to nonlinear side constraints; see the references for an approach that deals with the resulting optimization problem. The technological constraints are imposed by technological specifications that are given in the operating manuals of the machinery regarding either the operation or the capacity of the particular machinery. Rather than reproducing the corresponding manuals we summarize this information below.

As a concrete example, let us consider how the three types of constraints for the boxes of Figure B.1 look like in the case of the box BOILERS. For the mass balance we have to state that the amount of high pressure steam that is produced plus the amount of water that flows to the blowdown equals the amount of the incoming water from the deaerator, i.e. (high pressure steam produced) + (water to blowdown) - (incoming water) = 0. Each of the streams has some *enthalpy* which measures the “energy carried” by the stream and which is given as energy units per flow unit in Table B.4. For the specific streams it equals 772, 263 and 130, respectively; see the next section. Moreover, the fuel that is burned in the boilers releases some energy (heat) that is equal 8540 energy units per flow unit of burned fuel. The energy balance equation then is: 772 (high pressure steam produced) + 263 (water to blowdown) - 130 (incoming water) - 8540 (fuel burned) = 0. An example of a technological constraint here is the constraint (water to blowdown) - 0.025 (high pressure steam produced) = 0 which states that the blowdown is 2.5 % of the produced steam.

### B.3 Technological Constraints, Profits and Costs

For the **steam turbogenerator** (STG) the following technological restrictions apply. The steam turbogenerators cannot accept more than 100 Mt/h of high pressure steam. The condensed water(cond. w.) that goes back to the deaerator is at least 3 Mt/h and not more than 60 Mt/h. In order to include the electricity produced in the energy exchange, in lieu of an energy balance we have the following technological constraint:

$$0.122(\text{low press. steam}) + 0.249(\text{cond. w.}) - (\text{electricity}) = 1.143 . \quad (\text{B.1})$$

For the same reason in lieu of energy balances for the **gas turbogenerator** (GTG), we have the following three technological constraints:

$$(\text{fuel gas burned}) - 0.206 (\text{electricity}) = 4.199 , \quad (\text{B.2})$$

$$(\text{low press. steam}) - 0.355 (\text{electricity}) = 12.7 , \quad (\text{B.3})$$

$$(\text{high press. steam}) - 0.617 (\text{electricity}) = 35.2 . \quad (\text{B.4})$$

Both the steam and the gas turbogenerators generate electricity measured in million Watts as indicated by the equations (B.1),...,(B.4). The quantity of the fuel gas that can be burned in the gas

**Table B.4.** The enthalpies of the various streams

Pressure	hot steam	sat. steam	sat. liquid
Steam			
High	772	-	263
Medium	732	665	194
Low	677	656	150
Very low	686	651	134
Condensates			
High			255
Medium			181
Low			144
Very low			126
De-ionized water			10
Steam drained off from OFD			640
Water from OFD to DEAERATOR			100
Water from BBD to DEAERATOR			640
Water from STG to DEAERATOR			40
Water from BWP to BOILERS			130

turbogenerators cannot exceed 10.7 Mt/h. The blowdown of the boilers of the gas turbogenerators is 2.5% of the high pressure steam produced. The **boilers** (BOILERS) produce a minimum of 20 Mt/h and a maximum of 60 Mt/h of high pressure steam. The blowdown has been specified at 2.5% of the produced high pressure steam. The amount of steam used in the boilers fuel atomizer is 30% of the dispersed fuel, but other than that there are no mass and energy balances for the atomizer. The high pressure desuperheater (HD) cannot accept more than 60 Mt/h of high pressure steam, the medium pressure desuperheater (MD) and low pressure desuperheater (LD) no more than 50 Mt/h of medium and low pressure steam, respectively. Moreover, there is no energy balance equation for the lower pressure desuperheater (LD).

The maximum amount of water that can be produced from the **deaerator** (DEAERATOR) is 423.3 Mt/h. The refinery **must buy** 1.48 MW of electricity from an outside supplier in order to satisfy the electrical load required by the gas turbogenerators. The de-ionized water is stored in tanks and is consumed in the deaerator and the production units. Very low pressure oily condensate to the deaerator is produced only in the new production units.

The heating power of the fuel burned in the boilers is 8540 kcal/kg. During its preheating the fuel can absorb 22.5 kcal/kg of heat. The boiler water that is preheated in the device BWP flows through a pipe and absorbs 28 kcal/kg of heat in the preheater BWP. It thus does not enter into the mass balance of the box BWP and figures with 28 kcal/kg in the respective energy balance. Table B.4 gives the enthalpies of the various streams of steam, water and condensates (in kcal/kg) for the respective energy balance equations. In general, everything that gets into or comes out from the steam pipes (H, M, L, V) is *hot steam* at the respective pressure levels and labeled as such in Table B.4. However, the flash drums (MFD, LFD, VFD, GTGBD, UBBD) output *saturated steam* at pressure levels as indicated in Figure B.1 and *saturated liquid* at the corresponding

pressure levels. From the production units comes water (UW) having an enthalpy of 190 kcal/kg; the rest is *condensates* at the pressure levels as indicated in Figure B.1 and Table B.1. The blowdown of BOILERS into BBD and of GTG into GTGBD is a high pressure saturated liquid. The preheater BWP outputs medium pressure condensate into the DEAERATOR. The drain-off from GTGBD and UBBD has an enthalpy of 150 kcal/kg and the ones from BBD and BFP an enthalpy of 100 kcal/kg and of 181 kcal/kg, respectively. The DEAERATOR outputs water having an enthalpy of 102 kcal/kg. The remaining special enthalpy numbers are listed in Table B.4. The refinery buys electricity for \$44.4 per million Watthours (MWh) and sells electricity for \$35.5 per million Watthours. To account for the consumption of fuel oil 1 the refinery uses \$120 per metric ton and for fuel oil 2 the cost is \$100 per metric ton.

## B.4 Formulation of the Problem

With the information given above and in Tables B.1 through B.4 you can now write down a (mixed-integer) linear program to solve the problem. To achieve a common notation let us first identify and label appropriately all of the decision variables that you need in the problem. Part of the decision variables are indicated in Figure B.1 next to their respective arcs. As is customary we interpret every arc with an arrow head and e.g. an H at the top as an input to the high pressure steam pipe while the other arcs with an H at the top are outputs from that pipe. Likewise, arcs with an arrow head at a box are inputs into the respective box and impact both the energy and the mass balance of that box. Arcs that “pass” through a box participate in the energy balance, but not in the mass balance. The excess steam of very low pressure that is released into the air is denoted by  $x_{54}$ . The zero-one variables  $t_j \in \{0, 1\}$  for  $1 \leq j \leq 28$  and  $u_j \in \{0, 1\}$  for  $1 \leq j \leq 27$  indicate which turbines and which production units are “off” and “on”, respectively. Whenever a production unit works at less than 100% the turbines that reside in it *must* be shut off. The rest of the variables are defined as follows.

- $w_{01}$ : electricity purchased from an outside supplier in excess of the 1.48 MW.
- $w_{02}$ : electricity produced by the gas turbogenerators.
- $w_{03}$ : electricity produced by the steam turbogenerators.
- $w_{04}$ : electricity produced by the turbines.
- $w_{05}$ : electricity consumed by the production units.
- $w_{06}$ : electricity produced for sale outside the refinery.
- $f_{01}$ : fuel oil 1 consumed in the main boilers.
- $f_{02}$ : fuel oil 1 consumed in the GO1 furnaces.
- $f_{03}$ : fuel oil 2 consumed in the GO2 furnaces.
- $f_{04}$ : fuel gas consumed in the gas turbogenerators.
- $f_{05}$ : fuel gas consumed in the G furnaces.
- $f_{06}$ : fuel gas consumed in the GO1 furnaces.
- $f_{07}$ : fuel gas consumed in the GO2 furnaces.

Now to formulate the problem as a linear program follow the next seven steps.

1. Formulate the objective function of the problem.
2. Write the applicable enthalpy number next to each arc of Figure B.1.
3. Write down equations for the mass balance, the energy balance and the technological constraints for each box of Figure B.1 in the order of their numbering in Figure B.1.

4. Write down mass balance equations for the four different types of steam.
5. Write down mass balance equations for water (BW, DW, UW) and the condensates.
6. Write down the constraints for fuel management and electricity management.
7. Write down the inequalities that relate turbines and production units to express the fact that turbines can be operated only if the respective production unit where a turbine resides is itself operating.

The linear program you get should have 95 variables plus 27 variables  $u_j$  for the parameter settings and 100 constraints. Next add 27 equations to reflect the “state” of the refinery and for a start assume that all 27 units operate at 100% capacity.

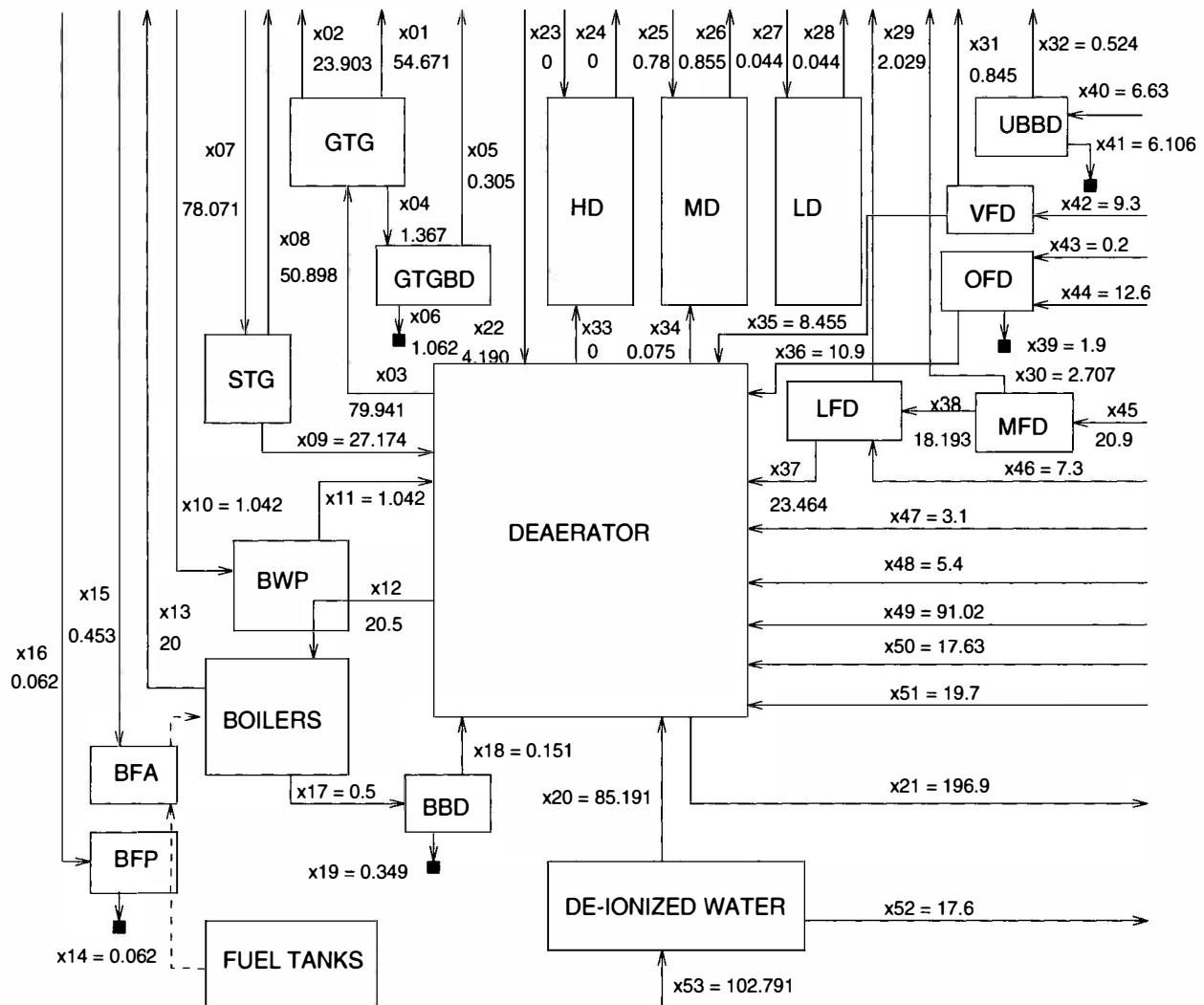
- (A)** Solve two linear programs assuming first that the refinery can sell electricity at \$35.5 and secondly, under the assumption that no electricity can be sold, i.e. with the additional constraint that  $w_{06} = 0$ .
- (B)** Suppose that the sales price for electricity that the refinery can get is dropping while the price for electricity from the outside supplier remains unchanged at \$44.4. At what price does the refinery stop selling electricity? Assuming that the refinery does not buy additional electricity, i.e. with the additional constraint  $w_{01} = 0$ , what is the maximum amount of electricity the refinery can sell?
- (C)** If your linear programming solver has a branch-and-bound facility, use it to find out which ones of the turbines are “on” and which are “off” in the respective runs. If it does not, use the following rounding procedure: If a value  $t_j \leq 0.5$  in the linear programming optimum then set the corresponding  $t_j = 0$  and  $t_j = 1$  otherwise. Rerun the problem to find a (sub-optimal) answer to the problem. Discuss how the solutions to the linear programs and the mixed-integer linear programs, respectively, differ.
- (D)** Invent your own “disaster” situation where some of the production units fail to operate fully and solve for the respective steam balance.

To check your formulation we give in Table B.5 the solution to the mixed-integer program with the additional constraint  $w_{06} = 0$  and rounded to three decimals after the point.

## B.5 Solution to the Refinery Case

To formulate the problem as a linear programming problem, we use the flow variables shown in Figure B.1 (with their respective numerical values for the optimal solution for Model 4; see below), twenty-eight variables  $t_j$  that model the operation of the turbines, the thirteen variables for fuel and electricity management introduced above and twenty-seven variables  $u_j$  that describe the “state” of the production units of the refinery. These “variables” are really parameters, i.e., inputs for the model, and assume values between zero and one according to whether or not production unit  $j$  works at 100%, i.e.,  $u_j = 1$ , or, e.g., at 50%, i.e.,  $u_j = 0.5$ , when the model is run. The *objective function* for the linear program reads

$$\text{minimize } 65.712 + 44.4w_{01} - 35.5w_{06} + 120f_{01} + 120f_{02} + 100f_{03} .$$



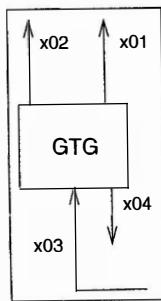
**Fig. B.5.** Flows (in metric tons per hour) in the main steam production area for Model 1

**Table B.5.** Nonzeroes of the solution to the problem with  $w_{06} = 0$ 

Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value
$x_{01}$	50.305	$x_{21}$	196.900	$x_{44}$	12.600	$u_{01}$	1.000	$u_{21}$	1.000
$x_{02}$	21.391	$x_{22}$	3.510	$x_{45}$	20.900	$u_{02}$	1.000	$u_{22}$	1.000
$x_{03}$	72.954	$x_{25}$	0.180	$x_{46}$	7.300	$u_{03}$	1.000	$u_{23}$	1.000
$x_{04}$	1.258	$x_{26}$	0.197	$x_{47}$	3.100	$u_{04}$	1.000	$u_{24}$	1.000
$x_{05}$	0.281	$x_{27}$	0.064	$x_{48}$	5.400	$u_{05}$	1.000	$u_{25}$	1.000
$x_{06}$	0.977	$x_{28}$	0.064	$x_{49}$	91.020	$u_{06}$	1.000	$u_{26}$	1.000
$x_{07}$	73.705	$x_{29}$	2.029	$x_{50}$	17.630	$u_{07}$	1.000	$u_{27}$	1.000
$x_{08}$	52.812	$x_{30}$	2.707	$x_{51}$	19.700	$u_{08}$	1.000	$t_{12}$	1.000
$x_{09}$	20.893	$x_{31}$	0.845	$x_{52}$	17.600	$u_{09}$	1.000	$t_{13}$	1.000
$x_{10}$	1.042	$x_{32}$	0.524	$x_{53}$	102.706	$u_{10}$	1.000	$t_{14}$	1.000
$x_{11}$	1.042	$x_{34}$	0.017	$f_{01}$	1.511	$u_{11}$	1.000	$t_{15}$	1.000
$x_{12}$	20.500	$x_{35}$	8.455	$f_{02}$	5.660	$u_{12}$	1.000	$t_{19}$	1.000
$x_{13}$	20.000	$x_{36}$	10.900	$f_{03}$	21.830	$u_{13}$	1.000	$t_{21}$	1.000
$x_{14}$	0.062	$x_{37}$	23.464	$f_{04}$	9.242	$u_{14}$	1.000	$t_{22}$	1.000
$x_{15}$	0.453	$x_{38}$	18.193	$f_{05}$	9.310	$u_{15}$	1.000	$t_{23}$	1.000
$x_{16}$	0.062	$x_{39}$	1.900	$f_{06}$	9.910	$u_{16}$	1.000	$t_{25}$	1.000
$x_{17}$	0.500	$x_{40}$	6.630	$w_{02}$	24.482	$u_{17}$	1.000		
$x_{18}$	0.151	$x_{41}$	6.106	$w_{03}$	10.502	$u_{18}$	1.000		
$x_{19}$	0.349	$x_{42}$	9.300	$w_{04}$	1.016	$u_{19}$	1.000		
$x_{20}$	85.106	$x_{43}$	0.200	$w_{05}$	36.000	$u_{20}$	1.000		
Objective function value: 3109.291									

The constant \$65.712 comes from the load requirement for the gas turbogenerator that for technological reasons must be bought from the electricity company. We now give a complete listing of all equations and inequalities except nonnegativity constraints. We start by writing for each box of Figure B.1 (in the order of their respective numbering) a mass balance (MB), an energy balance (EB) and/or the relevant technological constraints (TC) except for the capacity constraints.

### (1) Gas turbogenerators (GTG):



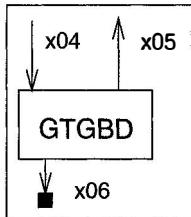
Variable	Variable Description	Enthalpy
$x_{01}$	High pressure steam	722
$x_{02}$	Low pressure steam	677
$x_{03}$	Water from DEAERATOR	102
$x_{04}$	Blowdown from GTG	263
$f_{04}$	Fuel gas burned in GTG	
$w_{01}$	Electricity purchased from outside	
$w_{02}$	Electricity produced by GTG	

$$\text{MB: } -x_{01} - x_{02} + x_{03} - x_{04} = 0,$$

$$\text{TC: } x_{01} - 0.617w_{02} = 35.2,$$

$$\text{TC: } x_{02} - 0.355w_{02} = 12.7,$$

$$\text{TC: } f_{04} - 0.206w_{02} = 4.199$$

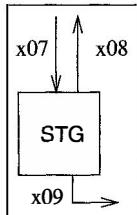
**(2) Gas turbogenerators blowdown flash drum (GTGBD):**

Variable	Variable Description	Enthalpy
$x_{04}$	Blowdown from GTG	263
$x_{05}$	Low pressure steam	656
$x_{06}$	Drain-off from GTGBD	150

$$\text{TC: } -0.025x_01 + x_{04} = 0,$$

$$\text{MB: } x_{04} - x_{05} - x_{06} = 0,$$

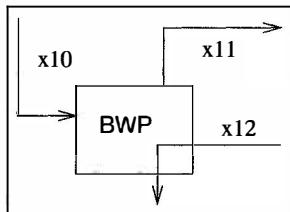
$$\text{EB: } 263x_{04} - 656x_{05} - 150x_{06} = 0$$

**(3) Steam turbogenerators (STG):**

Variable	Variable Description	Enthalpy
$x_{07}$	High pressure steam	772
$x_{08}$	Low pressure steam	677
$x_{09}$	Water to DEAERATOR	40
$w_{03}$	Electricity produced by STG	

$$\text{MB: } x_{07} - x_{09} - x_{08} = 0,$$

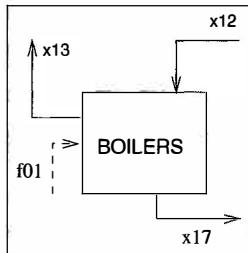
$$\text{EB: } 0.122x_{08} + 0.249x_{09} - w_{03} = 1.143$$

**(4) Boiler water preheater (BWP):**

Variable	Variable Description	Enthalpy
$x_{10}$	Medium pressure steam	732
$x_{11}$	Medium pressure condensate	181
$x_{12}$	Water preheated for BOILERS	28

$$\text{MB: } x_{10} - x_{11} = 0,$$

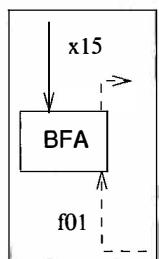
$$\text{EB: } 732x_{10} - 181x_{11} - 28x_{12} = 0$$

**(5) Boilers (BOILERS):**

Variable	Variable Description	Enthalpy
$x_{12}$	Preheated water from BWP	130
$x_{13}$	High pressure steam	772
$x_{17}$	Blowdown to BBD	263
$f_{01}$	Fuel oil 1 consumed in the main boilers	8540

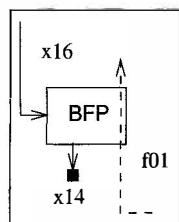
$$\text{MB: } x_{12} - x_{13} - x_{17} = 0,$$

$$\text{EB: } -130x_{12} + 772x_{13} + 263x_{17} - 8540f_{01} = 0$$

**(6) Boiler fuel atomizer (BFA):**

Variable	Variable Description	Enthalpy
$x_{15}$	Medium pressure steam	
$f_{01}$	Fuel atomized and sent to BOILERS	

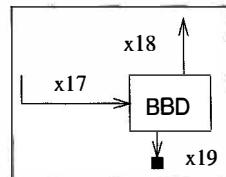
**TC:**  $x_{15} - 0.3f_{01} = 0$

**(7) Boiler fuel preheater (BFP):**

Variable	Variable Description	Enthalpy
$x_{14}$	Medium pressure steam	732
$x_{16}$	Drain-off from BFP	181
$f_{01}$	Fuel oil 1 (preheating)	22.5

**MB:**  $-x_{14} + x_{16} = 0$ ,

**EB:**  $22.5f_{01} + 181x_{14} - 732x_{16} = 0$

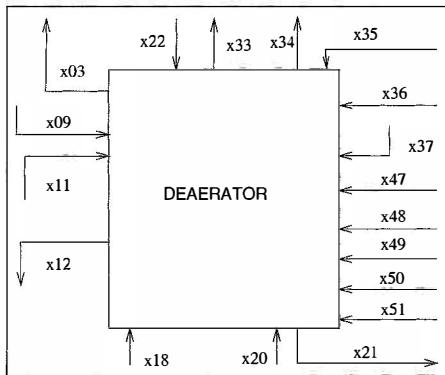
**(8) BBD flash drum :**

Variable	Variable Description	Enthalpy
$x_{17}$	Boiler blowdown	263
$x_{18}$	Water from BBD to DEAERATOR	640
$x_{19}$	Drain-off from BBD	100

**TC:**  $-0.025x_{13} + x_{17} = 0$ ,

**MB:**  $x_{17} - x_{18} - x_{19} = 0$ ,

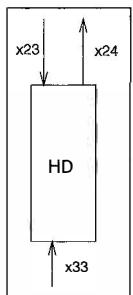
**EB:**  $263x_{17} - 640x_{18} - 100x_{19} = 0$

**(9) Daeaerator (DEAERATOR)**

Variable	Variable Description	Enthalpy
$x_{03}$	Water to GTG	102
$x_{09}$	Water from STG	40
$x_{11}$	Medium pressure condensate from BWP	181
$x_{12}$	Water to BOILERS	102
$x_{18}$	Water from BBD	640
$x_{20}$	De-ionized water	10
$x_{21}$	Boiler water to production units	10
$x_{22}$	Very low pressure steam	686
$x_{33}$	Water to HD	102
$x_{34}$	Water to MD	102
$x_{35}$	Very low pressure saturated liquid	134
$x_{36}$	Water from OFD	100
$x_{37}$	Low pressure saturated liquid	150
$x_{47}$	Very low pressure oily condensate	126
$x_{48}$	Medium pressure oily condensate	181
$x_{49}$	Low pressure condensate	144
$x_{50}$	Low pressure oily condensate	144
$x_{51}$	Very low pressure condensate	126

$$\text{MB: } -x_{03} + x_{09} + x_{11} - x_{12} + x_{18} + x_{20} - x_{21} + x_{22} - x_{33} - x_{34} + x_{35} + x_{36} + x_{37} + x_{47} + x_{48} + x_{49} + x_{50} + x_{51} = 0$$

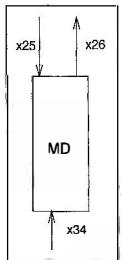
$$\text{EB: } -102x_{03} + 40x_{09} + 181x_{11} - 102x_{12} + 640x_{18} + 10x_{20} - 102x_{21} + 686x_{22} - 102x_{33} - 102x_{34} + 134x_{35} + 100x_{36} + 150x_{37} + 126x_{47} + 181x_{48} + 144x_{49} + 144x_{50} + 126x_{51} = 0$$

**(10) HD desuperheater :**

Variable	Variable Description	Enthalpy
$x_{23}$	High pressure steam	772
$x_{24}$	Medium pressure steam	732
$x_{33}$	Water from DEAERATOR	102

$$\text{MB: } x_{23} - x_{24} + x_{33} = 0,$$

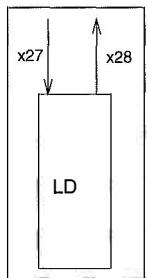
$$\text{EB: } 772x_{23} - 732x_{24} + 102x_{33} = 0$$

**(11) MD desuperheater :**

Variable	Variable Description	Enthalpy
$x_{25}$	Medium pressure steam	732
$x_{26}$	Low pressure steam	677
$x_{34}$	Water from DEAERATOR	102

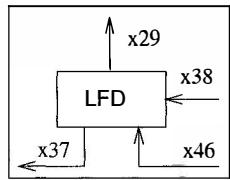
**MB:**  $x_{25} - x_{26} + x_{34} = 0,$

**EB:**  $732x_{25} - 677x_{26} + 102x_{34} = 0$

**(12) LD desuperheater :**

Variable	Variable Description	Enthalpy
$x_{27}$	Low pressure steam	677
$x_{28}$	Very low pressure steam	686

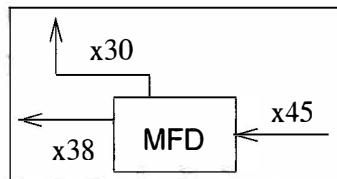
**MB:**  $x_{27} - x_{28} = 0$

**(13) LFD flash drum :**

Variable	Variable Description	Enthalpy
$x_{29}$	Low pressure saturated steam	656
$x_{37}$	Low pressure saturated liquid	150
$x_{38}$	Medium pressure saturated liquid	194
$x_{46}$	Medium pressure condensate	181

**MB:**  $-x_{29} - x_{37} + x_{38} + x_{46} = 0,$

**EB:**  $-656x_{29} - 150x_{37} + 194x_{38} + 181x_{46} = 0$

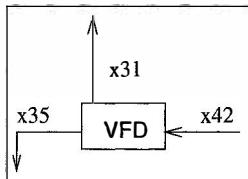
**(14) MFD flash drum :**

Variable	Variable Description	Enthalpy
$x_{30}$	Medium pressure saturated steam	665
$x_{38}$	Medium pressure saturated liquid	194
$x_{45}$	High pressure condensate	255

**MB:**  $-x_{30} - x_{38} + x_{45} = 0,$

**EB:**  $-665x_{30} - 194x_{38} + 255x_{45} = 0$

## (15) VFD flash drum :

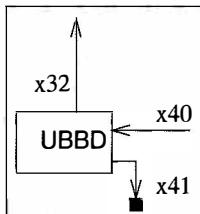


Variable	Variable Description	Enthalpy
$x_{31}$	Very low pressure saturated steam	651
$x_{35}$	Very low pressure saturated liquid	134
$x_{42}$	Medium pressure condensate	181

$$\text{MB: } -x_{31} - x_{35} + x_{42} = 0,$$

$$\text{EB: } -651x_{31} - 134x_{35} + 181x_{42} = 0$$

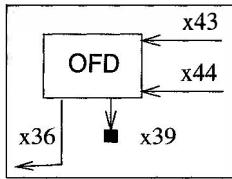
## (16) UBBD flash drum :



$$\text{MB: } -x_{32} + x_{40} - x_{41} = 0,$$

$$\text{EB: } -656x_{32} + 190x_{40} - 150x_{41} = 0$$

## (17) OFD flash drum :

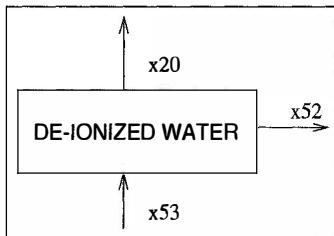


Variable	Variable Description	Enthalpy
$x_{36}$	Water to DEAERATOR	100
$x_{39}$	Drain off	150
$x_{43}$	Very low pressure oily condensate	126
$x_{44}$	Medium pressure oily condensate	181

$$\text{MB: } -x_{36} - x_{39} + x_{43} + x_{44} = 0,$$

$$\text{EB: } -100x_{36} - 640x_{39} + 126x_{43} + 181x_{44} = 0$$

## (18) De-ionized water tanks (DE-IONIZED WATER):



Variable	Variable Description	Enthalpy
$x_{20}$	Water to DEAERATOR	
$x_{52}$	De-ionized water	
$x_{53}$	Fresh water	

$$\text{MB: } -x_{20} - x_{52} + x_{53} = 0$$

Next we write down all the **capacity constraints** for the various devices.

$$x_{07} \leq 100, 3 \leq x_{09} \leq 60, 20 \leq x_{13} \leq 60, x_{23} \leq 60, x_{25} \leq 50, x_{27} \leq 50$$

$$x_{03} + x_{12} + x_{21} + x_{33} + x_{34} \leq 423.3, f_{04} \leq 10.7$$

We write a mass balance equation for the **high pressure steam pipe**,

$$x_{01} - x_{07} + x_{13} - x_{23} - 9.5t_{01} - 3.3t_{02} - 9t_{03} - 4.2t_{04} - 37t_{05} - 4.6t_{06} - 6.6t_{07} - 4.6t_{08} - 0.5t_{09} - 8t_{10} - 6.1t_{11} - 4.4u_{10} - 3.1u_{11} - 11.5u_{12} - 4.4u_{13} - 1.2u_{17} - 0.2u_{19} + 1.3u_{20} + 30u_{21} - 4u_{24} + 8u_{26} - 7.1u_{27} = 0,$$

for the **medium pressure steam pipe**,

$$-x_{10} - x_{15} - x_{16} + x_{24} - x_{25} + x_{30} + 9.5t_{01} + 3.3t_{02} + 9t_{03} + 4.2t_{04} + 37t_{05} - 7.3t_{12} - 7.3t_{13} - 4.8t_{14} - 3.3t_{15} - 2.7t_{16} - 3.3t_{17} - 2.4t_{18} - 1.3t_{19} - 2.1t_{20} - 2t_{21} - 4.7t_{22} - 2.2t_{23} - 3.8t_{24} - 0.9t_{25} - 0.9t_{26} - 1t_{27} - 2t_{28} - 3.5u_{01} - 5.8u_{03} - 16.8u_{04} - 1.8u_{06} - 5.1u_{08} + 12.4u_{11} - 2.6u_{12} + 26.9u_{13} - 0.3u_{14} - 2.8u_{15} + 18.6u_{17} + 19.8u_{19} + 19.7u_{20} - 10.1u_{21} - 5u_{25} + 0.13u_{26} - 10.9u_{27} = 0,$$

for the **low pressure steam pipe**,

$$x_{02} + x_{05} + x_{08} + x_{26} - x_{27} + x_{29} + x_{32} + 4.6t_{06} + 6.6t_{07} + 4.6t_{08} + 0.5t_{09} + 7.3t_{12} + 7.3t_{13} + 4.8t_{14} + 3.3t_{15} + 2.7t_{16} + 3.3t_{17} + 2.4t_{18} + 1.3t_{19} + 2.1t_{20} - 6.1u_{07} + 5.5u_{11} - 2.52u_{13} - 0.1u_{14} - 52u_{15} + 2.9u_{17} - 20.1u_{18} - 0.2u_{19} - 1.2u_{20} - 2.82u_{21} - 5.4u_{23} - 2.8u_{24} - 16.33u_{27} = 0,$$

and for the **very low pressure steam pipe**

$$-x_{22} + x_{28} + x_{31} - x_{54} + 8t_{10} + 6.1t_{11} + 2t_{21} + 4.7t_{22} + 2.2t_{23} + 3.8t_{24} + 0.9t_{25} + 0.9t_{26} + 1t_{27} + 2t_{28} - 1.5u_{03} - 0.2u_{08} - 1.8u_{09} - 1.2u_{10} - 2.5u_{27} = 0.$$

The **water (BW, DW, UW)** and the **condensates** from the various production units give rise to 13 constraints.

$$\begin{aligned} x_{21} - 4.6u_{01} - 7.5u_{10} - 39u_{11} - 26u_{13} - 4.3u_{15} - 3.80u_{16} - 24u_{17} - 24.5u_{19} - 28.5u_{20} - 33.5u_{21} - 0.5u_{23} - 0.7u_{24} &= 0 \\ x_{40} - 1.40u_{11} - 1.00u_{13} - 1.00u_{19} - 1.28u_{20} - 1.30u_{21} - 0.65u_{26} &= 0 \\ x_{42} - 9.3u_{04} = 0, \quad x_{43} - 0.20u_{08} = 0, \quad x_{44} - 12.6u_{08} = 0, \quad x_{45} - 11.5u_{12} - 1.4u_{17} - 4.7u_{24} - 3.3u_{27} &= 0 \\ x_{46} - 0.5u_{11} - 1.3u_{13} - 3.1u_{15} - 0.3u_{19} - 2.1u_{27} = 0, \quad x_{47} - 3.1u_{27} = 0, \quad x_{48} - 4.8u_{25} - 0.6u_{27} &= 0 \\ x_{49} - 0.50u_{11} - 2.4u_{13} - 0.1u_{14} - 55.7u_{15} - 1.4u_{17} - 20.1u_{18} - 0.2u_{19} - 1.1u_{20} - 1.82u_{21} - 7.7u_{27} &= 0 \\ x_{50} - 0.3u_{15} - 5.9u_{23} - 2.80u_{24} - 8.63u_{27} = 0, \quad x_{51} - 6.1u_{07} - 10.2u_{10} - 3.4u_{27} &= 0 \\ x_{52} - 0.5u_{15} - 0.4u_{19} - 1.1u_{24} - 15.6u_{26} &= 0 \end{aligned}$$

The **fuel** management – other than the fuel that is burned in the main boiler – is captured by the following five equations/inequalities.

$$\begin{aligned} f_{05} - 1.08u_{04} - 3.77u_{05} - 0.96u_{06} - 0.01u_{08} - 0.25u_{10} - 0.35u_{13} - 0.73u_{17} - 1.14u_{26} - 1.02u_{27} &= 0 \\ f_{04} + f_{05} + f_{06} + f_{07} - 5u_{07} - 3.925u_{08} - 17.103u_{15} - 0.856u_{24} - 1.578u_{26} &= 0 \\ f_{02} + f_{06} - 3.6u_{01} - 6.4u_{03} - 1.36u_{06} - 1.73u_{12} - 0.89u_{14} - 1.59u_{25} &= 0 \\ f_{03} + f_{07} - 5.65u_{11} - 9.22u_{13} - 2.59u_{19} - 2.03u_{20} - 2.34u_{21} = 0, \quad -f_{02} + 0.89u_{14} &\leq 0 \end{aligned}$$

**Electricity** production, consumption and management yield the following three equations.

$$\begin{aligned} w_{04} - 0.372t_{01} - 0.116t_{02} - 0.280t_{03} - 0.111t_{04} - 0.045t_{05} - 0.278t_{06} - 0.306t_{07} - 0.204t_{08} - 0.013t_{09} - 0.460t_{10} - 0.145t_{11} - 0.164t_{12} - 0.144t_{13} - 0.092t_{14} - 0.052t_{15} - 0.036t_{16} - 0.044t_{17} - 0.030t_{18} - 0.015t_{19} - 0.024t_{20} - 0.184t_{21} - 0.250t_{22} - 0.100t_{23} - 0.150t_{24} - 0.015t_{25} - 0.013t_{26} - 0.011t_{27} - 0.015t_{28} &= 0 \\ w_{05} - 0.800u_{01} - 0.115u_{02} - 1.245u_{03} - 0.820u_{04} - 0.395u_{06} - 0.050u_{07} - 0.220u_{08} - 0.105u_{10} - 1.620u_{11} - 1.270u_{12} - 4.350u_{13} - 1.130u_{14} - 0.925u_{15} - 0.093u_{16} - 0.685u_{17} - 0.205u_{18} - 1.140u_{19} - 2.975u_{20} - 6.330u_{21} - 0.002u_{22} - 0.140u_{23} - 0.720u_{24} - 0.565u_{25} - 0.545u_{26} - 9.555u_{27} &= 0 \\ w_{01} + w_{02} + w_{03} + w_{04} - w_{05} - w_{06} &= 0 \end{aligned}$$

The next 28 inequalities state that **turbines** can be operated only if the production unit to which they belong is itself operating at full capacity.

$$\begin{aligned} t_{01} - u_{19} &\leq 0, \quad t_{02} - u_{11} \leq 0, \quad t_{03} - u_{21} \leq 0, \quad t_{04} - u_{26} \leq 0, \quad t_{05} - u_{16} \leq 0, \quad t_{06} - u_{19} \leq 0, \quad t_{07} - u_{11} \leq 0 \\ t_{08} - u_{11} &\leq 0, \quad t_{09} - u_{20} \leq 0, \quad t_{10} - u_{22} \leq 0, \quad t_{11} - u_{22} \leq 0, \quad t_{12} - u_{15} \leq 0, \quad t_{13} - u_{13} \leq 0, \quad t_{14} - u_{13} \leq 0 \end{aligned}$$

$$\begin{aligned} t_{15} - u_{13} &\leq 0, t_{16} - u_{20} \leq 0, t_{18} - u_{20} \leq 0, t_{17} - u_{21} \leq 0, t_{19} - u_{21} \leq 0, t_{20} - u_{13} \leq 0, t_{21} - u_{12} \leq 0 \\ t_{22} - u_{16} &\leq 0, t_{23} - u_{16} \leq 0, t_{24} - u_{16} \leq 0, t_{25} - u_{16} \leq 0, t_{26} - u_{12} \leq 0, t_{27} - u_{22} \leq 0, t_{28} - u_{16} \leq 0 \end{aligned}$$

Not shown are an additional 27 equations that are supplied on as needed basis to reflect which ones of the production units are operating and which ones are not. Thus in the normal case when all production units are functioning at full capacity these additional equations read  $u_j = 1$  for  $1 \leq j \leq 27$ .

Thus, the linear program has 95 variables of which 28 must be zero-one valued plus 27 parameters corresponding to the variables  $u_j$  that describe the state of the refinery when some of its production units are down for maintenance and/or operating at e.g. 50% of their respective capacity. Depending on the state of the refinery the values of the 27 parameters are set to their respective values prior to solving the linear program that results. In the above formulation we have assumed that a production unit operating at less than 100% of capacity has its turbines shut off which is a realistic assumption to make.

We introduce a dummy variable  $x_0$  to include the constant \$65.712 in the objective function. The reason for this is that most LP solver like CPLEX do not permit the inclusion of a constant in the objective function. The basic linear programming model in CPLEX 1p format is as follows

```

Minimize
obj: 65.712 x0 + 44.4 w01 - 35.5 w06 + 120 f01 + 120 f02 + 100 f03
Subject To
c1: x0 = 1
c2: - x01 - x02 + x03 - x04 = 0
c3: x01 - 0.617 w02 = 35.2
c4: x02 - 0.355 w02 = 12.7
c5: - 0.206 w02 + f04 = 4.199
c6: - 0.025 x01 + x04 = 0
c7: x04 - x05 - x06 = 0
c8: 263 x04 - 656 x05 - 150 x06 = 0
c9: x07 - x09 - x08 = 0
c10: 0.249 x09 + 0.122 x08 - w03 = 1.143
c11: x10 - x11 = 0
c12: 732 x10 - 181 x11 - 28 x12 = 0
c13: x12 - x13 - x17 = 0
c14: - 8540 f01 - 130 x12 + 772 x13 + 263 x17 = 0
c15: - 0.025 x13 + x17 = 0
c16: - 0.3 f01 + x15 = 0
c17: - x14 + x16 = 0
c18: 22.5 f01 + 181 x14 - 732 x16 = 0
c19: x17 - x18 - x19 = 0
c20: 263 x17 - 640 x18 - 100 x19 = 0
c21: - x03 + x09 + x11 - x12 + x18 + x20 - x21 + x22 - x33 - x34 + x35 + x36
+ x37 + x47 + x48 + x49 + x50 + x51 = 0
c22: - 102 x03 + 40 x09 + 181 x11 - 102 x12 + 640 x18 + 10 x20 - 102 x21
+ 686 x22 - 102 x33 - 102 x34 + 134 x35 + 100 x36 + 150 x37 + 126 x47
+ 181 x48 + 144 x49 + 144 x50 + 126 x51 = 0
c23: x33 + x23 - x24 = 0

```

```

c24: 102 x33 + 772 x23 - 732 x24 = 0
c25: x34 + x25 - x26 = 0
c26: 102 x34 + 732 x25 - 677 x26 = 0
c27: x27 - x28 = 0
c28: - x37 - x29 + x38 + x46 = 0
c29: - 150 x37 - 656 x29 + 194 x38 + 181 x46 = 0
c30: - x38 - x30 + x45 = 0
c31: - 194 x38 - 665 x30 + 255 x45 = 0
c32: - x35 - x31 + x42 = 0
c33: - 134 x35 - 651 x31 + 181 x42 = 0
c34: - x32 + x40 - x41 = 0
c35: - 656 x32 + 190 x40 - 150 x41 = 0
c36: - x36 - x39 + x43 + x44 = 0
c37: - 100 x36 - 640 x39 + 126 x43 + 181 x44 = 0
c38: - x20 - x52 + x53 = 0
c39: x07 <= 100
c40: x09 >= 3
c41: x09 <= 60
c42: x13 >= 20
c43: x13 <= 60
c44: x23 <= 60
c45: x25 <= 50
c46: x27 <= 50
c47: x03 + x12 + x21 + x33 + x34 <= 423.3
c48: f04 <= 10.7
c49: x01 - x07 + x13 - x23 - 9.5 t01 - 3.3 t02 - 9 t03 - 4.2 t04 - 37 t05
     - 4.6 t06 - 6.6 t07 - 4.6 t08 - 0.5 t09 - 8 t10 - 6.1 t11 - 4.4 u10
     - 3.1 u11 - 11.5 u12 - 4.4 u13 - 1.2 u17 - 0.2 u19 + 1.3 u20 + 30 u21
     - 4 u24 + 8 u26 - 7.1 u27 = 0
c50: - x10 - x15 - x16 + x24 - x25 + x30 + 9.5 t01 + 3.3 t02 + 9 t03
     + 4.2 t04 + 37 t05 + 12.4 u11 - 2.6 u12 + 26.9 u13 + 18.6 u17
     + 19.8 u19 + 19.7 u20 - 10.1 u21 + 0.13 u26 - 10.9 u27 - 7.3 t12
     - 7.3 t13 - 4.8 t14 - 3.3 t15 - 2.7 t16 - 3.3 t17 - 2.4 t18 - 1.3 t19
     - 2.1 t20 - 2 t21 - 4.7 t22 - 2.2 t23 - 3.8 t24 - 0.9 t25 - 0.9 t26
     - t27 - 2 t28 - 3.5 u01 - 5.8 u03 - 16.8 u04 - 1.8 u06 - 5.1 u08
     - 0.3 u14 - 2.8 u15 - 5 u25 = 0
c51: x02 + x05 + x08 + x26 - x27 + x29 + x32 + 4.6 t06 + 6.6 t07 + 4.6 t08
     + 0.5 t09 + 5.5 u11 - 2.52 u13 + 2.9 u17 - 0.2 u19 - 1.2 u20 - 2.82 u21
     - 2.8 u24 - 16.33 u27 + 7.3 t12 + 7.3 t13 + 4.8 t14 + 3.3 t15 + 2.7 t16
     + 3.3 t17 + 2.4 t18 + 1.3 t19 + 2.1 t20 - 0.1 u14 - 52 u15 - 6.1 u07
     - 20.1 u18 - 5.4 u23 = 0
c52: - x22 + x28 + x31 + 8 t10 + 6.1 t11 - 1.2 u10 - 2.5 u27 + 2 t21
     + 4.7 t22 + 2.2 t23 + 3.8 t24 + 0.9 t25 + 0.9 t26 + t27 + 2 t28
     - 1.5 u03 - 0.2 u08 - x54 - 1.8 u09 = 0
c53: x21 - 7.5 u10 - 39 u11 - 26 u13 - 24 u17 - 24.5 u19 - 28.5 u20
     - 33.5 u21 - 0.7 u24 - 4.6 u01 - 4.3 u15 - 0.5 u23 - 3.8 u16 = 0

```

```

c54: x40 - 1.4 u11 - u13 - u19 - 1.28 u20 - 1.3 u21 - 0.65 u26 = 0
c55: x42 - 9.3 u04 = 0
c56: x43 - 0.2 u08 = 0
c57: x44 - 12.6 u08 = 0
c58: x45 - 11.5 u12 - 1.4 u17 - 4.7 u24 - 3.3 u27 = 0
c59: x46 - 0.5 u11 - 1.3 u13 - 0.3 u19 - 2.1 u27 - 3.1 u15 = 0
c60: x47 - 3.1 u27 = 0
c61: x48 - 0.6 u27 - 4.8 u25 = 0
c62: x49 - 0.5 u11 - 2.4 u13 - 1.4 u17 - 0.2 u19 - 1.1 u20 - 1.82 u21
    - 7.7 u27 - 0.1 u14 - 55.7 u15 - 20.1 u18 = 0
c63: x50 - 2.8 u24 - 8.63 u27 - 0.3 u15 - 5.9 u23 = 0
c64: x51 - 10.2 u10 - 3.4 u27 - 6.1 u07 = 0
c65: x52 - 0.4 u19 - 1.1 u24 - 15.6 u26 - 0.5 u15 = 0
c66: - 0.25 u10 - 0.35 u13 - 0.73 u17 - 1.14 u26 - 1.02 u27 - 1.08 u04
    - 0.96 u06 - 0.01 u08 + f05 - 3.77 u05 = 0
c67: f04 - 0.856 u24 - 1.578 u26 - 3.925 u08 - 17.103 u15 - 5 u07 + f05
    + f06 + f07 = 0
c68: f02 - 1.73 u12 - 3.6 u01 - 6.4 u03 - 1.36 u06 - 0.89 u14 - 1.59 u25
    + f06 = 0
c69: f03 - 5.65 u11 - 9.22 u13 - 2.59 u19 - 2.03 u20 - 2.34 u21 + f07 = 0
c70: - f02 + 0.89 u14 <= 0
c71: - 0.372 t01 - 0.116 t02 - 0.28 t03 - 0.111 t04 - 0.045 t05 - 0.278 t06
    - 0.306 t07 - 0.204 t08 - 0.013 t09 - 0.46 t10 - 0.145 t11 - 0.164 t12
    - 0.144 t13 - 0.092 t14 - 0.052 t15 - 0.036 t16 - 0.044 t17 - 0.03 t18
    - 0.015 t19 - 0.024 t20 - 0.184 t21 - 0.25 t22 - 0.1 t23 - 0.15 t24
    - 0.015 t25 - 0.013 t26 - 0.011 t27 - 0.015 t28 + w04 = 0
c72: - 0.105 u10 - 1.62 u11 - 1.27 u12 - 4.35 u13 - 0.685 u17 - 1.14 u19
    - 2.975 u20 - 6.33 u21 - 0.72 u24 - 0.545 u26 - 9.555 u27 - 0.8 u01
    - 1.245 u03 - 0.82 u04 - 0.395 u06 - 0.22 u08 - 1.13 u14 - 0.925 u15
    - 0.565 u25 - 0.05 u07 - 0.205 u18 - 0.14 u23 - 0.093 u16 + w05
    - 0.115 u02 - 0.002 u22 = 0
c73: w01 - w06 + w02 + w03 + w04 - w05 = 0
c74: t01 - u19 <= 0
c75: t02 - u11 <= 0
c76: t03 - u21 <= 0
c77: t04 - u26 <= 0
c78: t05 - u16 <= 0
c79: t06 - u19 <= 0
c80: t07 - u11 <= 0
c81: t08 - u11 <= 0
c82: t09 - u20 <= 0
c83: t10 - u22 <= 0
c84: t11 - u22 <= 0
c85: t12 - u15 <= 0
c86: - u13 + t13 <= 0
c87: - u13 + t14 <= 0

```

```

c88: - u13 + t15 <= 0
c89: - u20 + t16 <= 0
c90: - u20 + t18 <= 0
c91: - u21 + t17 <= 0
c92: - u21 + t19 <= 0
c93: - u13 + t20 <= 0
c94: - u12 + t21 <= 0
c95: t22 - u16 <= 0
c96: t23 - u16 <= 0
c97: t24 - u16 <= 0
c98: t25 - u16 <= 0
c99: - u12 + t26 <= 0
c100: t27 - u22 <= 0
c101: t28 - u16 <= 0
c102: u01 = 1
c103: u02 = 1
c104: u03 = 1
c105: u04 = 1
c106: u05 = 1
c107: u06 = 1
c108: u07 = 1
c109: u08 = 1
c110: u09 = 1
c111: u10 = 1
c112: u11 = 1
c113: u12 = 1
c114: u13 = 1
c115: u14 = 1
c116: u15 = 1
c117: u16 = 1
c118: u17 = 1
c119: u18 = 1
c120: u19 = 1
c121: u20 = 1
c122: u21 = 1
c123: u22 = 1
c124: u23 = 1
c125: u24 = 1
c126: u25 = 1
c127: u26 = 1
c128: u27 = 1
End

```

We refer to the above model as Model 1.

**(A)** Solve two linear programs assuming first that the refinery can sell electricity at \$35.5 and secondly, under the assumption that no electricity can be sold, i.e. with the additional constraint that  $w_{06} = 0$ .

The values of the variables for the optimal solution are shown in the first column of Tables B.6 and B.7. To implement the restriction that no electricity can be sold, we add the constraint  $w_{06} = 0$ , to the basic model; we refer to this new model as Model 2. The solution is shown in the second column of Tables B.6 and B.7. The restriction not to sell electricity costs to the refinery  $3108.774 - 2984.502 = 124.772$  dollars per hour. The electricity produced by the gas turbogenerators ( $w_{02}$ ) and the steam turbogenerators ( $w_{03}$ ) is reduced when sale is not permitted, while the electricity production of the turbines remains at the same level. When sale of electricity is permitted the refinery produces 8.443MW for sale.

**(B)** Suppose that the sales price for electricity that the refinery can get is dropping while the price for electricity from the outside supplier remains unchanged at \$44.4. At what price does the refinery stop selling electricity? Assuming that the refinery does not buy additional electricity, i.e. with the additional constraint  $w_{01} = 0$ , what is the maximum amount of electricity the refinery can sell?

The price at which the refinery will stop selling electricity can be seen from the sensitivity analysis information of the objective function coefficient of the corresponding variable, i.e., variable  $w_{06}$ . To obtain this information in CPLEX we issue the command

```
display objective w06
```

and get the answer that variable  $w_{06}$  will remain basic until the price goes down to \$20.7811. At this price the variable  $w_{06}$  will become zero and the optimal value will be the same with that of Model 2.

To find the maximum amount of electricity the refinery can produce for sale we change the objective function of Model 1 to

$$\max w_{06}$$

and set  $w_{01} = 0$ . We refer to this LP as Model 3 and present the solution in the third column of Tables B.6 and B.7. The maximum amount of electricity that can be sold is 16.296MW. This however is not the amount that is sold under the optimal production schedule when the total cost is minimized. The electricity produced by the turbines is two times as much as in Model 1. The electricity produced by the steam turbogenerators is increased while the electricity produced by the gas turbogenerators is at the same level. In terms of the original objective function we get \$3237.484. This means that selling electricity is not as profitable as selling the fuel oils.

**(C)** If your linear programming solver has a branch-and-bound facility, use it to find out which ones of the turbines are “on” and which are “off” in the respective runs. If it does not, use the following rounding procedure: If a value  $t_j \leq 0.5$  in the linear programming optimum then set the corresponding  $t_j = 0$  and  $t_j = 1$  otherwise. Rerun the problem to find a (sub-optimal) answer to the problem. Discuss how the solutions to the linear programs and the mixed-integer linear programs, respectively, differ.

In the LP solutions several turbines are “run” at less than full capacity which is infeasible from an engineering point of view. Thus the  $t_{ij}$  variables must be integer. To solve the **mixed-integer** problems we append to the LP the integrality restriction for the turbine variables. In CPLEX this is done by appending

Integers

$t_{01}$	$t_{02}$	$t_{03}$	$t_{04}$	$t_{05}$	$t_{06}$	$t_{07}$	$t_{08}$	$t_{09}$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$
$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{21}$	$t_{22}$	$t_{23}$	$t_{24}$	$t_{25}$	$t_{26}$	$t_{27}$	$t_{28}$		

before the End statement. We refer to the mixed-integer programs for Models 1,2,3 as Models 4,5,6, respectively. The solutions are shown in Tables B.6 and B.7 and the optimal flows are indicated in Figure B.1. The difference between the solutions of the linear and the mixed-integer programs is not significant. The solution of Model 4 has  $t_{16} = 0$  (it was 0.275 in Model 1),  $t_{23} = 0$  (it was 1), and  $t_{24} = 1$  (it was 0.431), while the other integer variables are the same as in Model 1. The optimal value is \$2985.069, i.e., \$0.567 more than the optimal value of Model 1. In the solution of Model 5 we have  $t_{17} = 1$  (it was 0.43 in Model 2),  $t_{20} = 1$  (it was 0),  $t_{24} = 0$  (it was 0.253), and the other integer variables are the same as in Model 2. The optimal value is \$3109.560, i.e., \$0.786 more than the optimal value of Model 2. Finally, in the solution of Model 6 we have  $t_{02} = 1$  (it was 0.694 in Model 3),  $t_{09} = 1$  (it was 0),  $t_{10} = 0$  (it was 0.229),  $t_{18} = 1$  (it was 0.462), and the other integer variables are the same as in Model 3. The optimal value is 16.26 MW, i.e., 0.036 MW less than the optimal value of Model 3.

**(D)** Invent your own “disaster” situation where some of the production units fail to operate fully and solve for the respective steam balance.

As a disaster situation we consider the situation under which unit 15 operates at 50% of its capacity. We refer to the model that results from changing constraint  $c_{116}$  to  $u_{15} = 0.5$  in Model 4, as Model 7. The solution is shown in Table B.6 and B.7. The operating cost is \$3899.872, i.e., \$914.803 more than in Model 4. This increase can be explained by the fact that unit 15 is the major producer of fuel gas. When it operates at 50%,  $8.5512(17.103/2)$  MT/h of fuel gas are lost. This fuel gas can be replaced by fuel oils, but at an extra cost since fuel oils are marketable while fuel gas is not. E.g. we see that the furnaces that can burn either fuel gas or fuel oil 1, do not burn any fuel gas ( $f_{06} = 0$ ) and burn 8.452 MT/h more fuel oil 1 ( $f_{02} = 15.57$ ) than in Model 4.

**Table B.6.** Optimal solutions for the flow and electricity variables

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Cost	2984.502	3108.774	16.296	2985.069	3109.560	16.260	3899.872
$x_{01}$	54.671	50.292	54.671	54.671	50.312	54.671	54.373
$x_{02}$	23.903	21.384	23.903	23.903	21.395	23.903	23.732
$x_{03}$	79.941	72.933	79.941	79.941	72.964	79.941	79.464
$x_{04}$	1.367	1.257	1.367	1.367	1.258	1.367	1.359
$x_{05}$	0.305	0.281	0.305	0.305	0.281	0.305	0.304
$x_{06}$	1.062	0.977	1.062	1.062	0.977	1.062	1.056
$x_{07}$	78.071	73.692	100.000	78.071	73.712	100.000	77.773
$x_{08}$	50.965	52.833	40.000	50.898	52.800	40.000	28.278
$x_{09}$	27.106	20.859	60.000	27.174	20.912	60.000	49.496
$x_{10}$	1.042	1.042	3.125	1.042	1.042	3.123	1.042
$x_{11}$	1.042	1.042	3.125	1.042	1.042	3.123	1.042
$x_{12}$	20.500	20.500	61.500	20.500	20.500	61.449	20.500
$x_{13}$	20.000	20.000	60.000	20.000	20.000	59.951	20.000
$x_{14}$	0.062	0.062	0.185	0.062	0.062	0.185	0.062
$x_{15}$	0.453	0.453	1.360	0.453	0.453	1.359	0.453
$x_{16}$	0.062	0.062	0.185	0.062	0.062	0.185	0.062
$x_{17}$	0.500	0.500	1.500	0.500	0.500	1.499	0.500
$x_{18}$	0.151	0.151	0.453	0.151	0.151	0.452	0.151
$x_{19}$	0.349	0.349	1.047	0.349	0.349	1.046	0.349
$x_{20}$	85.191	85.106	87.600	85.191	85.107	87.565	85.185
$x_{21}$	196.900	196.900	196.900	196.900	196.900	196.900	194.750
$x_{22}$	4.182	3.506	7.494	4.190	3.511	7.489	8.692
$x_{23}$	0.000	0.000	0.000	0.000	0.000	0.122	0.000
$x_{24}$	0.000	0.000	0.000	0.000	0.000	0.130	0.000
$x_{25}$	0.000	0.000	0.000	0.780	0.280	0.000	0.680
$x_{26}$	0.000	0.000	0.000	0.855	0.307	0.000	0.745
$x_{27}$	0.000	0.000	0.000	0.044	0.966	1.792	1.446
$x_{28}$	0.000	0.000	0.000	0.044	0.966	1.792	1.446
$x_{29}$	2.029	2.029	2.029	2.029	2.029	2.029	1.934
$x_{30}$	2.707	2.707	2.707	2.707	2.707	2.707	2.707
$x_{31}$	0.845	0.845	0.845	0.845	0.845	0.845	0.845
$x_{32}$	0.524	0.524	0.524	0.524	0.524	0.524	0.524
$x_{33}$	0.000	0.000	0.000	0.000	0.000	0.008	0.000
$x_{34}$	0.000	0.000	0.000	0.075	0.027	0.000	0.065
$x_{35}$	8.455	8.455	8.455	8.455	8.455	8.455	8.455
$x_{36}$	10.900	10.900	10.900	10.900	10.900	10.900	10.900
$x_{37}$	23.464	23.464	23.464	23.464	23.464	23.464	22.009
$x_{38}$	18.193	18.193	18.193	18.193	18.193	18.193	18.193
$x_{39}$	1.900	1.900	1.900	1.900	1.900	1.900	1.900
$x_{40}$	6.630	6.630	6.630	6.630	6.630	6.630	6.630
$x_{41}$	6.106	6.106	6.106	6.106	6.106	6.106	6.106
$x_{42}$	9.300	9.300	9.300	9.300	9.300	9.300	9.300
$x_{43}$	0.200	0.200	0.200	0.200	0.200	0.200	0.200
$x_{44}$	12.600	12.600	12.600	12.600	12.600	12.600	12.600
$x_{45}$	20.900	20.900	20.900	20.900	20.900	20.900	20.900
$x_{46}$	7.300	7.300	7.300	7.300	7.300	7.300	5.750
$x_{47}$	3.100	3.100	3.100	3.100	3.100	3.100	3.100
$x_{48}$	5.400	5.400	5.400	5.400	5.400	5.400	5.400
$x_{49}$	91.020	91.020	91.020	91.020	91.020	91.020	63.170
$x_{50}$	17.630	17.630	17.630	17.630	17.630	17.630	17.480
$x_{51}$	19.700	19.700	19.700	19.700	19.700	19.700	19.700
$x_{52}$	17.600	17.600	17.600	17.600	17.600	17.600	17.350
$x_{53}$	102.791	102.706	105.200	102.791	102.707	105.165	102.535
$x_{54}$	0.000	0.000	0.681	0.000	0.000	0.648	0.000
$w_{01}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$w_{02}$	31.558	24.461	31.558	31.558	24.492	31.558	31.075
$w_{03}$	11.824	10.497	18.677	11.833	10.506	18.677	14.631
$w_{04}$	1.061	1.043	2.060	1.036	1.002	2.025	1.059
$w_{05}$	36.000	36.000	36.000	36.000	36.000	36.000	35.538
$w_{06}$	8.443	0.000	16.296	8.427	0.000	16.260	11.228

**Table B.7.** Optimal solutions for the fuel, turbine, and unit variables

## C. Automatized Production: PCBs and Ulysses' Problem

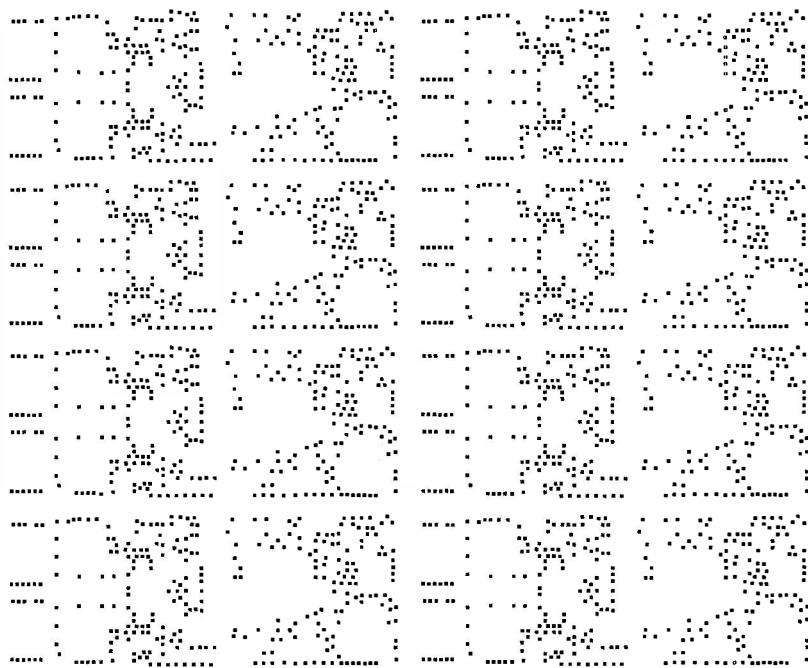
Printed circuit boards (PCBs) are omnipresent in today's high-technology world: you find them in washing machines, transistor radios, TVs, computers, etc. They are a typical example of a mass produced item the production process of which comprises several stages. One of the stages of this process is a drilling problem. In Figure C.1 we show the image of a flat, rectangular piece of material into which 2,392 holes must be drilled (by means of a mechanical drill or a laser gun). This is a necessary step in the production process of the PCB where the holes serve to place resistors, capacitors, etc on the board. Output per hour of this mass produced item is proportional to the time that it takes to complete the drilling of one board. The drill has to "visit" every hole exactly once, drilling time per hole is constant, and savings in the total processing time result from minimizing the travel time of the drill. For each hole the cartesian coordinates  $(x_u, y_u)$  for  $1 \leq u \leq 2,392$  are known. Depending upon the machinery used, the travel time of the drill can e.g. be proportional to the Euclidean distance of two consecutive holes to be drilled or it could e.g. be the distance of the two points in the  $\ell_1$ -norm. We assume for simplicity that the travel time is proportional to the Euclidean distance in our example, though the metric used does not matter much since it affects only the objective function of the problem. Figure C.2 shows the routing of the drill as produced by a professional scheduler in industry. Figure C.3 shows an optimal solution, i.e. one in which the total travel time in terms of the sum of the Euclidean distances is minimal. Given the enormous complexity of the scheduling task it is not surprising that the optimal solution is about half as long as the one constructed by the scheduler. The optimal solution was calculated using a branch-and-cut solver like the one shown in Figure 10.2, but with the additional feature of column generation as discussed in the context of the dynamic simplex algorithm of Chapter 6 to accommodate the 2,859,636 variables of the optimization problem. It took about 2 hours of CPU time on a time-sharing Sun-Sparc 10/41 workstation to optimize this particular problem; see also Figures 6.1 and 6.2 in Chapter 6.6 of the text.

The drilling problem for PCBs is one example of the so-called **traveling salesman problem** (TSP) which is a celebrated one-line problem that arises in so many different contexts of scheduling and routing that they are too numerous to be listed here in any detail.

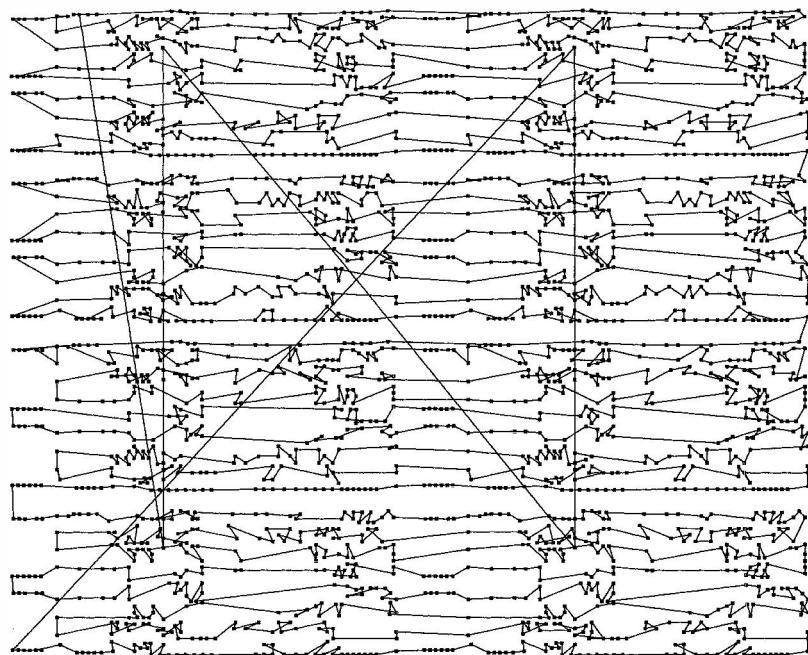
*Given  $n$  cities with arbitrary inter-city distances  $c_{uv}$  between cities  $u$  and  $v$  find a shortest roundtrip that visits every city exactly once.*

Interpreting the holes as "cities" this is exactly the problem that the scheduler faces when he routes the drill through the 2,392 holes to be drilled into the board. To solve his problem, the scheduler "invents" a roundtrip, see Figure C.2, programs the numerical drill to follow the particular sequence of his roundtrip and the machine produces a lot of 1,000, say, identical boards all having 2,392 holes in the same places following the programmed roundtrip; using the shortest roundtrip of Figure C.3 the output per hour could practically be doubled. To "optimize" the automatic production of PCBs – as one example of many for the application of TSPs – we must therefore, amongst other things, master the optimization of traveling salesman problems.

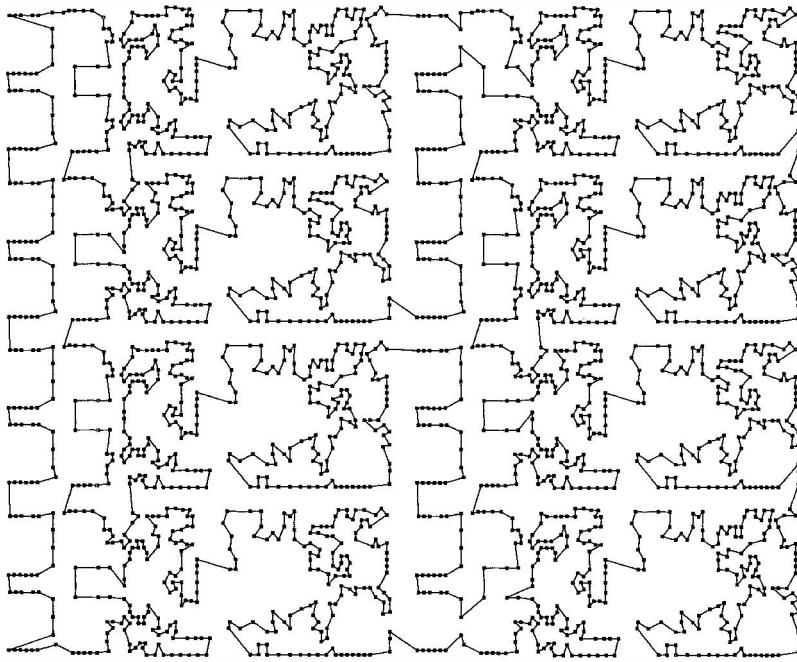
We formulate the combinatorial optimization problem of finding a shortest roundtrip for the drill or the traveling salesman as follows. Let  $V = \{1, \dots, n\}$  be the set of cities to be visited and denote by  $E = \{(u, v) : 1 \leq u < v \leq n\}$  the set of unordered pairs of cities. We call the pair  $G = (V, E)$  a *graph*,  $V$  the set of *nodes* of  $G$  and  $E$  the set of *edges* of  $G$ . There are exactly  $n(n - 1)/2$  edges in



**Fig. C.1.** PCB drilling problem with 2,392 points



**Fig. C.2.** A scheduler's solution for the 2,392 points of length 718,876



**Fig. C.3.** An optimal solution for the 2,392 points of length 378,032

our case and such graphs are called *complete graphs*. Since for a drill the distance  $c_{uv}$  from hole  $u$  to hole  $v$  is the same as the distance from  $v$  to  $u$  we do not distinguish the “direction” in which we traverse any edge  $e = (u, v)$ . This gives rise to a *symmetric* traveling salesman problem; the directed or asymmetric case of the TSP exists as well and is modeled in a similar fashion using directed edges or arcs. Here we will discuss only the symmetric version of the problem. Any roundtrip is a *simple cycle* in the graph  $G$ , i.e. a sequence  $(v_1, v_2, \dots, v_n)$  of the  $n$  distinct nodes that does not repeat any node. The corresponding edge set  $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1)$  is a graphical representation of a roundtrip or *tour*, for short. In graph theory tours are also called Hamiltonian cycles, so named after the Irish mathematician Sir William Rowan Hamilton (1805-1865). The length of a tour is the sum of the lengths of the edges that form the tour. The traveling salesman problem is thus the problem of finding a shortest tour or a Hamiltonian cycle in a graph  $G$ . You show (by induction on  $n \geq 3$ ) that there are precisely  $(n - 1)!/2$  tours in the graph  $G$  as we have defined it. In terms of our formulation (10.1) of combinatorial optimization problems we can thus proceed as follows: the ground set  $E$  is the set of all edges that are possible on  $n$  nodes and the family  $\mathcal{F}$  is given by

$$\mathcal{F} = \{T \subseteq E : T \text{ is the edge-set of a tour in } G\}.$$

To arrive at a formulation by way of a linear description we introduce a vector  $x \in \mathbb{R}^{n(n-1)/2}$  of

*decision variables*  $x_e$  for every edge  $e \in E$  as follows

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is selected,} \\ 0 & \text{if not.} \end{cases} \quad (\text{C.1})$$

So  $e \in E$  runs through all  $n(n-1)/2$  edges that are present in the complete graph  $G$ . In the case of the drilling problem of Figure C.1 this means that there are exactly 2,859,636 zero-one variables to be considered. Rather than writing  $\mathbb{R}^{n(n-1)/2}$  we write  $\mathbb{R}^E$ , for short.

If  $e = (u, v)$  then the edge  $e$  connects node  $u$  to node  $v$  and the integerized length of the edge in the Euclidean norm for the drilling problem is

$$c_e = \lfloor \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} + 0.5 \rfloor ,$$

where  $(x_u, y_u)$ ,  $(x_v, y_v)$  are the cartesian coordinates of the nodes  $u$  and  $v$ , respectively. We do not have to integerize, but it is more convenient to do so. The error committed this way is small enough, if the coordinates are integers having several digits or have been scaled to be such integers, which is often the case. The objective function of our problem is

$$\min \sum_{e \in E} c_e x_e ,$$

where we minimize subject to the condition that  $x = (x_e)_{e \in E}$ ,  $x \in \mathbb{R}^E$ , is the incidence vector of a tour in  $G$ . So in effect, we are looking for the minimum of a linear objective function over a discrete set of  $(n-1)!/2$  zero-one points in  $\mathbb{R}^{n(n-1)/2}$ .

We have to find now linear relations that express the requirement that we want incidence vectors of tours in the minimization. Since every node is met by exactly two edges of a tour, i.e. the drill comes from some hole and goes to some other hole before drilling the hole  $v$ , say, we have the equations

$$\sum_{e \text{ meets } v} x_e = 2 \quad \text{for all } v \in V . \quad (\text{C.2})$$

The equations (C.2) together with the 0-1 requirement (C.1) do not model our problem correctly. For let e.g.  $n = 6$ . Then the edge set

$$F = \{(1, 2), (2, 3), (3, 1), (4, 5), (5, 6), (6, 4)\}$$

gives a zero-one vector in  $\mathbb{R}^{15}$  that satisfies (C.1) and (C.2), but it evidently does not correspond to a tour in the graph on 6 nodes. However,

$$(1, 2), (2, 3), (3, 1)$$

is a tour on the graph with 3 nodes which is called a *subtour* for the graph with 6 nodes.

So more generally, for a graph with  $n$  nodes we have to forbid the occurrence of subtours on node sets  $S$  with  $3 \leq |S| \leq n-1$ . For  $S \subseteq V$  let us define

$$\begin{aligned} E(S) &= \{(u, v) \in E : u \in S, v \in S\} , \\ (S : V - S) &= \{(u, v) \in E : \text{either } u \text{ or } v \in S\} . \end{aligned}$$

$E(S)$  is the set of all edges in  $G$  having both endpoints in the set  $S$ .

$(S : V - S) = (V - S : S)$  is the set of edges of  $G$  that have exactly one endpoint in  $S$  and is called a *cut set* in  $G$ , since removing all edges in  $(S : V - S)$  for  $\emptyset \neq S \subset V$  disconnects the graph by breaking it into two disjoint parts. We can now formulate the *subtour elimination constraints* to rule out subtour solutions to (C.1) and (C.2) as follows

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \quad \text{for all } S \subset V, 2 \leq |S| \leq |V| - 1. \quad (\text{C.3})$$

The constraints reduce to  $x_e \leq 1$  where  $e$  is the unique edge in  $E(S)$  if  $|S| = 2$ , while for  $|S| \geq 3$  they rule out subtours as defined above. We claim that every zero-one vector  $\mathbf{x} \in \mathbb{R}^E$  satisfying (C.2) and (C.3) is the incidence vector of a tour. Let

$$F = \{e \in E : x_e = 1\}$$

for any such vector. Since every node of  $G$  is met by exactly two edges the set  $F$  must contain at least one cycle. Since  $|F| = n$  this cycle has exactly  $n$  edges, because otherwise a subtour elimination constraint (C.3) would be violated.

So the zero-one linear programming problem given by

$$(F_1) \quad \min\left\{\sum_{e \in E} c_e x_e : \mathbf{x} \in \mathbb{R}^E \text{ satisfies (C.1), (C.2), (C.3)}\right\}$$

models the symmetric traveling salesman problem in the form of a mixed-integer linear programming problem (MIP) of Chapter 10 and the polytope

$$Q_S^n = \{\mathbf{x} \in \mathbb{R}^E : \mathbf{x} \geq 0, \mathbf{x} \text{ satisfies (C.2) and (C.3)}\}$$

is a *formulation* of the problem (TSP) which, however, is known to be far from being an *ideal formulation* of the (symmetric) *traveling salesman polytope*

$$Q_T^n = \text{conv}(\{\mathbf{x} \in Q_S^n : x_e \in \{0, 1\} \text{ for all } e \in E\}),$$

i.e.,  $Q_T^n$  is in the terminology of Chapter 10 the convexification of the discrete set

$$DM = \{\mathbf{x} \in \mathbb{R}^E : \mathbf{x} \in Q_S^n, x_e \text{ integer for all } e \in E\}.$$

The extreme points of  $Q_T^n$  are precisely the incidence vectors of the  $(n - 1)!/2$  tours in  $G$ . It is not straightforward, but it is not too difficult either, to prove that

$$\dim Q_T^n = n(n - 3)/2, \quad \text{aff}(Q_T^n) = \{\mathbf{x} \in \mathbb{R}^E : \mathbf{x} \text{ satisfies (C.2)}\}.$$

So the traveling salesman polytope is a flat in  $\mathbb{R}^{n(n-1)/2}$  and consequently its ideal description is *quasi-unique*. To illustrate the quasi-uniqueness let us consider the subtour elimination constraints (C.3). Since every  $\mathbf{x} \in \mathbb{R}^E$  that satisfies (C.2) satisfies also

$$\sum_{v \in S} \left( \sum_{e \text{ meets } v} x_e \right) = 2 \sum_{e \in E(S)} x_e + \sum_{e \in (S : V - S)} x_e = 2|S|,$$

**Table C.1.** The facial structure of small traveling salesman polytopes

$n$	3	4	5	6	7	8	9	10
No. of variables	3	6	10	15	21	28	36	45
Dimension	0	2	5	9	14	20	27	35
No. of tours	1	3	12	60	360	2,520	20,160	181,440
Equations	3	4	5	6	7	8	9	10
Facets	0	3	20	100	3,437	194,187	42,104,442	$\geq 51,043,900,866$

for every  $S \subseteq V$ , it follows that the constraints (C.3) can be written *equivalently* as follows

$$\sum_{e \in (S : V - S)} x_e \geq 2 \quad \text{for all } S \subset V, 2 \leq |S| \leq |V| - 1, \quad (\text{C.4})$$

where  $(S : V - S)$  is the cut set of edges in  $G$  defined by the nodes in  $S$  and  $V - S$ . The logical interpretation of (C.4) is clear and we leave it to you to figure it out. Since (C.3) and (C.4) are equivalent constraints for all  $\mathbf{x} \in \mathbb{R}^E$  satisfying (C.2) we can write the above formulation  $Q_S^n$  for (TSP) as follows

$$(\text{F}_2) \quad Q_S^n = \{\mathbf{x} \in \mathbb{R}^E : \mathbf{x} \geq \mathbf{0}, \mathbf{x} \text{ satisfies (C.2) and (C.4)}\},$$

which is “visually” different from, but mathematically equivalent to (F<sub>1</sub>).

Counting the constraints (C.3) or (C.4) we find that there are  $2^n - n - 2$  constraints, i.e. our formulation of (TSP) has  $\mathcal{O}(2^n)$  or exponentially many constraints. So if we want to solve TSPs using linear programming, then the straightforward way of *listing* all constraints of the problem is out of the question. By Remark 9.20 we know, however, that the linear optimization problem

$$(\text{TSP}_{LP}) \quad \min\{\mathbf{c}\mathbf{x} : \mathbf{x} \in Q_S^n\}$$

can be solved in polynomial time if we can solve the polyhedral separation problem for  $Q_S^n$  likewise.

So let  $\mathbf{x}^* \in \mathbb{R}^E$  be arbitrary. Using the algorithm LIST-and-CHECK we can check the constraints (C.2) and the constraints  $0 \leq x_e \leq 1$  for all  $e \in E$  in polynomial time. So suppose  $\mathbf{x}^* \in \mathbb{R}^E$  satisfies those constraints. For  $S \subseteq V$  we write for short

$$\mathbf{x}(S : V - S) = \sum_{e \in (S : V - S)} x_e, \quad \mathbf{x}(S) = \sum_{e \in E(S)} x_e.$$

To check the exponentially many constraints (C.4) for  $\mathbf{x}^* \in \mathbb{R}^E$  we must solve the problem

$$\min\{\mathbf{x}^*(S : V - S) : S \subset V, 2 \leq |S| \leq |V| - 1\}, \quad (\text{C.5})$$

which is a *minimum cut problem* in the *weighted* graph  $(G, \mathbf{x}^*)$  where each edge  $e \in E$  gets the weight  $x_e^*$ . If  $S \subseteq V$  solves the minimum cut problem and

$$\mathbf{x}^*(S : V - S) \geq 2,$$

then  $\mathbf{x}^* \in Q_S^n$ , whereas otherwise the inequality  $\mathbf{h}\mathbf{x} = \mathbf{x}(S) \leq |S| - 1$  is a most violated separator for  $\mathbf{x}^*$  and  $Q_S^n$ ; see Chapter 9.5. Several polynomial-time algorithms for the minimum cut problem

are known and thus – in spite of the exponential size of the formulation  $Q_S^n$  of TSP – we can solve  $(\text{TSP}_{LP})$  in polynomial time using the ellipsoid algorithm.

In computational practice we will, of course, use the *fastest* LP solver in the branch-and-cut framework of Figure 10.2. This can be a simplex algorithm, a barrier method or whatever algorithm is fastest when you need to solve the problem. The important observation to make is that the constraint generator now comprises at least one algorithm different from the one that solves linear programs. Here again you have a choice: you will choose the fastest available algorithm to solve your minimum cut problem.

The linear program  $(\text{TSP}_{LP})$  frequently provides a zero-one solution, i.e. a solution to the traveling salesman problem, but of course not always. Indeed, the *facial structure* of  $Q_T^n$  is extremely complicated and has been the object of many investigations for many years. Ideal descriptions are known for “small”  $n$  and in Table C.1 we show the number of distinct facets, etc of  $Q_T^n$  for values of  $n$  satisfying  $3 \leq n \leq 10$ , which demonstrate the incredible complexity of the TSP polytope.

The subtour elimination constraints (C.3) define facets of  $Q_T^n$  and half of them define distinct facets of the polytope. Consequently, they or equivalent representations like (C.4) of them are *required* in every minimal and complete, i.e. ideal, linear description of the polytope  $Q_T^n$ . Let  $H \subseteq V$  and  $T_i \subseteq V$  for  $1 \leq i \leq k$ , where  $k \geq 3$  is an odd integer, be any subsets of nodes of  $G$  satisfying

$$|H \cap T_i| \geq 1, |T_i - H| \geq 1 \text{ for } 1 \leq i \leq k, T_i \cap T_j = \emptyset \text{ for } 1 \leq i < j \leq k. \quad (\text{C.6})$$

The configuration  $C = (H, T_1, \dots, T_k)$  in the graph  $G$  is called a *comb* in  $G$  with  $H$  being the “handle”,  $T_i$  being the “teeth” of the comb; see Figure C.4 for two illustrations. The inequality

$$\mathbf{x}(H) + \sum_{i=1}^k \mathbf{x}(T_i) \leq |H| + \sum_{i=1}^k (|T_i| - 2) + \lfloor \frac{k}{2} \rfloor \quad (\text{C.7})$$

is called a *comb inequality* and it is facet-defining for  $Q_T^n$  as well for all  $n \geq 6$ . Let  $C = (H, T_1, \dots, T_k)$  be a comb in  $G$  and partition  $T_i$  into  $r_i \geq 1$  nonempty sets  $T_i^j$  such that

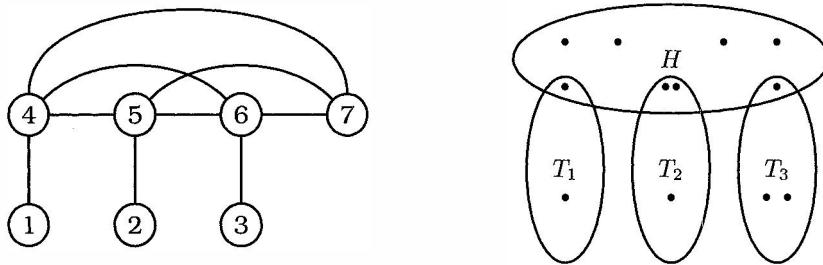
$$|H \cap T_i^j| \geq 1, |T_i^j - H| \geq 1 \text{ for } 1 \leq j \leq r_i, T_i^j \cap T_i^\ell = \emptyset \text{ for } 1 \leq j < \ell \leq r_i$$

holds for all  $1 \leq i \leq k$ . Then the *refined comb (r-comb) inequality*

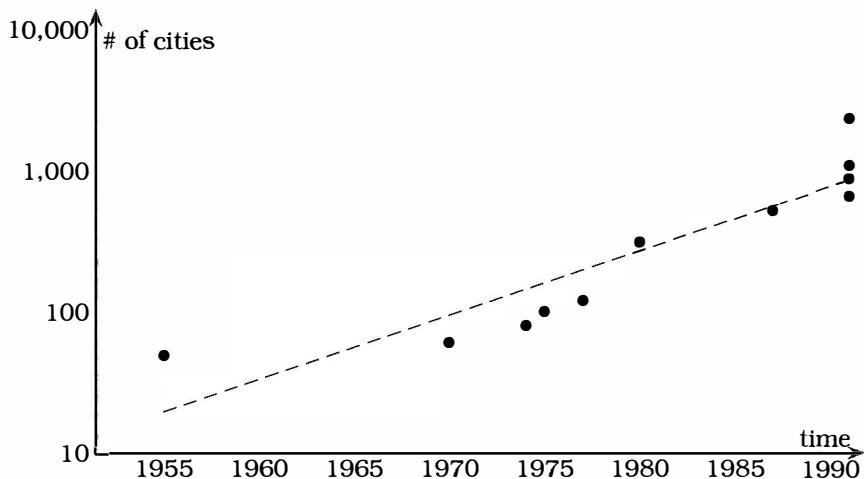
$$\begin{aligned} \mathbf{h}^C \mathbf{x} &:= \mathbf{x}(H) + \sum_{i=1}^k \left( \sum_{j=1}^{r_i} \mathbf{x}(T_i^j) \right) + \sum_{1 \leq \ell < j \leq r_i} (\mathbf{x}(T_i^\ell \cap H : T_i^j - H) + \mathbf{x}(T_i^\ell - H : T_i^j \cap H)) \\ &\leq |H| + \sum_{i=1}^k (|T_i| - r_i - 1) + \lfloor \frac{k}{2} \rfloor \end{aligned}$$

generalizes the comb inequality and it is facet defining as well for  $Q_T^n$  and  $n \geq 6$ . To prove the validity of the *r-comb* inequality is easy and it goes as follows. We write

$$\begin{aligned} 2\mathbf{h}^C \mathbf{x} &\leq \sum_{v \in H} \left( \sum_{e \text{ meets } v} x_e \right) + \sum_{i=1}^k (\mathbf{x}(T_i) + \sum_{j=1}^{r_i} (\mathbf{x}(T_i^j \cap H) + \mathbf{x}(T_i^j - H))) \\ &\leq 2|H| + \sum_{i=1}^k (|T_i| - 1 + \sum_{j=1}^{r_i} (|T_i^j \cap H| - 1 + |T_i^j - H| - 1)) = 2|H| + 2 \sum_{i=1}^k (|T_i| - r_i - 1) + k, \end{aligned}$$



**Fig. C.4.** Graphical representations of comb inequalities



**Fig. C.5.** Progress in the exact optimization of symmetric TSPs

where we have used (C.2), (C.3) and the nonnegativity of  $\boldsymbol{x}$ . Dividing both sides of the inequality by 2 and rounding the right-hand side down to the nearest integer the validity of the  $r$ -comb inequality for  $Q_T^n$  follows because  $h^C \boldsymbol{x}$  is an integer number for every zero-one vector  $\boldsymbol{x}$ . Note that the coefficients of the  $r$ -comb inequalities are 0, 1 or 2.  $R$ -comb inequalities are facet defining for the traveling salesman polytope  $Q_T^n$  for all  $n \geq 6$  and half of them define distinct facets of  $Q_T^n$ . To prove these assertions goes beyond the aims of our exposition and we refer you to the references. Many more, facet-defining inequalities for  $Q_T^n$  are known. Nevertheless, an ideal description of  $Q_T^n$  might possibly prove to be elusive. But then the outer inclusion principle of Chapters 7.5.4 and 9.5 applies here as well; see our discussion of this point in Chapter 10.3.

The total number of  $r$ -comb inequalities is enormous; it grows far worse with  $n$  than the number of subtour elimination constraints. To use the inequalities in the numerical solution of TSPs we must therefore find algorithms that solve the corresponding polyhedral separation problem for the polytope  $Q_C^n$ , say, that results by intersecting the polytope  $Q_S^n$  by the totality of all  $r$ -comb inequalities that are possible in the complete graph on  $n$  nodes. For the special case of comb inequalities satisfying

$$|H \cap T_i| = 1, \quad |T_i - H| = 1 \quad \text{for } 1 \leq i \leq k, \quad (\text{C.8})$$

a polynomial-time separation algorithm is known. For a given  $x^* \in \mathbb{R}^E$  the problem that needs to be solved is a *minimum odd cut problem* in a suitably constructed weighted graph  $(\widehat{G}, \widehat{x}^*)$  that has, roughly, twice the size of the weighted graph  $(G, x^*)$  and whose nodes are labelled “odd” or “even”. The problem that must be solved in  $(\widehat{G}, \widehat{x}^*)$  is essentially the problem (C.5) with the added requirement that the subsets  $S \subset \widehat{V}$  contain an odd number of odd labelled nodes where  $\widehat{V}$  is the node set of the graph  $\widehat{G}$ . Like we reasoned above we can thus – theoretically – optimize in polynomial time the linear program that results when the formulation  $Q_S^n$  is intersected with the totality of all combs satisfying (C.8). For general comb constraints no *exact* algorithms are known to date that solve the separation problem. It remains a challenge to find such algorithms. Several heuristics for the separation (or constraint identification) problem for combs are known and we refer you to the references for more detail.

The preceding *partial* characterization of ideal formulations of symmetric traveling salesman problems yields the theoretical basis for a powerful branch-and-cut solver for this notorious problem. The number of cities  $n$  is arbitrary and typically, the larger  $n$  the more difficult it becomes to find an optimal tour for a particular instance of the problem. In Figure C.5 we show the progress of exact optimization of symmetric TSPs from the 1950's to the early 1990's in a city-time diagram according to the *publication* dates which you will find in the references. Note the logarithmic scale (base 10) on the cities-axis. The problem with  $n = 2,392$  is not the largest real-world problem solved to date. Research along these lines continues around the globe and succeeds in pushing the frontiers of the exact solvability of combinatorial problems further and further. Most – if not all – of the *exact* algorithms that are used in this endeavor are of the branch-and-cut variety and utilize partial linear descriptions of ideal formulations of the combinatorial problems – like we have discussed it in full generality in Chapter 10.

To give you an operational understanding of the material of this appendix we have included two small numerical problems. The first problem is a 16-city problem, the second one a 48-city problem. We want you to solve both problems and recommend that you start by reading G.B. Dantzig, D.R. Fulkerson and S.M. Johnson's article “Solution of a large-scale traveling salesman problem”, *Operations Research* 2 (1954) 393–410. Rather than writing a branch-and-cut algorithm – which you should be able to do – it might be a good idea to solve the problem first visually like Dantzig, Fulkerson and Johnson did in 1954. Their work is a milestone in the **exact optimization** of large-scale combinatorial problems and remained a *world record* in terms of problem size for over 15 years.

In Table C.2 we list 16 Mediterranean tourist spots that according to Homer were visited by Ulysses, a former king of the island of Ithaca (Greece), a long time ago. Table C.2 also gives their polar coordinates from which we have computed the table of the inter-city distances in kilometers for your convenience, see Table C.3. The distances are the great-circle distances on the globe which are proportional to the time it takes to fly from city  $u$  to city  $v$ , where  $1 \leq u < v \leq 16$ . Our modern-day Ulysses will not criss-cross the Mediterranean like Homer's Ulysses did who visited the sixteen spots in the following order

$$1, 2, 3, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12, 13, 14, 15, 16, 1.$$

This is evidently not a tour, but then according to Homer's *Odyssey* the ancient Ulysses' voyage was quite an *odyssey* – an ordeal if you wish. Having participated in the Trojan war, all that the king truly wanted was to get home safely to his beloved Penelope. Our Ulysses wants to visit

**Table C.2.** Polar coordinates of 16 locations in the Mediterranean

1	Ithaca, Greece	38.24N	20.42E
2	Troy, Turkey	39.57N	26.15E
3	Maronia, Turkey	40.56N	25.32E
4	Malea, Greece	36.26N	23.12E
5	Djerba, Tunisia	33.48N	10.54E
6	Favignana, Italy	37.56N	12.19E
7	Ustica, Italy	38.42N	13.11E
8	Zakinthos, Greece	37.52N	20.44E
9	Bonifacio, France	41.23N	9.10E
10	Circeo, Italy	41.17N	13.05E
11	Gibraltar, Spain	36.08N	5.21W
12	Stromboli, Italy	38.47N	15.13E
13	Messina, Italy	38.15N	15.35E
14	Taormina, Italy	37.51N	15.17E
15	Birzebbuga, Malta	35.49N	14.32E
16	Corfu, Greece	39.36N	19.56E

the sixteen places without wasting his time – time is money for him, but maybe he, too, has an amorous motivation. For his journey he has, of course, a helicopter at his disposal. You are charged to find not only the shortest roundtrip for him, see Figure C.6, but more.

- (A) Calculate a lower bound on the total length of the ancient Ulysses' voyage.
- (B) Formulate Ulysses' problem and solve the associated linear program consisting only of the equations (C.2) and the constraints  $0 \leq x_e \leq 1$  for all  $e \in E$ . Make a map of the Mediterranean like Figure C.6. Plot the solution graphically and identify violated subtour elimination and/or comb constraints. Add all violated ones that you found to your linear program, reoptimize the linear program and iterate.
- (C) Devise a method to prove or disprove the *uniqueness* of the optimal tour you found in part (B). What is the second best tour for Ulysses?

The second problem that we want you to solve is a traveling salesman problem with 48 cities. It is different from the 1954 problem of the article by Dantzig et al who considered the largest cities in each one of the 48 continental states of U.S.A. plus Washington, D.C., and not the state capitals. The “virtual” coordinates of the 48 cities are given in Table C.4 and the corresponding inter-city distances for cities  $u$  and  $v$  are calculated by the following computer instructions in pseudo-code.

---

```

XDIST =((xu - xv) * (xu - xv) + (yu - yv) * (yu - yv))/10
XDIST =SQRT(XDIST)
MDIST =XDIST
if MDIST < XDIST then MDIST=MDIST+1

```

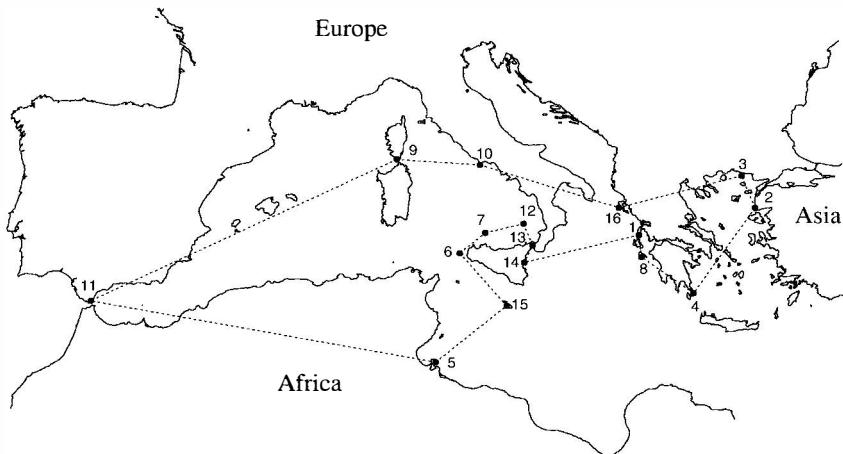
---

**Table C.3.** The distance table for Ulysses' problem

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	509	501	312	1019	736	656	60	1039	726	2314	479	448	479	619	150
2	126	474	1526	1226	1133	532	1449	1122	2789	958	941	978	1127	542	
3		541	1516	1184	1084	536	1371	1045	2728	913	904	946	1115	499	
4			1157	980	919	271	1333	1029	2553	751	704	720	783	455	
5				478	583	996	858	855	1504	677	651	600	401	1033	
6					115	740	470	379	1581	271	289	261	308	687	
7						667	455	288	1661	177	216	207	343	592	
8							1066	759	2320	493	454	479	598	206	
9								328	1387	591	650	656	776	933	
10									1697	333	400	427	622	610	
11										1838	1868	1841	1789	2248	
12											68	105	336	417	
13												52	287	406	
14													237	449	
15														636	

The virtual coordinates  $(x_u, y_u)$  given in Table C.4 are read into the computer for  $1 \leq u \leq 48$  and stored in double precision (64 bit) words. XDIST is a double precision word, MDIST a 32 bit integer. The preceding instructions serve to calculate the integerized length  $c_e = \text{MDIST}$  of the edge  $e = (u, v)$ . The coordinates are called “virtual” because they approximate the “true” distance on the curved North American continent via a planar, Euclidean distance calculation. You can solve this problem like you did in the case of Ulysses’ problem. But we recommend that you automatize at least parts of the procedure.

- (D) Write a problem generator in a computer language of your choice that automatically sets up the linear program for TSPs that you want to solve in the format required by your LP solver.
- (E) Given a node set  $S \subseteq V$  or a comb  $C = (H, T_1, \dots, T_k)$  in a graph  $G$  write a routine that automatically sets up the linear inequality (C.3) or (C.7) in the format required by your LP solver.
- (F) Now solve the 48-city problem like you did above starting the calculations like in part (A). Plot each linear programming solution on a map of the United States of America, like the one of Figure C.7 and iterate. Put all violated subtour elimination and comb constraints that you find in the course of your calculations into a memory “pool”, i.e. store them in some separate computer file.
- (G) Having solved the problem the “hard” way – you should be able to do so, without branching, on the first try by solving about 15-25 linear programs – do it again, but in the following way: at every iteration take from the memory pool that you created in the first run *all* constraints that are violated by the current LP optimum (but not the ones that are not violated!) and iterate. How many iterations do you need now to solve the problem? Why?



**Fig. C.6.** An optimal solution for Ulysses' problem of length 6,859 km

- (H) (Optional) Now suppose that comb constraints are unknown to you and that you know only the subtour elimination constraints (C.3). Solve the problem again using only constraints (C.3), i.e. solve the corresponding problem  $(TSP_{LP})$ . What is its optimal value? Using branch-and-bound in a suitably modified form (zero-one solutions that correspond to subtours must be cut off by constraints of the form (C.3)!) find an optimal solution to the TSP. State your rules for the selection of branching variables and the selection of the “next” problem from the problem stack clearly. How many nodes does your search tree have? Plot the search tree like we did in Figure 10.1.

The literature on the traveling salesman problem has grown exponentially in the last decade. In the references to Appendix C of the text you will find several survey papers that should permit you to locate many additional articles and works that we have omitted.

**Table C.4.** Virtual coordinates of the 48 state capitals of the continental U.S.A.

$u$	City	$x_u$	$y_u$	$u$	City	$x_u$	$y_u$
1	Montgomery, AL	7692	2247	25	Lincoln, NE	6823	4674
2	Phoenix, AZ	9135	6748	26	Carson City, NV	8139	8306
3	Little Rock, AR	7721	3451	27	Concord, NH	4326	1426
4	Sacramento, CA	8304	8580	28	Trenton, NJ	5164	1440
5	Denver, CO	7501	5899	29	Santa Fe, NM	8389	5804
6	Hartford, CT	4687	1373	30	Albany, NY	4639	1629
7	Dover, DE	5429	1408	31	Raleigh, NC	6344	1436
8	Tallahassee, FL	7877	1716	32	Bismarck, ND	5840	5736
9	Atlanta, GA	7260	2083	33	Columbus, OH	5972	2555
10	Boise, ID	7096	7869	34	Oklahoma City, OK	7947	4373
11	Springfield, IL	6539	3513	35	Salem, OR	6929	8958
12	Indianapolis, IN	6272	2992	36	Harrisburg, PA	5366	1733
13	Des Moines, IA	6471	4275	37	Providence, RI	4550	1219
14	Topeka, KS	7110	4369	38	Columbia, SC	6901	1589
15	Frankfort, KY	6462	2634	39	Pierre, SD	6316	5497
16	Baton Rouge, LA	8476	2874	40	Nashville, TN	7010	2710
17	Augusta, ME	3961	1370	41	Austin, TX	9005	3996
18	Annapolis, MD	5555	1519	42	Salt Lake City, UT	7576	7065
19	Boston, MA	4422	1249	43	Montpelier, VT	4246	1701
20	Lansing, MI	5584	3081	44	Richmond, VA	5906	1472
21	St. Paul, MN	5776	4498	45	Olympia, WA	6469	8971
22	Jackson, MS	8035	2880	46	Charleston, WV	6152	2174
23	Jefferson City, MO	6963	3782	47	Madison, WI	5887	3796
24	Helena, MT	6336	7348	48	Cheyenne, WY	7203	5958

## C.1 Solutions to Ulysses' Problem.

**(A)** Calculate a lower bound on the total length of the ancient Ulysses' voyage.

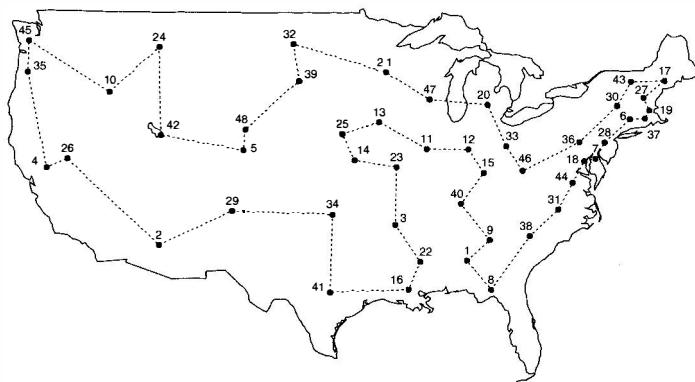
We give two ways to find a lower bound to the length of the shortest tour. The first one is based on the simple observation that if we remove an edge from a tour then we get a spanning tree, which in this case will be a simple path that goes through every city. Therefore the length of the optimal tour is strictly less than the length of a spanning tree. It follows that if we calculate the minimum length of a spanning tree then we immediately have a lower bound to the length of the optimal tour. The algorithm to calculate the minimum spanning tree is very simple and can be described as follows:

Step 1: Order the edges of the graph in ascending order. The number of selected edges is zero.

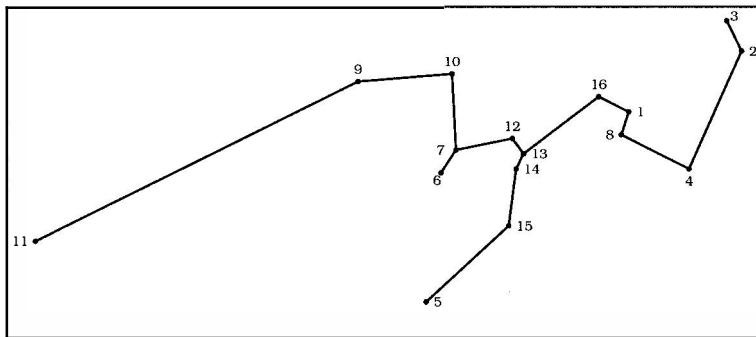
Step 2: **while** the number of selected edges is less than  $n$

select the shortest edge that does not form a cycle with the previous selected edges

**end**



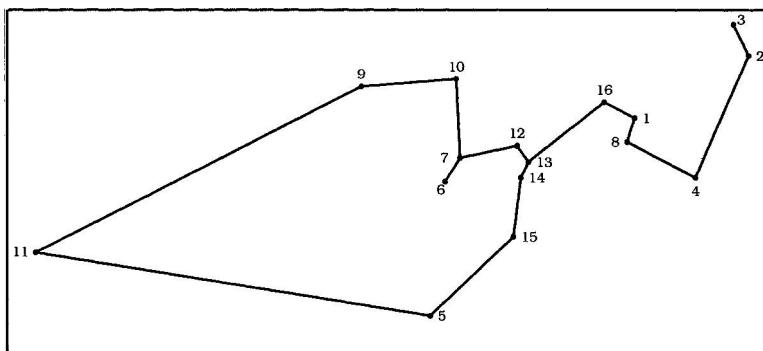
**Fig. C.7.** Optimal solution of the 48 city problem of length 10,628 miles



**Fig. C.8.** Minimum length spanning tree for Ulysses' problem

Applying this algorithm to the data of the Ulysses' problem we get the solution shown in Figure C.8. The length of the tree is 4,540 km. As will become clear later, this lower bound is very weak.

Every solution to the traveling salesman problem, that is, every tour, contains one cycle, it has as many edges as the number of nodes of the graph, is connected, and an arbitrarily chosen node is connected to exactly two distinct nodes and is contained in the cycle. Dropping any of these properties, we obtain a set of solutions that *properly contains* the set of all solutions to the TSP, and thus minimizing over these solutions we obtain a lower bound to the optimal length of a tour. For example, dropping the last property we get the so-called *1-trees* (in fact a more restricted class of graphs). The algorithm to calculate the minimum length of such an 1-tree is very simple. We delete the special node and calculate the minimum spanning tree on the reduced graph. Then we connect the special node to the two nodes that are closer to it. The solution is shown in Figure C.9 and has a length of 6,044 km. The lower bound obtained is much tighter



**Fig. C.9.** Minimum-length 1-tree with special node 11

than the one obtained by the minimum spanning tree calculation in this case.

The above values are lower bounds on the **shortest** roundtrip that Ulysses could have followed to visit the sixteen spots, which he did not. Indeed, we know from Homer's *Odyssey* the zig-zagging trip that Ulysses took. According to the story, Ulysses visited and revisited the sixteen tourist spots in the following order

1, 2, 3, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12, 13, 14, 15, 16, 1.

Adding the corresponding distances from Table C.3 we thus find that the ancient Ulysses must have traveled at least 9,913 km as his ship followed the whims and the currents of the sea.

**(B)** *Formulate Ulysses' problem and solve the associated linear program consisting only of the equations (C.2) and the constraints  $0 \leq x_e \leq 1$  for all  $e \in E$ . Make a map of the Mediterranean like Figure C.6. Plot the solution graphically and identify violated subtour elimination and/or comb constraints. Add all violated ones that you found to your linear program, reoptimize the linear program and iterate.*

The formulation of the initial LP, i.e., the one consisting of the degree constraints and the bound constraints is as follows (in CPLEX lp format), where  $x_{ij}$  are the decision variables and both  $i$  and  $j$  are two digit numbers, e.g.,  $x_{0102}$  models the edge from node 1 to node 2 and so forth.

Minimize

$$\begin{aligned}
 \text{obj: } & 509 x_{0102} + 501 x_{0103} + 312 x_{0104} + 1019 x_{0105} + 736 x_{0106} + 656 x_{0107} \\
 & + 60 x_{0108} + 1039 x_{0109} + 726 x_{0110} + 2314 x_{0111} + 479 x_{0112} + 448 x_{0113} \\
 & + 479 x_{0114} + 619 x_{0115} + 150 x_{0116} + 126 x_{0203} + 474 x_{0204} + 1526 x_{0205} \\
 & + 1226 x_{0206} + 1133 x_{0207} + 532 x_{0208} + 1449 x_{0209} + 1122 x_{0210} + 2789 x_{0211} \\
 & + 958 x_{0212} + 941 x_{0213} + 978 x_{0214} + 1127 x_{0215} + 542 x_{0216} + 541 x_{0304} \\
 & + 1516 x_{0305} + 1184 x_{0306} + 1084 x_{0307} + 536 x_{0308} + 1371 x_{0309} + 1045 x_{0310} \\
 & + 2728 x_{0311} + 913 x_{0312} + 904 x_{0313} + 946 x_{0314} + 1115 x_{0315} + 499 x_{0316} \\
 & + 1157 x_{0405} + 980 x_{0406} + 919 x_{0407} + 271 x_{0408} + 1333 x_{0409} + 1029 x_{0410} \\
 & + 2553 x_{0411} + 751 x_{0412} + 704 x_{0413} + 720 x_{0414} + 783 x_{0415} + 455 x_{0416}
 \end{aligned}$$

$$\begin{aligned}
& + 478 x_{0506} + 583 x_{0507} + 996 x_{0508} + 858 x_{0509} + 855 x_{0510} + 1504 x_{0511} \\
& + 677 x_{0512} + 651 x_{0513} + 600 x_{0514} + 401 x_{0515} + 1033 x_{0516} + 115 x_{0607} \\
& + 740 x_{0608} + 470 x_{0609} + 379 x_{0610} + 1581 x_{0611} + 271 x_{0612} + 289 x_{0613} \\
& + 261 x_{0614} + 308 x_{0615} + 687 x_{0616} + 667 x_{0708} + 455 x_{0709} + 288 x_{0710} \\
& + 1661 x_{0711} + 177 x_{0712} + 216 x_{0713} + 207 x_{0714} + 343 x_{0715} + 592 x_{0716} \\
& + 1066 x_{0809} + 759 x_{0810} + 2320 x_{0811} + 493 x_{0812} + 454 x_{0813} + 479 x_{0814} \\
& + 598 x_{0815} + 206 x_{0816} + 328 x_{0910} + 1387 x_{0911} + 591 x_{0912} + 650 x_{0913} \\
& + 656 x_{0914} + 776 x_{0915} + 933 x_{0916} + 1697 x_{1011} + 333 x_{1012} + 400 x_{1013} \\
& + 427 x_{1014} + 622 x_{1015} + 610 x_{1016} + 1838 x_{1112} + 1868 x_{1113} + 1841 x_{1114} \\
& + 1789 x_{1115} + 2248 x_{1116} + 68 x_{1213} + 105 x_{1214} + 336 x_{1215} + 417 x_{1216} \\
& + 52 x_{1314} + 287 x_{1315} + 406 x_{1316} + 237 x_{1415} + 449 x_{1416} + 636 x_{1516}
\end{aligned}$$

Subject To

$$\begin{aligned}
c1: & \quad x_{0102} + x_{0103} + x_{0104} + x_{0105} + x_{0106} + x_{0107} + x_{0108} + x_{0109} + x_{0110} \\
& + x_{0111} + x_{0112} + x_{0113} + x_{0114} + x_{0115} + x_{0116} = 2 \\
c2: & \quad x_{0102} + x_{0203} + x_{0204} + x_{0205} + x_{0206} + x_{0207} + x_{0208} + x_{0209} + x_{0210} \\
& + x_{0211} + x_{0212} + x_{0213} + x_{0214} + x_{0215} + x_{0216} = 2 \\
c3: & \quad x_{0103} + x_{0203} + x_{0304} + x_{0305} + x_{0306} + x_{0307} + x_{0308} + x_{0309} + x_{0310} \\
& + x_{0311} + x_{0312} + x_{0313} + x_{0314} + x_{0315} + x_{0316} = 2 \\
c4: & \quad x_{0104} + x_{0204} + x_{0304} + x_{0405} + x_{0406} + x_{0407} + x_{0408} + x_{0409} + x_{0410} \\
& + x_{0411} + x_{0412} + x_{0413} + x_{0414} + x_{0415} + x_{0416} = 2 \\
c5: & \quad x_{0105} + x_{0205} + x_{0305} + x_{0405} + x_{0506} + x_{0507} + x_{0508} + x_{0509} + x_{0510} \\
& + x_{0511} + x_{0512} + x_{0513} + x_{0514} + x_{0515} + x_{0516} = 2 \\
c6: & \quad x_{0106} + x_{0206} + x_{0306} + x_{0406} + x_{0506} + x_{0607} + x_{0608} + x_{0609} + x_{0610} \\
& + x_{0611} + x_{0612} + x_{0613} + x_{0614} + x_{0615} + x_{0616} = 2 \\
c7: & \quad x_{0107} + x_{0207} + x_{0307} + x_{0407} + x_{0507} + x_{0607} + x_{0708} + x_{0709} + x_{0710} \\
& + x_{0711} + x_{0712} + x_{0713} + x_{0714} + x_{0715} + x_{0716} = 2 \\
c8: & \quad x_{0108} + x_{0208} + x_{0308} + x_{0408} + x_{0508} + x_{0608} + x_{0708} + x_{0809} + x_{0810} \\
& + x_{0811} + x_{0812} + x_{0813} + x_{0814} + x_{0815} + x_{0816} = 2 \\
c9: & \quad x_{0109} + x_{0209} + x_{0309} + x_{0409} + x_{0509} + x_{0609} + x_{0709} + x_{0809} + x_{0910} \\
& + x_{0911} + x_{0912} + x_{0913} + x_{0914} + x_{0915} + x_{0916} = 2 \\
c10: & \quad x_{0110} + x_{0210} + x_{0310} + x_{0410} + x_{0510} + x_{0610} + x_{0710} + x_{0810} + x_{0910} \\
& + x_{1011} + x_{1012} + x_{1013} + x_{1014} + x_{1015} + x_{1016} = 2 \\
c11: & \quad x_{0111} + x_{0211} + x_{0311} + x_{0411} + x_{0511} + x_{0611} + x_{0711} + x_{0811} + x_{0911} \\
& + x_{1011} + x_{1112} + x_{1113} + x_{1114} + x_{1115} + x_{1116} = 2 \\
c12: & \quad x_{0112} + x_{0212} + x_{0312} + x_{0412} + x_{0512} + x_{0612} + x_{0712} + x_{0812} + x_{0912} \\
& + x_{1012} + x_{1112} + x_{1213} + x_{1214} + x_{1215} + x_{1216} = 2 \\
c13: & \quad x_{0113} + x_{0213} + x_{0313} + x_{0413} + x_{0513} + x_{0613} + x_{0713} + x_{0813} + x_{0913} \\
& + x_{1013} + x_{1113} + x_{1213} + x_{1314} + x_{1315} + x_{1316} = 2 \\
c14: & \quad x_{0114} + x_{0214} + x_{0314} + x_{0414} + x_{0514} + x_{0614} + x_{0714} + x_{0814} + x_{0914} \\
& + x_{1014} + x_{1114} + x_{1214} + x_{1314} + x_{1415} + x_{1416} = 2 \\
c15: & \quad x_{0115} + x_{0215} + x_{0315} + x_{0415} + x_{0515} + x_{0615} + x_{0715} + x_{0815} + x_{0915} \\
& + x_{1015} + x_{1115} + x_{1215} + x_{1315} + x_{1415} + x_{1516} = 2 \\
c16: & \quad x_{0116} + x_{0216} + x_{0316} + x_{0416} + x_{0516} + x_{0616} + x_{0716} + x_{0816} + x_{0916} \\
& + x_{1016} + x_{1116} + x_{1216} + x_{1316} + x_{1416} + x_{1516} = 2
\end{aligned}$$

Bounds

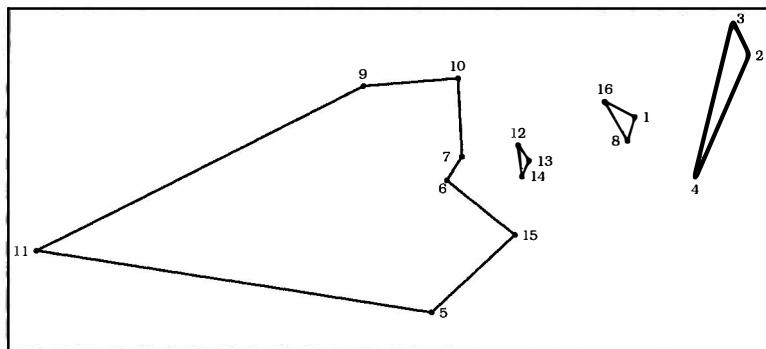
$$\begin{aligned}
0 &\leq x_{0102} \leq 1 \quad 0 \leq x_{0305} \leq 1 \quad 0 \leq x_{0512} \leq 1 \quad 0 \leq x_{0815} \leq 1 \\
0 &\leq x_{0103} \leq 1 \quad 0 \leq x_{0306} \leq 1 \quad 0 \leq x_{0513} \leq 1 \quad 0 \leq x_{0816} \leq 1 \\
0 &\leq x_{0104} \leq 1 \quad 0 \leq x_{0307} \leq 1 \quad 0 \leq x_{0514} \leq 1 \quad 0 \leq x_{0910} \leq 1 \\
0 &\leq x_{0105} \leq 1 \quad 0 \leq x_{0308} \leq 1 \quad 0 \leq x_{0515} \leq 1 \quad 0 \leq x_{0911} \leq 1 \\
0 &\leq x_{0106} \leq 1 \quad 0 \leq x_{0309} \leq 1 \quad 0 \leq x_{0516} \leq 1 \quad 0 \leq x_{0912} \leq 1
\end{aligned}$$

```

0 <= x0107 <= 1  0 <= x0310 <= 1  0 <= x0607 <= 1  0 <= x0913 <= 1
0 <= x0108 <= 1  0 <= x0311 <= 1  0 <= x0608 <= 1  0 <= x0914 <= 1
0 <= x0109 <= 1  0 <= x0312 <= 1  0 <= x0609 <= 1  0 <= x0915 <= 1
0 <= x0110 <= 1  0 <= x0313 <= 1  0 <= x0610 <= 1  0 <= x0916 <= 1
0 <= x0111 <= 1  0 <= x0314 <= 1  0 <= x0611 <= 1  0 <= x1011 <= 1
0 <= x0112 <= 1  0 <= x0315 <= 1  0 <= x0612 <= 1  0 <= x1012 <= 1
0 <= x0113 <= 1  0 <= x0316 <= 1  0 <= x0613 <= 1  0 <= x1013 <= 1
0 <= x0114 <= 1  0 <= x0405 <= 1  0 <= x0614 <= 1  0 <= x1014 <= 1
0 <= x0115 <= 1  0 <= x0406 <= 1  0 <= x0615 <= 1  0 <= x1015 <= 1
0 <= x0116 <= 1  0 <= x0407 <= 1  0 <= x0616 <= 1  0 <= x1016 <= 1
0 <= x0203 <= 1  0 <= x0408 <= 1  0 <= x0708 <= 1  0 <= x1112 <= 1
0 <= x0204 <= 1  0 <= x0409 <= 1  0 <= x0709 <= 1  0 <= x1113 <= 1
0 <= x0205 <= 1  0 <= x0410 <= 1  0 <= x0710 <= 1  0 <= x1114 <= 1
0 <= x0206 <= 1  0 <= x0411 <= 1  0 <= x0711 <= 1  0 <= x1115 <= 1
0 <= x0207 <= 1  0 <= x0412 <= 1  0 <= x0712 <= 1  0 <= x1116 <= 1
0 <= x0208 <= 1  0 <= x0413 <= 1  0 <= x0713 <= 1  0 <= x1213 <= 1
0 <= x0209 <= 1  0 <= x0414 <= 1  0 <= x0714 <= 1  0 <= x1214 <= 1
0 <= x0210 <= 1  0 <= x0415 <= 1  0 <= x0715 <= 1  0 <= x1215 <= 1
0 <= x0211 <= 1  0 <= x0416 <= 1  0 <= x0716 <= 1  0 <= x1216 <= 1
0 <= x0212 <= 1  0 <= x0506 <= 1  0 <= x0809 <= 1  0 <= x1314 <= 1
0 <= x0213 <= 1  0 <= x0507 <= 1  0 <= x0810 <= 1  0 <= x1315 <= 1
0 <= x0214 <= 1  0 <= x0508 <= 1  0 <= x0811 <= 1  0 <= x1316 <= 1
0 <= x0215 <= 1  0 <= x0509 <= 1  0 <= x0812 <= 1  0 <= x1415 <= 1
0 <= x0216 <= 1  0 <= x0510 <= 1  0 <= x0813 <= 1  0 <= x1416 <= 1
0 <= x0304 <= 1  0 <= x0511 <= 1  0 <= x0814 <= 1  0 <= x1516 <= 1
End

```

In Figure C.10 we show the solution of this initial LP. The solution is integer but does not correspond to a tour. There are four subtours, namely,  $S_1 = \{2, 3, 4\}$ ,  $S_2 = \{1, 8, 16\}$ ,  $S_3 = \{12, 13, 14\}$  and  $S_4 = \{5, 15, 6, 7, 10, 9, 11\}$ . The subtour elimination constraints (SEC) are shown in the right part of the figure. Adding these constraints to the LP and reoptimizing we get the solution shown in Figure C.11. Here again we have an integer solution that contains two subtours, namely,  $S_1 = \{1, 8, 4, 2, 3, 16\}$  and  $S_2 = \{5, 15, 14, 13, 12, 6, 7, 10, 9, 11\}$ . Only one of these is needed – since the other is implied – and the one we add is shown in the right part of the figure. Adding this constraint and reoptimizing we get the solution of Figure C.1. The dotted lines correspond to values of 0.5 while the solid ones to values of 1. Here we have one subtour, i.e.,  $S_1 = \{1, 8, 4, 2, 3\}$  and two combs. The first comb is defined by the handle  $H_1 = \{6, 15, 14\}$  and the teeth  $T_1^1 = \{6, 7\}$ ,  $T_2^1 = \{15, 5\}$  and  $T_3^1 = \{14, 13\}$ . The second comb is defined by the handle  $H_2 = \{7, 12, 13, 16, 10\}$  and the teeth  $T_1^2 = \{7, 6\}$ ,  $T_2^2 = \{13, 14\}$  and  $T_3^2 = \{10, 9\}$ . The corresponding inequalities are shown in the right part of the figure. Adding these inequalities to the LP and reoptimizing



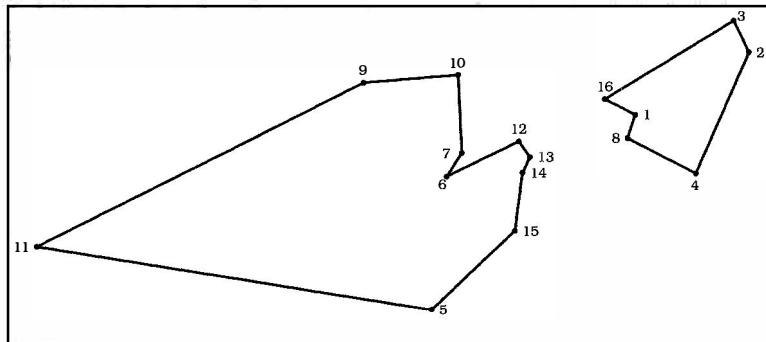
Optimal value: 6113.00

Violated inequalities:

SEC:

$$\begin{aligned} &x_{0203} + x_{0204} + x_{0304} \leq 2 \\ &x_{0108} + x_{0116} + x_{0816} \leq 2 \\ &x_{1213} + x_{1214} + x_{1314} \leq 2 \\ &x_{0515} + x_{0506} + x_{0507} + x_{0509} + x_{0510} + x_{0511} \\ &+ x_{0607} + x_{0610} + x_{0609} + x_{0611} + x_{0615} + x_{0710} \\ &+ x_{0709} + x_{0711} + x_{0715} + x_{1011} + x_{1015} + x_{0910} \\ &+ x_{0911} + x_{0915} + x_{1115} \leq 6 \end{aligned}$$

**Fig. C.10.** Optimal solution of the initial LP for Ulysses' problem



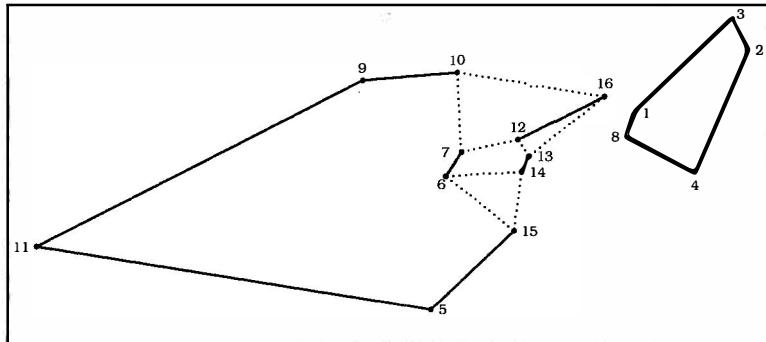
Optimal value: 6228.00

Violated inequalities:

SEC:

$$\begin{aligned} &x_{0108} + x_{0104} + x_{0102} + x_{0103} + x_{0116} + x_{0816} \\ &+ x_{0408} + x_{0416} + x_{0203} + x_{0216} + x_{0208} + x_{0204} \\ &+ x_{0316} + x_{0308} + x_{0304} \leq 5 \end{aligned}$$

**Fig. C.11.** Optimal solution of the second LP for Ulysses' problem



Optimal value: 6813.50

Violated inequalities:

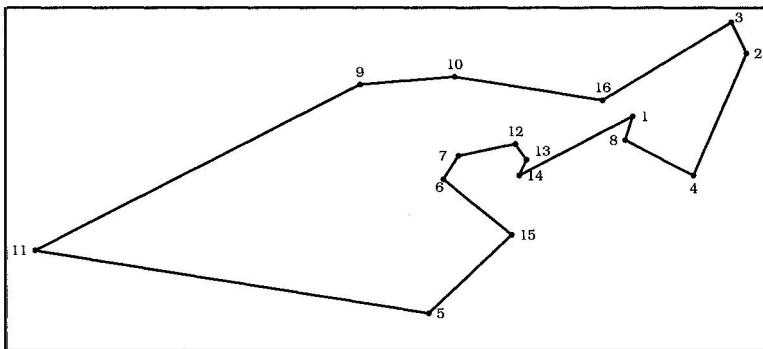
SEC:

$$\begin{aligned} &x_{0108} + x_{0104} + x_{0102} + x_{0103} + x_{0408} + x_{0203} \\ &+ x_{0208} + x_{0204} + x_{0308} + x_{0304} \leq 4 \end{aligned}$$

Comb:

$$\begin{aligned} &x_{0614} + x_{0615} + x_{1415} + x_{0607} + x_{1314} + x_{0515} \leq 4 \\ &x_{0710} + x_{0712} + x_{0713} + x_{0716} + x_{1012} + x_{1013} \\ &+ x_{1016} + x_{1213} + x_{1216} + x_{1316} + x_{0910} + x_{0607} \\ &+ x_{1314} \leq 6 \end{aligned}$$

**Fig. C.12.** Optimal solution of the third LP for Ulysses' problem



**Fig. C.13.** Optimal tour for Ulysses' problem of length 6,859 km

we get the solution of Figure C.13 which is a tour and thus the optimal solution. The length of the optimal tour is 6,850 km. It happens in this problem that one can solve it to optimality without using the comb inequalities. To verify this, just add only the SEC constraint in the last iteration. Of course, this is not always the case, as we will see in the problem with the 48 USA cities.

**(C)** Devise a method to prove or disprove the uniqueness of the optimal tour you found in part (B). What is the second best tour for Ulysses?

To decide whether the optimal tour found in part (B) is unique we proceed as follows. We set the length of each edge in the optimal tour to a big number, say 10,000, one at a time, and solve the resulting TSP. In Table C.5 we give the selected edge, the optimal tour and its length for each of the 16 problems. It follows that the tour found in part (B) is the unique optimum and that the second best tour is obtained by excluding edge (12, 13) or edge (1, 14) and has a length of 6,865km. Of course, at this point we cannot claim that the second best tour is unique as well. Rather we would have to iterate the whole procedure for this tour with the additional constraint that the objective function is greater than or equal 6,865. We did, of course, *not* carry out these calculations by the “visual inspection” method described under (B), rather we used a computer program developed by M. Padberg and G. Rinaldi (Rome) to solve the sixteen problems.

**(D)** Write a problem generator in a computer language of your choice that automatically sets up the linear program for TSPs that you want to solve in the format required by your LP solver.

The following program is written in C++ and prepares the lp-format input for CPLEX.

```
#include <stdio.h>

void
main()
{
    FILE *fp, *fq;
    fp=fopen("dist.dat","r");
    fq=fopen("tsp.lp","w");
    ...
```

**Table C.5.** Finding the second best tour for Ulysses' problem

Edge	Optimal tour	Length
(13,14)	14 15 5 11 9 10 6 7 12 13 16 3 2 4 8 1	6911
(12,13)	13 14 12 7 6 15 5 11 9 10 16 3 2 4 8 1	6865
(7,12)	8 4 2 3 16 12 13 14 6 7 10 9 11 5 15 1	6870
(6,7)	8 4 2 3 16 12 7 10 9 11 5 15 6 14 13 1	7001
(6,15)	8 4 2 3 16 12 13 14 6 7 10 9 11 5 15 1	6870
(5,15)	8 4 2 3 16 10 9 11 5 6 7 12 13 14 15 1	7005
(5,11)	8 4 2 3 16 12 7 10 9 11 6 5 15 14 13 1	7224
(9,11)	12 13 14 15 5 11 6 7 9 10 16 3 2 4 8 1	7260
(9,10)	8 4 2 3 16 10 7 6 9 11 5 15 14 13 12 1	7041
(10,16)	8 4 2 3 16 12 13 14 6 7 10 9 11 5 15 1	6870
(3,16)	16 13 14 12 7 6 15 5 11 9 10 3 2 4 8 1	6909
(2,3)	8 15 5 11 9 10 7 6 14 13 12 16 3 4 2 1	7502
(2,4)	8 4 3 2 16 10 9 11 5 15 6 7 12 13 14 1	6969
(4,8)	4 2 3 16 12 13 14 6 7 10 9 11 5 15 8 1	6890
(1,8)	16 12 13 14 6 7 10 9 11 5 15 8 4 2 3 1	6941
(1,14)	13 14 12 7 6 15 5 11 9 10 16 3 2 4 8 1	6865

```

int nof_nodes = 0, dist = 0;

fscanf(fp,"%d",&nof_nodes);

fprintf(fq,"minimize\n");
int cnt = 0;
for (int i=1; i < nof_nodes; i++) {
for (int j=i+1; j <= nof_nodes; j++) {
cnt++;
fscanf(fp,"%d",&dist);
fprintf(fq,"%d x%02d%02d",dist,i,j);
if (cnt % 7 == 0) fprintf(fq,"\n");
if (i != nof_nodes-1 || j != nof_nodes)
fprintf(fq," + ");
}
}
fclose(fp);
fprintf(fq,"\nsubject to\n");
for (i=1; i <= nof_nodes; i++) {
cnt = 0;
for (int j=1; j <= nof_nodes; j++) {
if (i == j) continue;

```

```

        cnt++;
        if (i < j)
fprintf(fq, "x%02d%02d", i, j);
        else
fprintf(fq, "x%02d%02d", j, i);
        if (cnt % 7 == 0) fprintf(fq, "\n");
        if (i != nof_nodes || j != nof_nodes)
fprintf(fq, " + ");
    }
fprintf(fq, " = 2\n");
}

fprintf(fq, "Bounds\n");
for (i=1; i < nof_nodes; i++)
for (int j=i+1; j <= nof_nodes; j++)
fprintf(fq, "0 <= x%02d%02d <= 1\n", i, j);

fprintf(fq, "End\n");
}

```

The program reads the distance matrix from the file `dist.dat` and produces the file `tsp.lp` that can be read in CPLEX with the command `read`. It is assumed that the file `dist.dat` contains integer numbers  $d(i,j)$  given in the following order

$$d(1,2), d(1,3), \dots, d(1,n), d(2,3), \dots, d(2,n), \dots, d(n-1,n).$$

**(E)** Given a node set  $S \subseteq V$  or a comb  $C = (H, T_1, \dots, T_k)$  in a graph  $G$  write a routine that automatically sets up the linear inequality (C.3) or (C.7) in the format required by your LP solver.

The following program is written in C++ and produces SEC and/or comb inequalities in CPLEX lp format.

```

#include <stdio.h>
#include <string.h>

void write_x_of_S(FILE *, int, int *);

void
write_x_of_S(FILE *fq, int nof_nodes, int *node)
{
    int cnt = 0, j, k;
    for (j=0; j < nof_nodes-1; j++) {
for (k=j+1; k < nof_nodes; k++) {
    cnt++;
    if (node[k] < node[j])
fprintf(fq, "x%02d%02d", node[k], node[j]);
    else
fprintf(fq, "x%02d%02d", node[j], node[k]);
    if (j != nof_nodes-2 || k != nof_nodes-1) {

```

```

fprintf(fq, " + ");
if (cnt%7 == 0) fprintf(fq, "\n");
}
}
}
}

void
main()
{
    FILE *fp, *fq;
    char filein[10];
    int nof_secs = 0;
    int nof_nodes = 0;
    printf("Give the input file name: ");
    scanf("%s",&filein);
    fp=fopen(filein,"r");
    strcat(filein,".lp");
    fq=fopen(filein,"w");

    fscanf(fp,"%d",&nof_secs);
    for (int i=1; i<=nof_secs; i++) {
        fscanf(fp,"%d",&nof_nodes);
        int *node= new int [nof_nodes];

        for (int j=0; j < nof_nodes; j++)
        fscanf(fp,"%d",&node[j]);

        write_x_of_S(fq,nof_nodes,node);
        fprintf(fq, " <= %d \n",nof_nodes-1);

        delete [] node;
    }

    int nof_combs = 0;
    fscanf(fp,"%d",&nof_combs);
    for (i=1; i<=nof_combs; i++) {
        fscanf(fp,"%d",&nof_nodes);
        int *node= new int [nof_nodes];

        for (int j=0; j < nof_nodes; j++)
        fscanf(fp,"%d",&node[j]);

        write_x_of_S(fq,nof_nodes,node);
        delete [] node;
        int rhs = nof_nodes;
        int nof_teeth = 0;
        fscanf(fp,"%d",&nof_teeth);
        for (j=1; j <= nof_teeth; j++) {
            fprintf(fq, " + \n",nof_nodes-1);

```

```

fscanf(fp,"%d",&nof_nodes);
int *node= new int[nof_nodes];

for (int j=0; j < nof_nodes; j++)
    fscanf(fp,"%d",&node[j]);

write_x_of_S(fq,nof_nodes,node);
rhs = rhs + nof_nodes - 2;
delete [] node;
}
rhs = rhs + nof_teeth/2;

fprintf(fq," <= %d \n", (int) rhs);
}
fclose(fp);
fclose(fq);
}

```

The program asks the user to give the input file name and it writes the inequalities in a file with the same name and extension .lp. The input file is assumed to have a particular format which we show in the following example. Suppose we want to create the inequalities for the third LP of Ulysses' problem; see Figure C.1. The input file looks as follows

```

1      <--- number of subtours
5 1 2 3 4 8      <--- number of nodes and the nodes of the subtour
2      <--- number of combs
3 6 14 15      <--- number of nodes and the nodes of the handle (1st comb)
3      <--- number of teeth
2 6 7      <--- number of nodes of the 1st tooth and the nodes of the 1st tooth
2 14 13      <--- number of nodes of the 2nd tooth and the nodes of the 2nd tooth
2 5 15      <--- number of nodes of the 3rd tooth and the nodes of the 3rd tooth
5 10 7 12 13 16 <--- number of nodes and the nodes of the handle (2nd comb)
3      <--- number of teeth in the 2nd comb
2 10 9      <--- number of nodes of the 1st tooth and the nodes of the 1st tooth
2 7 6      <--- number of nodes of the 2nd tooth and the nodes of the 2nd tooth
2 13 14      <--- number of nodes of the 3rd tooth and the nodes of the 3rd tooth

```

**(F)** Now solve the 48-city problem like you did above starting the calculations like in part (A). Plot each linear programming solution on a map of the United States of America, like the one of Figure C.7 and iterate. Put all violated subtour elimination and comb constraints that you find in the course of your calculations into a memory "pool", i.e. store them in some separate computer file.

Table C.4 contains the problem data for the 48 city problem, i.e., the "virtual" coordinates of the 48 state capitals of the continental U.S.A. From these coordinates the intercity distances for cities  $u$  and  $v$  are calculated by the following computer instructions in pseudo-code using double precision (64 bit) words.

---

```

XDIST =((xu - xv) * (xu - xv) + (yu - yv) * (yu - yv))/10
XDIST =SQRT(XDIST)
MDIST =XDIST
if MDIST < XDIST then MDIST=MDIST+1

```

---

In the following pages we present the solutions of the LP for each iteration of the cutting plane algorithm. The constraints generated during the execution of the algorithm are the following.

### **Iteration 0**

We identify the following six subtours:  $S_1 = \{6, 37, 19, 27, 17, 43, 30\}$ ,  $S_2 = \{3, 34, 41, 16, 22\}$ ,  $S_3 = \{7, 28, 36\}$ ,  $S_4 = \{13, 25, 14\}$ ,  $S_5 = \{42, 24, 45, 35, 4, 26, 10\}$  and  $S_6 = \{13, 25, 14, 23\}$ , and one comb with handle  $H^1 = \{20, 12, 33\}$  and three teeth  $T_1^1 = \{20, 47\}$ ,  $T_2^1 = \{12, 11\}$  and  $T_3^1 = \{33, 46\}$ .

### **Iteration 1**

We identify four subtours  $S_7 = \{28, 6, 37, 19, 27, 17, 43, 30\}$ ,  $S_8 = \{7, 36, 18\}$ ,  $S_9 = \{1, 8, 38, 31, 44, 46, 33, 20, 47, 21, 13, 25, 14, 34, 41, 16, 22, 3, 23, 11, 12, 15, 40, 9\}$  and  $S_{10} = \{2, 29, 5, 48, 39, 32, 24, 10, 45, 35, 4, 26, 42\}$ .

### **Iteration 2**

We identify seven subtours:  $S_{11} = \{28, 6, 37, 19, 27, 17, 43, 30, 36\}$ ,  $S_{12} = \{28, 6, 37, 19, 27, 17, 43, 30, 36, 7\}$ ,  $S_{13} = \{1, 8, 9, 38\}$ ,  $S_{14} = \{46, 33, 20, 47, 13, 25, 14, 34, 41, 16, 22, 3, 23, 11, 12, 15, 40\}$ ,  $S_{15} = \{21, 32, 39\}$ ,  $S_{16} = \{2, 29, 5, 48, 42\}$  and  $S_{17} = \{24, 10, 45, 35, 4, 26\}$ , and one comb with handle  $H^2 = \{7, 28, 36\}$  and three teeth  $T_1^2 = \{28, 6\}$ ,  $T_2^2 = \{36, 30\}$  and  $T_3^2 = \{7, 18\}$ .

### **Iteration 3**

We identify the following three subtours:  $S_{18} = \{7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 18\}$ ,  $S_{19} = \{1, 8, 38, 31, 44, 46, 33, 20, 47, 21, 32, 39, 25, 13, 14, 34, 41, 16, 22, 3, 23, 11, 12, 15, 40, 9\}$  and  $S_{20} = \{2, 29, 5, 48, 42, 24, 10, 45, 35, 4, 26\}$ .

### **Iteration 4**

We identify one subtour  $S_{21} = \{3, 22, 16\}$  and two combs, one with handle  $H^3 = \{13, 47, 21\}$  and three teeth  $T_1^3 = \{21, 32\}$ ,  $T_2^3 = \{47, 20\}$  and  $T_3^3 = \{13, 25\}$ , and the other with handle  $H^4 = \{29, 34, 41, 16, 22, 3, 23, 14\}$  and three teeth  $T_1^4 = \{29, 2\}$ ,  $T_2^4 = \{14, 25\}$  and  $T_3^4 = \{23, 11\}$ .

### **Iteration 5**

We identify two combs, one with handle  $H^5 = \{39, 48, 25\}$  and three teeth  $T_1^5 = \{48, 5\}$ ,  $T_2^5 = \{25, 13, 14\}$  and  $T_3^5 = \{39, 32\}$ , and the other with handle  $H^6 = \{21, 47, 13\}$  and three teeth  $T_1^6 = \{47, 20\}$ ,  $T_2^6 = \{21, 32\}$  and  $T_3^6 = \{13, 14, 25\}$ .

### **Iteration 6**

We identify one subtour  $S_{22} = \{12, 15, 40, 9, 1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 20, 47, 11, 12, 15, 40, 9\}$  and one comb with handle  $H^7 = \{13, 21, 47, 20, 12, 11\}$  and five teeth  $T_1^7 = \{20, 33\}$ ,  $T_2^7 = \{12, 15\}$ ,  $T_3^7 = \{11, 23\}$ ,  $T_4^7 = \{13, 14, 25\}$  and  $T_5^7 = \{21, 32\}$ .

### **Iteration 7**

We identify one subtour  $S_{23} = \{1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 20, 47, 11, 12, 15, 40, 9\}$ .

### **Iteration 8**

We identify one comb with handle  $H^8 = \{13, 21, 47\}$  and three teeth  $T_1^8 = \{47, 20\}$ ,  $T_2^8 = \{13, 14, 25\}$  and  $T_3^8 = \{21, 32, 39\}$ .

**Iteration 9**

We identify one comb with handle  $H^9 = \{39, 48, 25\}$  and three teeth  $T_1^9 = \{25, 14\}$ ,  $T_2^9 = \{48, 5\}$  and  $T_3^9 = \{32, 39\}$ .

**(G)** Having solved the problem the “hard” way – you should be able to do so, without branching, on the first try by solving about 15-25 linear programs – do it again, but in the following way: at every iteration take from the memory pool that you created in the first run all constraints that are violated by the current LP optimum (but not the ones that are not violated!) and iterate. How many iterations do you need now to solve the problem? Why?

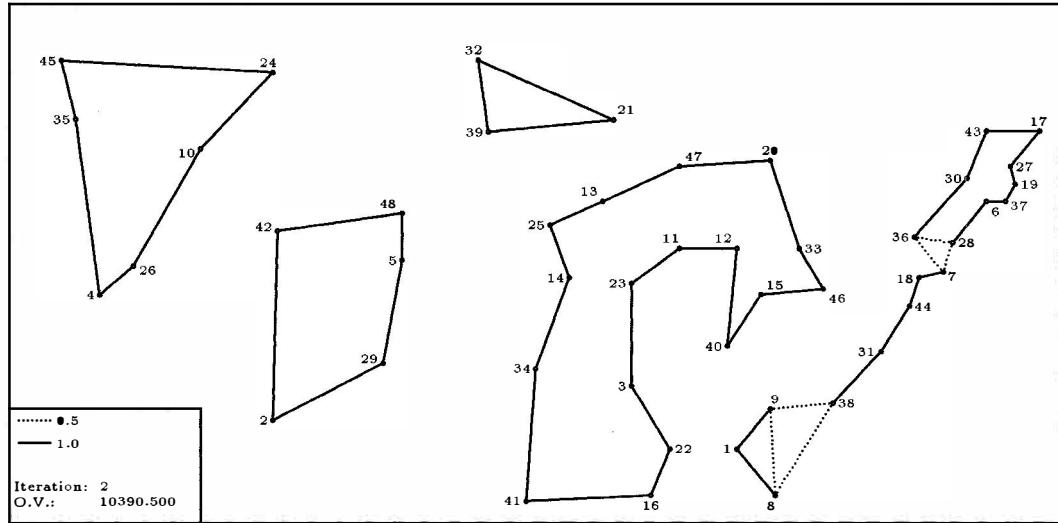
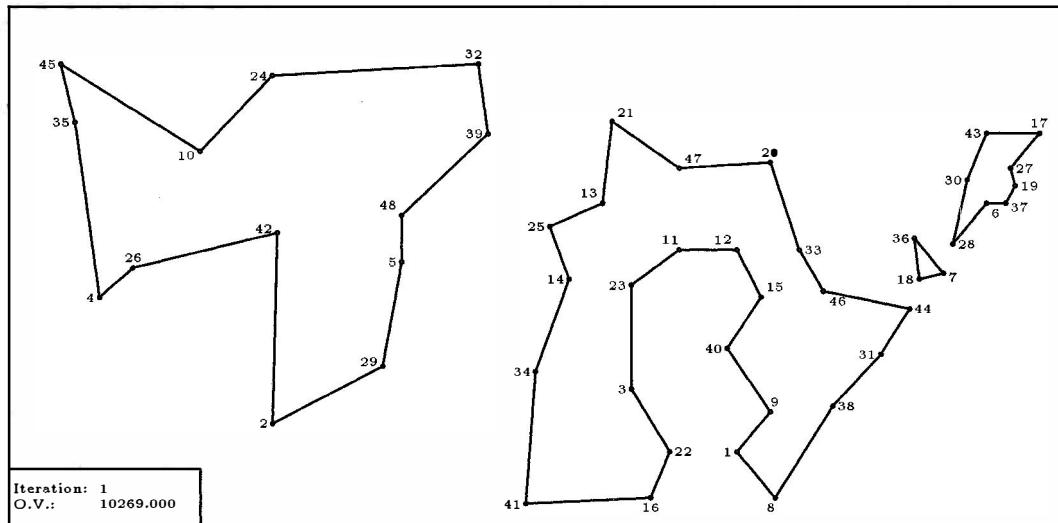
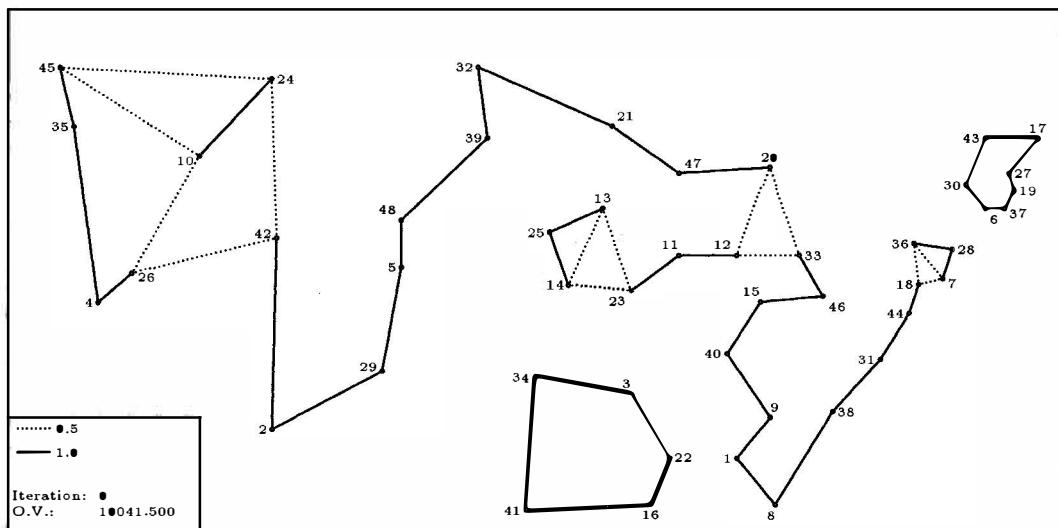
Iteration	O.V.	Violated Inequalities
0	10041.500	$S_1, \dots, S_6, H_1, S_{10}, S_{12}, S_{17}, S_{18}, S_{20}, H_5, H_6, H_8$
1	10542.000	$S_7, S_{13}, H_7$
2	10603.500	$S_{11}, S_{21}, H_4, S_{22}, S_{23}$
3	10627.000	$H_9$
4	10628.000	

Had we known all the inequalities we added beforehand we would have spent less time to find the optimal solution as we show now. Given the above “pool” of inequalities what we need to do is to be able to check which ones are violated by the given solution to the current LP and add them. This can be done by hand, or by a small routine. We followed a different way, using CPLEX. We introduce slack variables  $s_{49}$  to  $s_{80}$  which are free in sign (this is done in CPLEX by stating e.g.  $s_{80}$  Free in the Bounds section of the LP) for constraints  $c_{49}$  to  $c_{80}$  (these are the generated constraints; constraints  $c_1$  to  $c_{48}$  are the degree constraints). Then we append the equations that define the current solution. To avoid adding all the zero values we include the degree constraints. Now we solve this auxiliary LP with a zero objective function. Clearly, the constraints with negative slack variables are violated by the current solution, and therefore, they are added to the main LP. The following table shows how this procedure progresses (we represent a comb by its handle).

So in four iterations we reach the optimal solution. The reduction in the number of iterations achieved this way is substantial and indicative of the acceleration of branch-and-cut algorithms that is possible when the polyhedral separation problem is solved by the “perfect” identification of a large number of (most) violated constraints; see also our discussion of the problem of finding the “right cut” on pages 319-320 of the text.

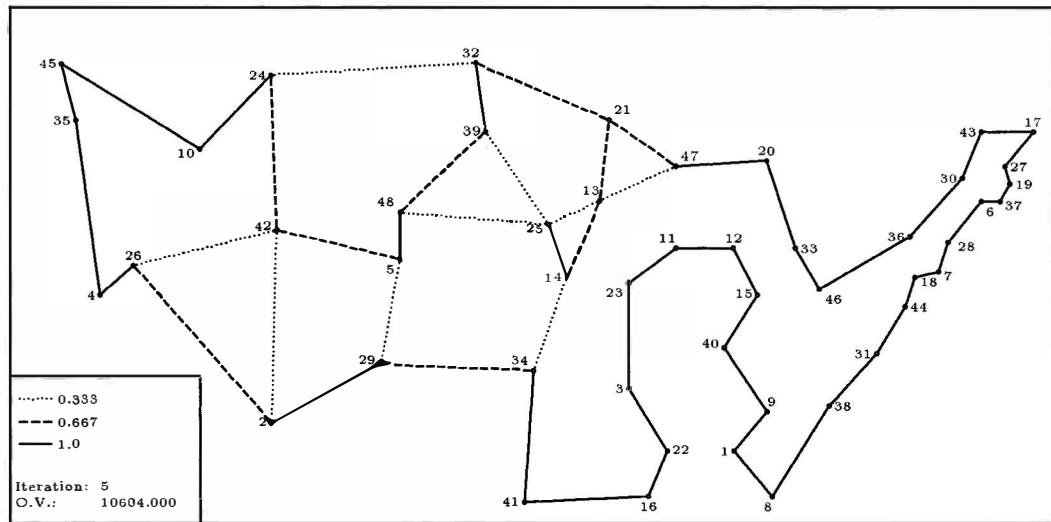
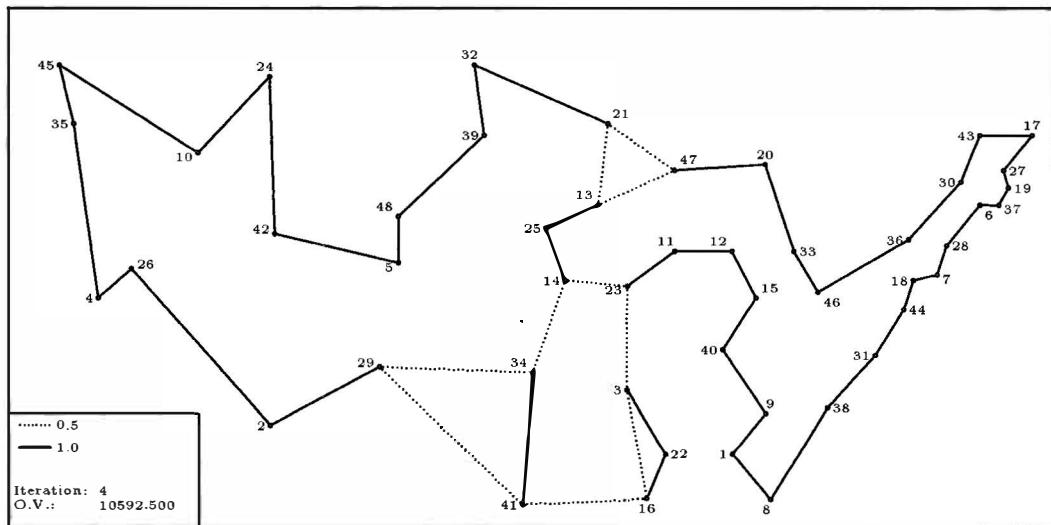
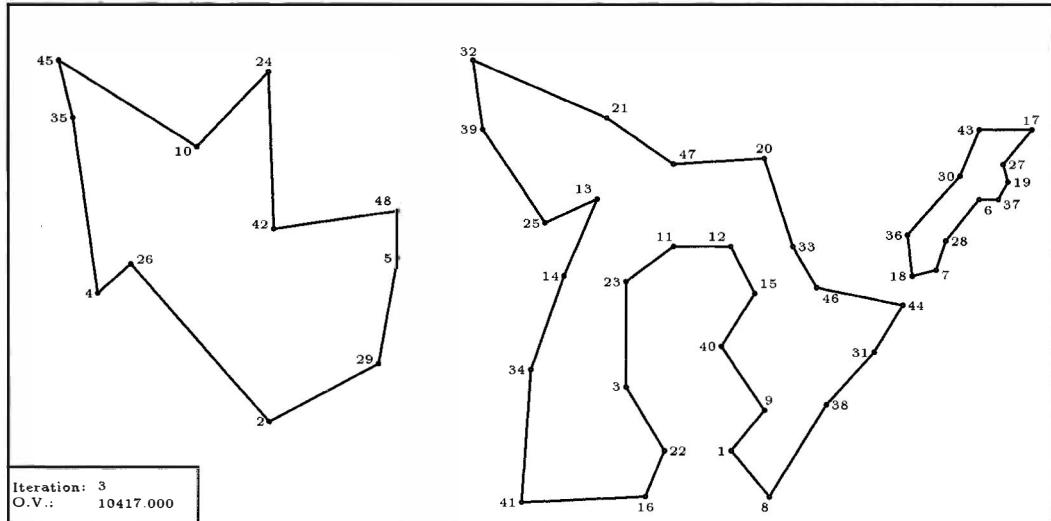
**(H)** Now suppose that comb constraints are unknown to you and that you know only the subtour elimination constraints (C.3). Solve the problem again using only constraints (C.3), i.e. solve the corresponding problem ( $TSP_{LP}$ ). What is its optimal value? Using branch-and-bound in a suitably modified form (zero-one solutions that correspond to subtours must be cut off by constraints of the form (C.3)!) find an optimal solution to the TSP. State your rules for the selection of branching variables and the selection of the “next” problem from the problem stack clearly. How many nodes does your search tree have? Plot the search tree like we did in Figure 10.1.

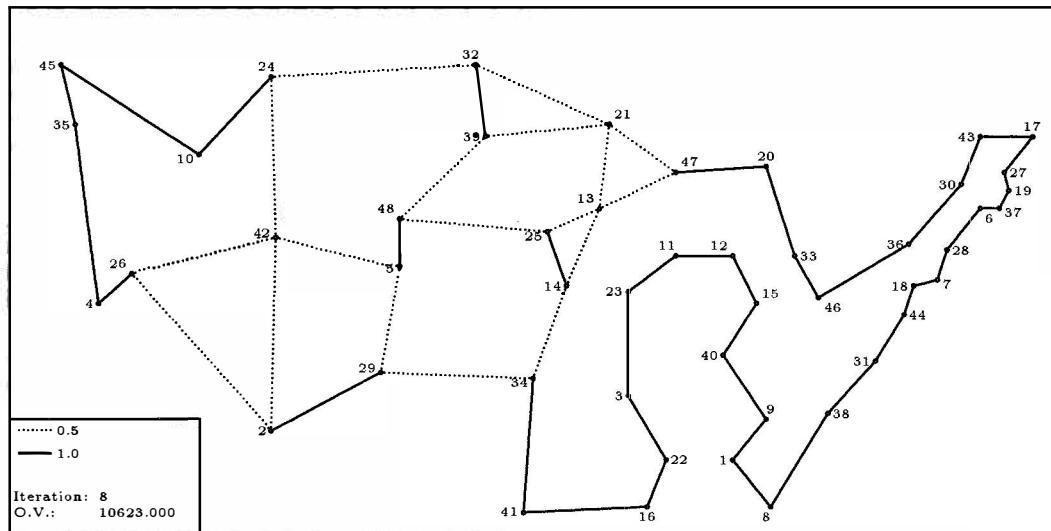
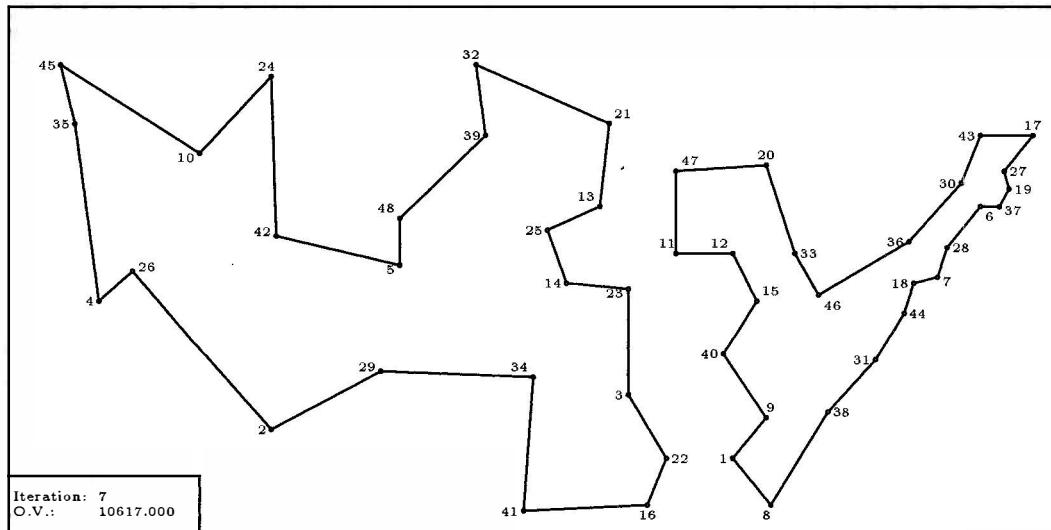
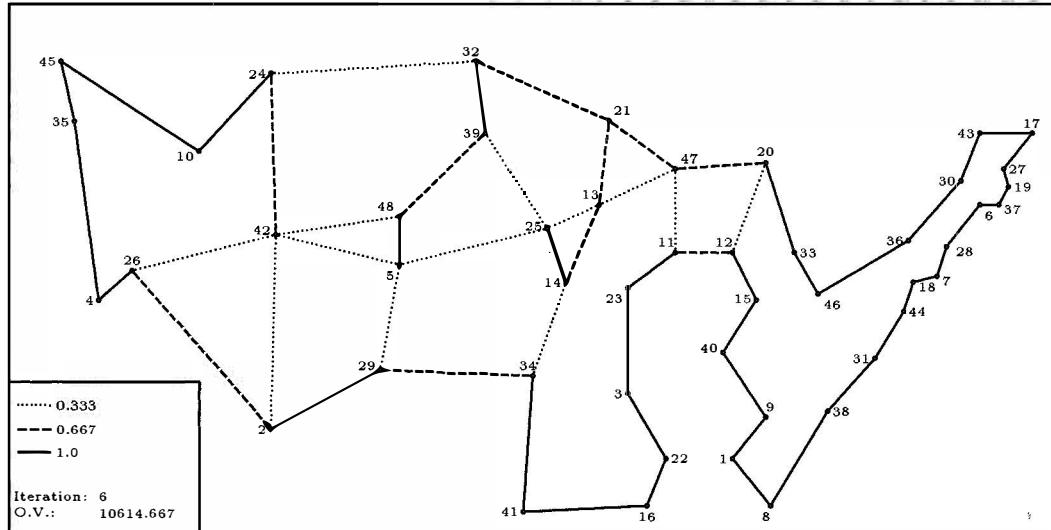
Assuming that the SEC are the only constraints known to us we follow the same procedure as in part (F). In Table C.6 we show for Node 1 which is the root node of the branch-and-cut tree, see Figure C.15 the iterations taken to solve the problem and the subtours identified in each iteration. From this table we see that the optimization over the subtour polytope gives an

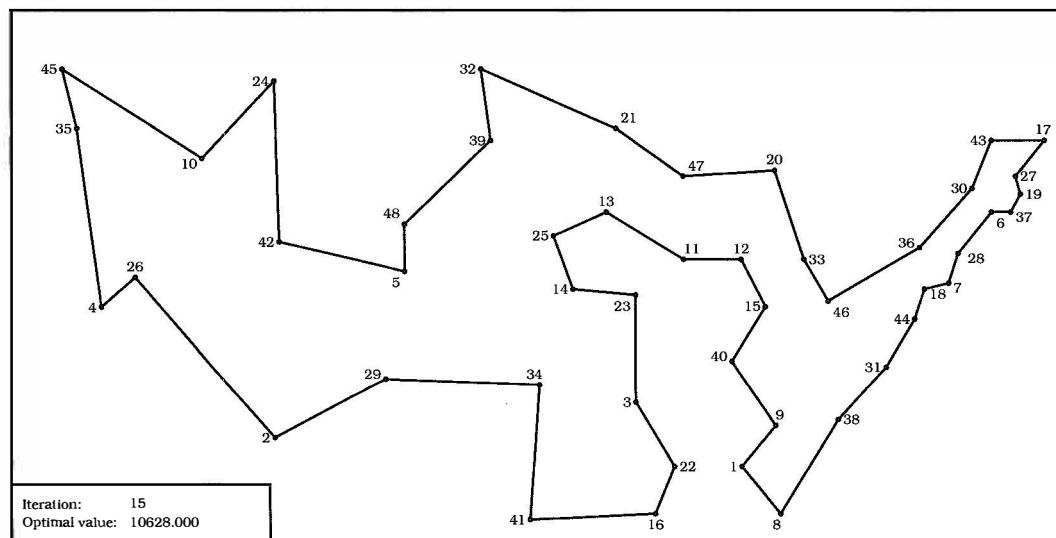
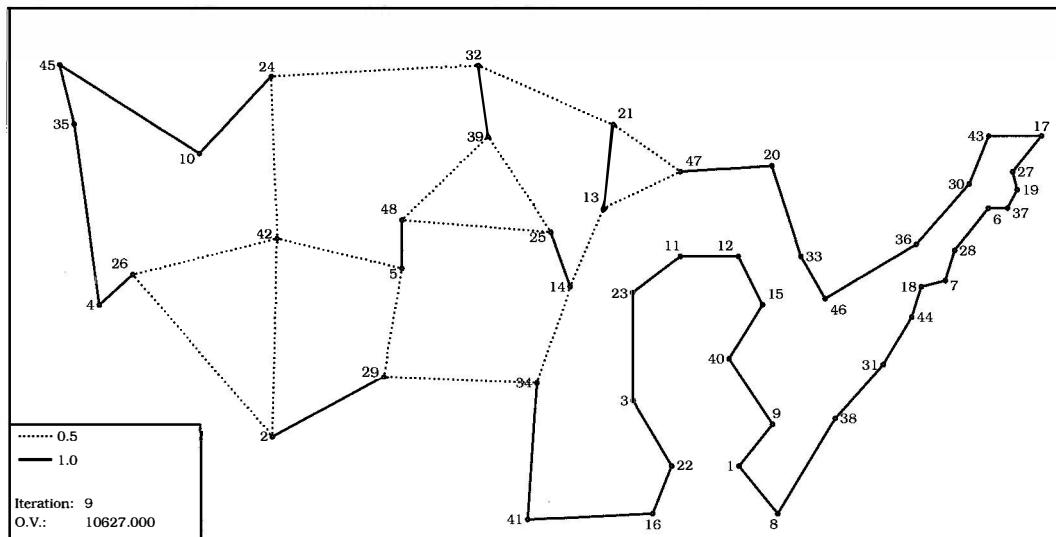


### C.1. SOLUTIONS TO ULYSSES' PROBLEM

425



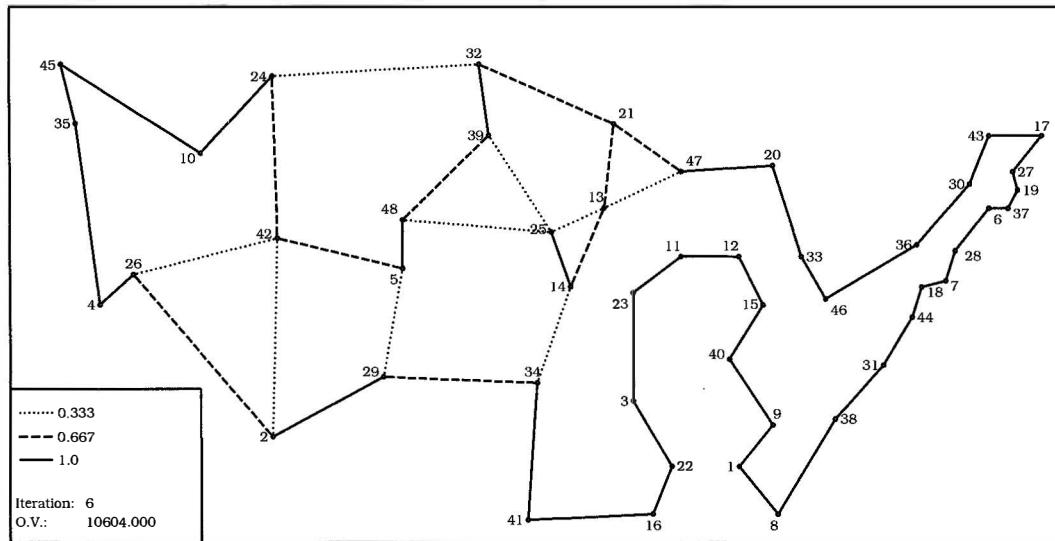




**Table C.6.** SEC added in the branch-and-cut algorithm

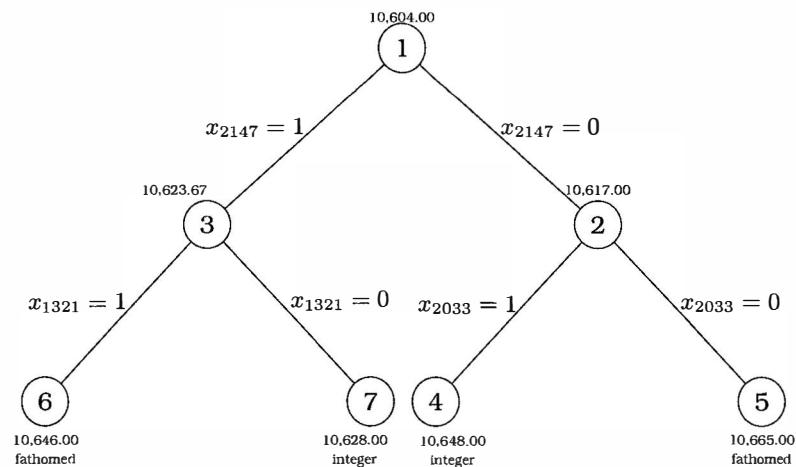
Node	Iter.	O.V.	Subtours added
1	0	10041.50	6 37 19 27 17 43 30 3 34 41 16 22 7 28 36 13 25 14 42 24 45 35 4 26 10 13 25 14 23
	1	10269.00	28 6 37 19 27 17 43 30
			7 36 18 1 8 38 31 44 46 33 20 47 21 13 25 14 34 41 16 22 3 23 11 12 15 40 9 2 29 5 48 39 32 24 10 45 35 4 26 42
	2	10390.00	32 39 21 24 10 26 4 35 45 2 42 48 5 29 20 47 13 25 14 34 41 16 22 3 23 11 12 20 47 13 25 14 34 41 16 22 3 23 11 12 33 36 30 43 17 27 19 37 6 28 36 30 43 17 27 19 37 6 28 7
	3	10417.00	7 28 6 37 19 27 17 43 30 36 18 1 8 38 31 44 46 33 20 47 21 32 39 25 13 14 34 41 16 22 3 23 11 12 15 40 9 2 29 5 48 42 24 10 45 35 4 26
	4	10571.50	1 8 9 1 8 9 38 21 32 39 48 5 42 24 10 45 35 4 26 2 29 34 41 16 22 3 40 15 12 11 23 14 25 13 21 32 39 48 5 42 24 10 45 35 4 26 2 29 34 41 16 22 3 40 15 12 11 23 14 25 13 47
	5	10592.00	3 22 16
	6	10604.00	no more subtours
2	0	10612.00	4 26 2 29 34 41 16 22 3 23 14 25 13 21 32 39 48 5 42 24 10 45 35
	1	10623.67	no more subtours
3	0	10617.00	12 15 40 9 1 8 38 31 44 18 7 28 6 37 19 27 17 43 30 36 46 33 20
	1	10634.00	no more subtours

optimal value of 10604.00. The solution is shown in Figure C.14. From that point we have to resort to branching in order to proceed. We select to branch on the variable with highest value not equal to one, breaking ties arbitrarily. Figure C.15 shows the tree. Our branching strategy is to solve both child-LPs and proceed with the one of highest value. Thus in the case of the 48-city problem this branching scheme works reasonably well. But don't be misled by this particular instance. We include here the data set tka076 for which this proceeding produces a search-tree that is very, very long. In the case of tka076 we have 76 cities, the optimal tour length equals 108,159 and the relaxation over the subtour elimination constraints gives an LP value of 105,120; see Figures C.17 and C.18, where solid lines correspond to variables at value one, dashed ones



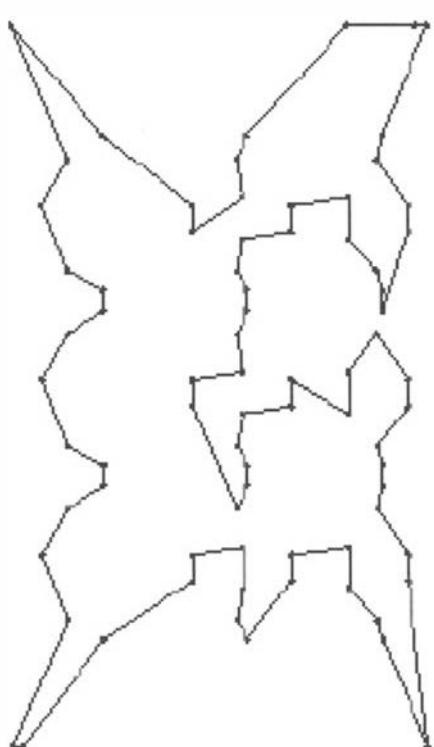
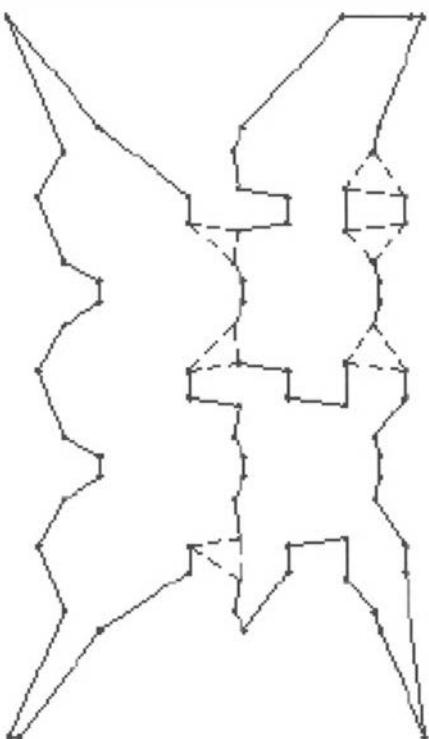
**Fig. C.14.** The optimal solution for  $TSP_{LP}$

to variables at value 1/2. The data shown in Figure C.16 are the  $(x_i, y_i)$ -coordinates of the 76 points where  $1 \leq i \leq 76$  and we start reading in the upper left corner. Distances are Euclidean, they are computed in double precision and then rounded to the nearest integer by adding 0.5 and truncating to an integer number.



**Fig. C.15.** Search tree for question (H)

76  
 02300 03600 03300 03100 05750 04700 05750 05400 07103 05608  
 07102 04493 06950 03600 07250 03100 08450 04700 08450 05400  
**10053 05610 10052 04492 10800 03600 10950 03100 11650 04700**  
 11650 05400 10800 06650 10950 07300 07250 07300 06950 06650  
 03300 07300 02300 06650 01600 05400 02300 08350 03300 07850  
 05750 09450 05750 10150 07103 10358 07102 09243 06950 08350  
 07250 07850 08450 09450 08450 10150 10053 10360 10052 09242  
 10800 08350 10950 07850 11650 09450 11650 10150 10800 11400  
 10950 12050 07250 12050 06950 11400 03300 12050 02300 11400  
 01600 10150 02300 13100 03300 12600 05750 14200 05750 14900  
 07103 15108 07102 13993 06950 13100 07250 12600 08450 14200  
 08450 14900 10053 15110 10052 13992 10800 13100 10950 12600  
 11650 14200 11650 14900 10800 16150 10950 16800 07250 16800  
 06950 16150 03300 16800 02300 16150 01600 14900 00800 19800  
 10000 19800 11900 19800 12200 19800 12200 00200 01100 00200  
 00800 00200

**Fig. C.16.** The data for the tka076 problem**Fig. C.17.** Optimal tour of length 108159 for problem tka076**Fig. C.18.** Optimal solution of length 105120 over the subtour polytope for problem tka076

# Bibliography

- Adler, I., N. Karmarkar, M. Resende and C. Veiga [1989] "Data structures and programming techniques for the implementation of Karmarkar's algorithm", *ORSA Journal on Computing* 1 84–106.
- Αλευράς, Δ. " Αριστοποίηση του συστήματος παραγωγής, διανομής και χρήσης ατμού στα ΕΛ.Δ.Α. ", Διπλωματική Εργασία Ε.Μ.Π., 1988, in Greek. English translation: D. Alevras "Optimization of the steam production, distribution and usage system in H.A.R.", Master's Thesis, National Technical University of Athens, Greece, 1988.
- Alevras, D. and M. Padberg [1992] "Operations management in a refinery: A case study of daily steam operations", Working paper SOR-92-22, New York University, New York.
- Alevras, D. and M.P Rijal [1995] "The convex hull of a linear congruence relation in zero-one variables", *ZOR-Mathematical Methods of Operations Research* 41, 1-23.
- Anonymous [1972] "A new algorithm for optimization", *Mathematical Programming* 3 124-128.
- Anstreicher, K. [1986] "A monotonic projective algorithm for fractional linear programming", *Algorithmica* 1 483-498.
- Antosiewicz, H.A. (ed) [1955] *Linear Programming*, Proceedings of the Second Symposium in Linear Programming, Volumes I and II, National Bureau of Standards, Washington.
- Arabayre, J.P., J. Fearnley, F. Steiger and W. Theather [1969] "The air crew scheduling problem: A survey", *Trans. Sci.* 3 140–163.
- Aronofsky, J.S. (ed) [1969] *Progress in Operations Research: Relationship between Operations Research and the Computer*, Wiley, New York.
- Balas, E. [1970] Handwritten lecture notes, unpublished.
- Balas, E. and W. Pulleyblank [1983] "The perfectly matchable polytope of a bipartite graph", *Networks* 13 495–516.
- Balinski, M.L. (ed) [1974] *Pivoting and Extensions: In Honor of A.W. Tucker*, Mathematical Programming Study 1, North-Holland, Amsterdam.
- Balinski, M.L. [1991] "Gaspard Monge: Pour la patrie, les sciences et la gloire", in Carasso *et al* (eds), "Applied Mathematics for Engineering Science", Cépaduès-Éditions, Toulouse 21–37.
- Balinski, M.L. and E. Hellerman (eds) [1976] *Computational Practice in Mathematical Programming*, Mathematical Programming Study 4, North-Holland, Amsterdam.
- Balinski, M.L. and K. Spielberg [1969] "Methods for integer programming: Algebraic, combinatorial and enumerative", in Aronofsky (ed) *Progress in Operations Research*, Wiley, New York, 195–192.
- Bareiss, E.H. [1968] "Sylvester's identity and multistep integer-preserving Gaussian elimination", *Mathematics of Computation* 22 565–578.
- Barnes, E.R. [1986] "A variation on Karmarkar's algorithm for solving linear programming problems", *Mathematical Programming* 36 174–182.
- Bartels, R.H. [1971] "A stabilization of the simplex method", *Numerische Mathematik* 16 414–434.
- Bartels, R.H. and G.H. Golub [1969] "The simplex method of linear programming using LU decomposition", *Communications of the Association for Computing Machinery* 12 266–268.
- Bayer, D.A. and J.C. Lagarias [1991] "Karmarkar's linear programming algorithm and Newton's method", *Mathematical Programming* 50 291-330.
- Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. [1990] *Linear Programming and Network Flows*, Wiley, New York.
- Beale, E.M.L. [1965] "Survey of integer programming", *Operational Research Quarterly* 16 219–228.
- Beale, E.M.L. [1954] "An alternative method for linear programming", *Proceedings of the Cambridge Philosophical Society* 50 513–523.
- Beale, E.M.L. [1955] "Cycling in the dual simplex algorithm", *Naval Research Logistics Quarterly* 2 269–275.
- Beale, E.M.L. [1968] *Mathematical Programming in Practice*, Pitman & Sons Ltd, London.

- Beckmann, P. [1971] *A History of  $\pi$  (Pi)*, The Golem Press, New York.
- Berge, C. [1972] "Balanced matrices", *Mathematical Programming* 2 19–31.
- Benichou, M., J.M. Gauthier, P. Girodet, G. Heutges, G. Ribière and O. Vincent [1971] "Experiments in mixed-integer linear programming", *Mathematical Programming* 1 76–94.
- Benichou, M., Gauthier, J.M., Hentges, G. and G. Ribière [1977] "The efficient solution of large-scale linear programming problems – Some algorithmic techniques and computational results" *Mathematical Programming* 13 280–322.
- Benoit [1924] "Sur une méthode de résolution des équations normales etc. (procédé du commandant Cholesky)", *Bull. géodésique* 2.
- Bixby, R.E. [1989] Personal communication.
- Bixby, R.E. [1990] "Implementing the simplex methods, Part I: Introduction, Part II: The initial basis", Working paper TR90-32 Mathematical Sciences, Rice U., Houston. Part II published in *ORSA Journal on Computing*, 4, 1992, 267–284.
- Bixby, R. and M. Saltzman [1994] "Recovering an optimal LP basis from an interior point solution", *Operations Research Letters* 15 69–178.
- Bland, R. [1977] "New finite pivot rules for the simplex method", *Mathematics of Operations Research* 2 103–107.
- Bland, R.G., D. Goldfarb and M.J. Todd [1981] "The ellipsoid method: a survey", *Operations Research* 29 1039–1091.
- Borgwardt, K.H. [1982] "Some distribution independent results about the asymptotic order of the average number of pivot steps in the simplex algorithm", *Mathematics of Operations Research* 7 441–462.
- Bouilloud, P.H. [1969] "Compute steam balance by LP", *Hydrocarbon Process.* 127–128.
- Boyd, S.C. and W.H. Cunningham [1991] "Small traveling salesman polytopes", *Mathematics of Operations Research* 16 259–271.
- Brooke, A., D. Kendrick and A. Meeraus [1988] *GAMS: A User's Guide*, The Scientific Press, Redwood City.
- Brown, G.W. [1949] "Some computation methods for linear systems involving inequalities", Abstract, *Econometrica* 17 162–163.
- Burger, E. [1956] "Über homogene lineare Ungleichungssysteme", *Zeitschrift für Angewandte Mathematik und Mechanik* 36 135–139.
- Cahn, A.S. [1948] "The warehouse problem", *Bull.Amer.Math.Soc.* 54 1073.
- Camerini, P.M., L. Fratta and F. Maffioli [1975] "On improving relaxation methods by modified gradient techniques", *Mathematical Programming Study* 3 26–34.
- Carathéodory, C. [1911] "Über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen", *Rendiconti del Circolo Mathematico di Palermo* 32 193–217.
- Carathéodory, C. [1935] *Variationsrechnung und partielle Differentialgleichungen erster Ordnung*, Teubner, Leipzig.
- Cassels, J.W.S. [1965] *An Introduction to Diophantine Approximation*, Cambridge University Press, Cambridge.
- Charnes, A. [1952] "Optimality and degeneracy in linear programming", *Econometrica* 20 160–170.
- Charnes, A. and W.W. Cooper [1960] *Management Models and Industrial Applications of Linear Programming*, Volumes I & II, Wiley, New York.
- Charnes, A., W.W. Cooper and A. Henderson [1957] *An Introduction to Linear Programming*, Wiley, New York.
- Charnes A., W.W. Cooper and R. Mellon [1952] "Blending aviation gasolines: A study in programming interdependent activities in an integrated oil company", *Econometrica* 20 135–159.
- Charnes A., W.W. Cooper and R. Mellon [1954] "A model for programming and sensitivity analysis in an integrated oil company", *Econometrica* 22 193–217.
- Charnes A., W.W. Cooper and R. Mellon [1955] "A model for optimizing production by reference to cost surrogates", *Econometrica* 23 307–323.

- Charnes, A., W.W. Cooper and M.H. Miller [1959] "Application of linear programming to financial budgeting and costing of funds", *Journal of Business* 32 20–46.
- Christof, T., M. Jünger and G. Reinelt [1991] "A complete description of the traveling salesman polytope on 8 nodes", *Operations Research Letters* 10 497–500.
- Chvátal, V. [1983] *Linear Programming*, Freeman Press, New York.
- Clark Jr., J.K. and N.E. Helmick [1980] "How to optimize the design of steam systems", *Chemical Engineering* 116–128.
- Cohen, K.J. and F.S. Hammer (eds) [1966] "Analytical Methods in Banking", Irwin, Homewood.
- Cornuéjols, G., J. Fonlupt and D. Naddef [1985] "The traveling salesman problem on a graph and some related polyhedra", *Mathematical Programming* 33 1–27.
- Cornuéjols, G. and B. Novick [1994] "Ideal 0,1 matrices", *Journal of Combinatorial Theory, Series B*, 60 145–157.
- Courant, R. and H. Robbins [1978] *What is Mathematics*, Oxford University Press, New York.
- Crowder, H. and J.M. Hattingh [1975] "Partially normalized pivot selection in linear programming", *Mathematical Programming Study Number 4* 12–25.
- Crowder, H. and M. Padberg [1980] "Solving large-scale symmetric traveling salesman problems to optimality", *Management Science* 26 393–410.
- Crowder, H., E.L. Johnson and M. Padberg [1983] "Solving large-scale zero-one linear programming problems", *Operations Research* 31 803–834.
- Curtis, A. and J. Reid [1972] "On the automatic scaling of matrices for Gaussian elimination", *J. Inst. Maths Applications* 10 118–124.
- Dakin, R.J. [1965] "A tree-search algorithm for mixed-integer programming", *Computer J.* 8 250–255.
- Dantzig, G.B. [1949] "Programming of interdependent activities II: Mathematical model", *Econometrica* 17 200–211.
- Dantzig, G.B. [1949] "Programming in a linear structure", *Econometrica* 17 73–74.
- Dantzig, G.B. [1951] "Maximization of a linear function of variables subject to linear inequalities", in Koopmans (ed) *Activity Analysis of Production and Allocation*, Wiley, New York.
- Dantzig, G.B. [1960] "Inductive proof of the simplex method", *IBM Journal of Research and Development* 4 505–506.
- Dantzig, G.B. [1963] *Linear Programming and Extensions*, Princeton U. Press, Princeton.
- Dantzig, G.B. [1982] "Reminiscences about the origins of linear programming", *Operations Research Letters* 1 43–48.
- Dantzig, G.B. [1988] "Impact of linear programming on computer development", *OR/MS Today* August 1988 12–17.
- Dantzig, G.B., D.R. Fulkerson and S.M. Johnson [1954] "Solution of a large-scale travelling salesman problem", *Operations Research* 2 393–410.
- Dantzig, G.B. and W. Orchard-Hays [1954] "The product form for the inverse in the simplex method", *Mathematical Tables and Other Aids to Computation* 8 64–67.
- Dantzig, G.B., Orden, A. and Ph. Wolfe [1955] "The generalized simplex method for minimizing a linear form under linear inequality restraints", *Pacific Journal of Mathematics* 5 183–195.
- Dantzig, G.B. and J.H. Ramser [1959] "The truck dispatching problem", *Management Science* 6 80–91.
- Dantzig, G.B. and R.M. Van Slyke [1967] "Generalized upper bounding techniques", *Journal of Computer and System Sciences* 1 213–226.
- Dantzig, G.B. and Ph. Wolfe [1960] "Decomposition principle for linear programs", *Operations Research* 8 101–111.
- Descartes, R. [1637] *La Géometrie*, republished as *The geometry of René Descartes*, D.S. Smith and M.L. Latham (transltrs.), Dover Publications, New York, 1954.
- Dietrich, B.L. and L.F. Escudero [1990] "Coefficient reduction for knapsack constraints in 0-1 programs with VUBs", *Operations Research Letters* 9 9–14.

- Dietrich, B.L. and L.F. Escudero [1993] "Efficient reformulation for 0-1 programs: Methods and results" *Discrete Applied Mathematics* 42 147–175.
- Dikin, I.I. [1967] "Iterative solution of problems of linear and quadratic programming", *Soviet Mathematics Doklady* 8 674–675.
- Dorfmann, R., P.A. Samuelson and R. Solow [1958] *Linear Programming and Economic Analysis*, McGraw-Hill, New York.
- Driebeek, N.J. [1966] "An algorithm for the solution of mixed integer programming problems", *Management Science* 12 576–587.
- Edmonds, J. [1965] "Paths, trees and flowers", *Canadian Journal of Mathematics* 17 449–467.
- Erwe, F. [1964] *Differential- und Integralrechnung* I, II, Bibliographisches Institut, Mannheim.
- Fabian, T. [1958] "A linear programming model of integrated iron and steel production", *Management Science* 4 415–449.
- Farkas, J. [1902] "Theorie der einfachen Ungleichungen", *Journal für die reine und angewandte Mathematik* 124 1–27.
- Fasano, G. [1999] "Cargo analytical integration in space engineering: a three-dimensional model", in Ciriani et al (eds), *Operational Research in Industry*, MacMillan, 1999.
- Fiacco, A.V. and G.P. McCormick [1968] *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* Wiley & Sons, New York; republished 1990, SIAM, Philadelphia.
- Forrest, J.H. [1989] Personal communication.
- Forrest, J.H. and D. Goldfarb [1992] "Steepest-edge simplex algorithms for linear programming", *Mathematical Programming*, 57, 341–374.
- Forrest, J.H. and J.A. Tomlin [1972] "Updating triangular factors of the basis to maintain sparsity in the product form simplex method", *Mathematical Programming* 2 263–278.
- Fourier, J. [1822] *Théorie analytique de la chaleur*, republished in Grattan-Guiness, I. [1972] *Joseph Fourier 1768–1830*, MIT Press, Cambridge.
- Freund, R.M. [1985] "Postoptimal analysis of a linear program under simultaneous changes in the matrix coefficients", *Mathematical Programming Study* 24 1–13.
- Fulkerson, D.R. [1971] "Blocking and antiblocking pairs and polyhedra", *Mathematical Programming* 1 168–194.
- Gács, P. and L. Lovász [1981] "Khachian's algorithm for linear programming", *Mathematical Programming Study* 14 61–68.
- Gale, D. [1951] "Convex polyhedral cones and linear inequalities", in Koopmans T. (ed) *Activity Analysis of Production and Allocation*, Wiley, New York, 287–297.
- Gale, D. [1960] *The Theory of Linear Economic Models*, McGraw-Hill, New York.
- Gale, D. , Kuhn, H.W. and A.W. Tucker [1951] "Linear programming and the theory of games", in Koopmans (ed) *Activity Analysis of Production and Allocation*, Wiley, New York.
- Gantmacher, F.R. [1958] *Matrizenrechnung, Teil I und Teil II* VEB Deutscher Verlag der Wissenschaften, Berlin. [German translation of the Russian original dated 1954.]
- Garey, M.R. and D.S. Johnson [1979] *Computers and Intractability*, Freeman, San Francisco.
- Gass, S.I. [1955] "A first feasible solution to the linear programming problem", in Antosiewicz (ed) *Linear Programming Volume II*, National Bureau of Standards, Washington, 495–508.
- Gass, S.I. [1958] *Linear Programming: Methods and Applications*, McGraw-Hill, New York.
- Gass, S.I. and T.L. Saaty [1955] "The computational algorithm for the parametric objective function", *Naval Logistics Research Quarterly* 2 39–45.
- Gass, S.I. and T.L. Saaty [1955] "Parametric objective function. Part II: Generalization", *Operations Research* 3 395–401.

- Gay, D.M. [1974] "On Skolnik's proposed polynomial-time linear programming algorithm", *SIGMAP Newsletter* 16 15–21.
- Gay, D.M. [1978] "On combining the schemes of Reid and Saunders for sparse LP bases", in Duff and Stewart (eds) *Sparse Matrix Proceedings*, SIAM, Philadelphia, 313–334.
- Gay, D.M. [1987] "A variant of Karmarkar's linear programming algorithm for problems in standard form", *Mathematical Programming* 37 81–90.
- Gerstenhaber, M. [1951] "Theory of convex polyhedral cones", in Koopmans T. (ed) *Activity Analysis of Production and Allocation*, Wiley, New York, 298–316.
- de Ghellinck, G. and J-P Vial [1986] "A polynomial Newton method for linear programming", *Algorithmica* 1 425–453.
- Gilbert, J. and T. Peierls [1988] "Sparse partial pivoting in time proportional to arithmetic operations", *SIAM J. Sci. Stat. Computing*, 9 862–874.
- Gill, D.E., W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright [1986] "On projected Newton barrier methods for linear programming and an equivalent to Karmarkar's projective method", *Mathematical Programming* 36 183–209.
- Giloni, A. [2000], "Essays on Optimization in Data Analysis and Operations Management", Ph.D. thesis, Stern School of Business, New York University, May 2000.
- Goldfarb, D. [1977] "On the Bartels-Golub decomposition for linear programming bases", *Mathematical Programming* 13 272–279.
- Goldfarb, D. and J.K. Reid [1977] "A practicable steepest-edge simplex algorithm", *Mathematical Programming* 12 361–371.
- Goldfarb, D. and S. Mehrotra [1989] "A self-correcting version of Karmarkar's algorithm", *SIAM Journal on Numerical Analysis* 26 1006–1015.
- Goldfarb, D. and D. Xiao [1991] "On the complexity of a class of projective interior point methods", Manuscript, Columbia University, submitted to *Mathematics of Operations Research*.
- Goldfarb, D. and D. Xiao [1991] "A primal projective interior point method for linear programming", *Mathematical Programming* 51 17–43.
- Goldman, A.J. [1956] "Resolution and separation theorems for polyhedral convex sets", in Kuhn and Tucker (eds) *Linear Inequalities and Related Systems*, Princeton U. Press, Princeton, 41–51.
- Goldman, A.J. and A.W. Tucker [1956] "Theory of linear programming", in Kuhn and Tucker (eds), *Linear Inequalities and Related Systems*, Princeton U. Press, Princeton, 53–97.
- Golub, G.H. and C.F. Van Loan [1983] *Matrix Computations*, The Johns Hopkins University Press, Baltimore.
- Gonzaga, C. [1989] "An algorithm for solving linear programming in  $\mathcal{O}(n^3L)$  operations", in N. Megiddo (ed) *Progress in Mathematical Programming—Interior Point and Related Methods*, Springer-Verlag, Berlin 1–28.
- Gordan, P. [1873] "Über die Auflösung linearer Gleichungen mit reellen Coefficienten", *Mathematische Annalen* VI 23–28.
- Graves, R.L. and Ph. Wolfe (eds) [1963] *Recent Advances in Mathematical Programming*, McGraw-Hill, New York.
- Greenberg, H.J. [1983] "A functional description of ANALYZE: A computer assisted analysis system for linear programming models" *ACM Transactions on Mathematical Software* 9 18–56.
- Grigoriadis, M.D. [1971] "A dual generalized upper bounding technique", *Management Science* 17 269–284.
- Grötschel, M. [1977] *Polyhedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, Verlag Anton Hain, Meisenheim-am-Glan, FRG.
- Grötschel, M. and O. Holland [1987] "A cutting-plane algorithm for minimum perfect 2-matching", *Computing* 39 327–344.
- Grötschel, M. and O. Holland [1991] "Solution of large-scale symmetric travelling salesman problems", *Mathematical Programming* 51 141–202.
- Grötschel, M., M. Jünger and G. Reinelt [1984] "A cutting plane algorithm for the linear ordering problem", *Operations Research* 32 1195–1220.

- Grötschel, M., M. Jünger and G. Reinelt [1991] "Optimal control of plotting and drilling machines: A case study", *Zeitschrift für Operations Research – Methods and Models of Operations Research* 35 61–84.
- Grötschel, M., L. Lovász and A. Schrijver [1981] "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* 1 169–197.
- Grötschel, M., L. Lovász and A. Schrijver [1988] *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin.
- Grötschel, M., C.L. Monma and M. Stoer [1992] "Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints", *Operations Research* 40 309–330.
- Grötschel, M. and M. Padberg [1978] "On the symmetric traveling salesman problem: theory and computation", in R. Henn et al (eds), *Optimization and Operations Research*, Lecture Notes in Economics and Mathematical Systems 157, Springer, Berlin, 105–115.
- Grötschel, M. and M. Padberg [1985] "Polyhedral theory", in Lawler, E. et al (eds) *The Traveling Salesman Problem*, Wiley, New York, Chapter 8.
- Grötschel, M. and M. Padberg [1999] "Die optimierte Odyssee", *Spektrum der Wissenschaft*, 76–85, April 1999; translated into French as "L'odyssée abrégée", *Pour la Science*, August 1999; reprinted in *Spektrum der Wissenschaft*, Digest 2: Wissenschaftliches Rechnen, 32–41, October 1999.
- Grötschel, M. and W.R. Pulleyblank [1986] "Clique tree inequalities and the symmetric traveling salesman problem", *Mathematics of Operations Research* 11 537–569.
- Grünbaum, B. [1967] *Convex Polytopes*, Wiley, London.
- Guignard, M. and K. Spielberg [1981] "Logical reduction methods in zero-one programming: Minimal preferred inequalities", *Operations Research* 29 49–74.
- Hadley, G. [1962] *Linear Programming*, Addison-Wesley, Reading.
- Haimovich, M. [1983] "The simplex method is very good! – On the expected number of pivot steps and related properties of random linear programs", Technical Report, Graduate School of Business, Columbia University, N.Y.
- Harris, P.M.J [1973] "Pivot selection methods of the devex LP code", *Mathematical Programming* 5 1–28.
- Helbig Hansen, K. and J. Krarup [1974] "Improvements of the Held-Karp algorithm for the symmetric traveling-salesman problem", *Mathematical Programming* 7 87–96.
- Held, M. and Karp, R.M. [1971] "The traveling-salesman problem and minimum spanning trees: part II", *Mathematical Programming* 1 6–25.
- Hellerman E. and D. Rarick [1971] "Reinversion with the preassigned pivot procedure", *Mathematical Programming* 1 195–216.
- Hillier, F.S. and G.J. Lieberman [1967] *Introduction to Operations Research*, Holden-Day, San Francisco.
- Hitchcock, F.L. [1941] "The distribution of a product from several sources to numerous localities", *J. Math. Phys.* 20 224–230.
- Ho, J.K. and E. Loute [1981] "An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming", *Mathematical Programming* 20 303–326.
- Hoffman, A.J. [1952] "On approximate solutions of systems of linear inequalities", *Journal of Research of National Bureau of Standards* 49 263–265.
- Hoffman, A.J. [1953] "Cycling in the simplex method", National Bureau of Standards, Report No. 2974 , Washington.
- Hoffman, A.J [1955] "How to solve a linear programming problem ", in Antosiewicz (ed) *Linear Programming* Volume II, National Bureau of Standards, Washington, 397–424.
- Hoffman, A., Mannos, N., Sokolowsky, D. and N. Wiegman [1953] "Computational experience in solving linear programs", *Journal of the Society for Industrial and Applied Mathematics* 1 17–33.
- Hoffman, A. and J. Kruskal [1958] "Integral boundary points of convex polyhedra", in Kuhn, H. and A. Tucker (eds), *Linear inequalities and related systems*, Princeton University Press, Princeton 223–246.
- Hoffman, K.L. and M. Padberg [1985] "LP-based combinatorial problem solving", *Annals of Operations Research* 4 145–194.

- Hoffman, K.L. and M. Padberg [1991] "Improving LP-representations of zero-one linear programs for branch-and-cut", *ORSA Journal on Computing* 3 121-134.
- Hoffman, K.L. and M. Padberg [1993] "Solving airline crew scheduling problems by branch-and-cut", *Management Science* 39 657-682.
- Hooker, J.N. [1986] "Karmarkar's linear programming algorithm", *Interfaces* 16 75-90.
- Huard, P. [1967] "Resolution of mathematical programming with nonlinear constraints by the method of centres" in: J. Abadie (ed) *Nonlinear Programming*, North-Holland, Amsterdam, 209-219.
- Iri, M. and H. Imai [1986] "A multiplicative barrier function method for linear programming", *Algorithmica* 1 455-482.
- John, F. [1948] "Extremum problems with inequalities as subsidiary conditions" in *Studies and Essays*, Courant anniversary volume, New York, Interscience.
- Jünger, M., G. Reinelt and G. Rinaldi [1994] "The traveling salesman problem", Report No.92.113, Inst. für Informatik, Universität Köln (Germany), to appear in Ball, M.O. et al (eds) *Networks, Handbooks in Operations Research and Management Science*, North-Holland, Amsterdam (forthcoming).
- Kantorovich, L.V. [1939] "Mathematical methods in the organization and planning of production" English translation in *Management Science* (1960) 6 366-422.
- Kantorovich, L.V. [1942] "On the translocation of masses", *C.R.Acad.Sci. URSS* 37 199-201.
- Kantorovich, L.V. [1965] *The Best Use of Economic Resources*, Harvard U. Press, Cambridge.
- Karmarkar, N. [1984] "A new polynomial-time algorithm for linear programming", *Combinatorica* 4 373-395.
- Karp, R.M. and C.H. Papadimitriou [1982] "On linear characterization of combinatorial optimization problems" *SIAM Journal on Computing* 11 620-632.
- Khachian, L.G. [1979] "A polynomial algorithm in linear programming", *Soviet Mathematics Doklady* 20 191-194.
- Khachian, L.G. [1980] "Polynomial algorithms in linear programming", *USSR Comp. Math. and Math. Phys.* 20 53-72.
- Khachian, L.G. [1982] "On the exact solution of systems of linear inequalities and linear programming problems", *USSR Comp. Math. and Math. Phys.* 22 239-242.
- Khachian, L.G. [1982] "Convexity and computational complexity in polynomial programming", *Engineering Cybernetics* 22 46-56.
- Khinchin, A.Y. [1935] *Continued Fractions* (in Russian). English translation [1964], The University of Chicago Press, Chicago, Illinois.
- Kirchberger, P. [1903] "Über Tschebyschesche Annäherungsmethoden", *Mathematische Annalen* 57 509-540.
- Kiefer, J. [1952] "Sequential minimax search for a maximum", *Proc. Amer. Math. Soc.* 4 502-506.
- Klee, V. and G.J. Minty [1972] "How good is the simplex algorithm?" in Sisha (ed) *Inequalities - III*, Academic Press, New York 159-175.
- Klein, F. [1928] *Vorlesungen über nicht-euklidische Geometrie*, reproduction by Chelsea Publishing Company, New York.
- Koopmans, Tj. [1949] "Optimum utilization of the transport system", *Suppl. Econometrica* 7 136-146.
- Koopmans, T.C. (ed) [1951] *Activity Analysis of Production and Allocation*, Wiley, New York.
- Korte, B. and R. Schrader [1980] "A note on convergence proofs for Shor-Khachian methods", in Auslender, Oettli, Stoer (eds) *Optimization and Optimal Control*, Lecture Notes in Control and Information Sciences Vol.30, Springer, Berlin, New York, 51-57.
- Kotiah, T.C.T. and D.I. Steinberg [1978] "On the possibility of cycling with the simplex method", *Operations Research* 26 374-376.
- Kraemer, R.D. and M.R. Hillard [1991] "Mission (not) impossible", *OR/MS Today* April 1991 44-45.
- Kranich, E. [1993] "Interior point methods for mathematical programming: A bibliography", Universität Wuppertal, Germany.

- Kuhn, H.W. [1956] "Solvability and consistency for systems of linear equations and inequalities", *The American Mathematical Monthly* 63 217–232.
- Kuhn, H.W. and R.E. Quandt [1963] "An experimental study of the simplex method", *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society, Providence Vol 15 107–124.
- Kuhn, H.W. and A.W.Tucker (eds) [1956] *Linear Inequalities and Related Systems*, Annals of Mathematics Studies 38, Princeton U. Press, Princeton.
- Kulisch, U.W. and W.L. Miranker [1986] "The arithmetic of the digital computer: A new approach", *SIAM Review* 28 1–40.
- Land, A.H. and A.G. Doig [1960] "An automatic method for solving discrete programming problems", *Econometrica* 28 497–520.
- Lawler, E.L. and D.E. Woods [1966] "Branch-and-bound methods: A survey", *Operations Research* 14 699–719.
- Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds) [1985] *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, Chichester.
- Lehman, A. [1990] "The width-length inequality and degenerate projective planes", in Cook, W. and P.D. Seymour (eds), *Polyhedral Combinatorics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science.
- Lemke, C.E. [1954] "The dual method of solving the linear programming problem", *Naval Research Logistics Quarterly* 1 36–47.
- Leontief, W.W. [1928] "Die Wirtschaft als Kreislauf", *Archiv für Sozialwissenschaft und Sozialpolitik*, 60 577–623.
- Leontief, W.W [1951] *The Structure of the American Economy 1919–1929*, Oxford, New York.
- Levin, A.Y. [1965] "On an algorithm for convex function minimization", *Soviet Mathematics Doklady* 6 286–290.
- Little, J.D.C., K.G. Murty, D.W. Sweeney and C. Karel [1963] "An algorithm for the traveling salesman problem", *Operations Research* 11 979–989.
- Lustig, I.J., R.E. Marsten and D.F. Shanno [1994] "Interior point methods for linear programming: computational state of the art", *ORSA Journal on Computing* 6 1–14.
- Luther, H.A. and L.F. Guseman, Jr. [1962] "A finite sequentially compact process for the adjoints of matrices over arbitrary integral domains", *Comm. ACM* 5 447–448.
- Mangasarian, O.L. [1969] *Nonlinear Programming*, McGraw-Hill, New York.
- Manne, A.S. [1956] "Scheduling of Petroleum Refinery Operations", Harvard U. Press, Cambridge.
- Markowitz, H.M. [1957] "The elimination form of the inverse and its application to linear programming" *Management Science* 3 255–269.
- Marshall, K.T. and J.W. Suurballe [1969] "A note on cycling in the simplex method", *Naval Research Logistics Quarterly* 16 121–137.
- Marsten, R., R. Subramanian, I. Lustig and D. Shanno [1990] "Interior point methods for linear programming: Just call Newton, Lagrange, and Fiacco and McCormick!", *Interfaces* 20 105–116.
- Maurras, J.F. [1975] "Some results on the convex hull of hamiltonian cycles of symmetric complete graphs", in Roy, B. (ed), *Combinatorial programming: Methods and applications*, Reidel, Dordrecht 179–190.
- Maurras, J.F. [1976] Polytopes à sommets dans  $\{0, 1\}^n$ , Thèse de doctorat d' État, Université Paris VII, Paris.
- McClellan, M.T. [1973] "The exact solution of systems of linear equations with polynomial coefficients", *J. ACM* 20 563–588.
- McMullen, P. and G.C. Shephard [1971] *Convex polytopes and the upper bound conjecture*, London Math.Soc. Lecture Notes Series 3, Cambridge University Press, London.
- McShane, K.A., C.L. Monma and D. Shanno [1989] "An implementation of a primal-dual interior point method for linear programming", *ORSA Journal on Computing*, 1 70–83.
- Megiddo, N. [1989] (ed) *Progress in Mathematical Programming*, Springer Verlag, Berlin.

- Megiddo, N. [1989] "Pathways to the optimal set in linear programming", in N. Megiddo (ed) *Progress in Mathematical Programming: Interior Point Algorithms and Related Methods*, Springer Verlag, New York, 131-158.
- Mehta, D.R. [1974] *Working capital management*, Prentice Hall, Englewood Cliffs.
- Meyer, R.R. [1974] "On the existence of optimal solutions to integer and mixed integer programming problems", *Mathematical Programming* 7 223-235.
- Minkowski, H. [1896] *Geometrie der Zahlen (Erste Lieferung)* Teubner, Leipzig.
- Minkowski, H. [1897] "Allgemeine Lehrsätze über die konvexen Polyheder", *Nachrichten der königlichen Gesellschaft der Wissenschaften zu Göttingen, math.-physik. Klasse* 198-219
- Mityagin, B.S. [1969] "Two inequalities for volumes of convex bodies", *Math. Notes Acad. Science USSR* 5 61-65.
- Monge, G. [1781] "Mémoire sur la théorie des remblais et des déblais", *Mém. Acad. Royale* 666-704.
- Monteiro, R.D.C. and I. Adler [1989] "Interior path following primal-dual algorithms. Part I: Linear programming", *Mathematical Programming* 44 27-41.
- Moore, R.E. [1966] *Interval Analysis*, Prentice Hall, Englewood Cliffs.
- Motzkin, Th.S. [1933] *Beiträge zur Theorie der linearen Ungleichungen*, Doctoral Thesis, University of Basel. English translation in Cantor et al (eds) *Theodore S. Motzkin: Selected Papers*, Birkhäuser, Boston, 1983 1-81.
- Motzkin, T.S., H. Raiffa, G.L. Thompson and R.M. Thrall [1953] "The double description method" in Kuhn, H.W. and A.W. Tucker (eds) *Contributions to the Theory of Games Vol. II*, Princeton University Press, Princeton, 51 - 73.
- Müller-Merbach, H. [1970] *On Round-off Errors in Linear Programming*, Springer-Verlag, Berlin.
- Murty, K.G. [1983] *Linear Programming*, Wiley, New York.
- Neisser, H. [1932] "Lohnhöhe und Beschäftigungsgrad im Marktgleichgewicht", *Weltwirtschaftliches Archiv* 36 413-455.
- Nemhauser, G.L. and L.A. Wolsey [1988] *Integer and Combinatorial Optimization*, Wiley, New York.
- Newman, D.J. [1965] "Location of the maximum on unimodal surfaces", *Journal of the Association for Computing Machinery* 12 395-398.
- von Neumann, J. [1936] "Über ein ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes", *Ergebnisse eines mathematischen Kolloquiums* 8.
- von Neumann, J. and O. Morgenstern [1953] *Theory of Games and Economic Behavior*, Princeton U. Press, Princeton.
- Orchard-Hays, W. [1954] "A composite simplex algorithm - II", Technical Report P-525, The Rand Corporation, Santa Monica.
- Orchard-Hays, W. [1968] *Advanced Linear Programming Computing Techniques*, McGraw-Hill, New York.
- Orgler, Y.E. [1969] "An unequal period model for cash management decisions", *Management Science* 16 B77-B92.
- Orgler, Y.E. [1970] *Cash management: Methods and models*, Wadsworth Publ. Co., Belmont.
- Ostrowski, A.M. [1973] *Solution of equations in Euclidean and Banach spaces*, 3rd Ed., Academic Press, New York.
- Padberg, M. [1973] "On the facial structure of set packing polyhedra", *Mathematical Programming* 5 199-251.
- Padberg, M. [1974] "Perfect zero-one matrices", *Mathematical Programming* 6 180-196.
- Padberg, M. [1975] "Characterization of totally unimodular, balanced and perfect matrices" in Roy, B. (ed), *Combinatorial programming: Methods and applications*, Reidel, Dordrecht 275-284.
- Padberg, M. [1976] "Almost integral polyhedra related to certain combinatorial optimization problems", *Linear Algebra and Its Applications* 15 69-88.
- Padberg, M. [1985] "Solution of a nonlinear programming problem arising in the projective method for linear programming", Manuscript, New York University, New York.

- Padberg, M. [1986] "A different convergence proof for the projective method", *Operations Research Letters* 4 253–257.
- Padberg, M. [1988] "Total unimodularity and the Euler-subgraph problem", *Operations Research Letters* 7 173–179.
- Padberg, M. [1993] "Lehman's forbidden minor characterization of ideal 0-1 matrices", *Discrete Mathematics* 111 409–420.
- Padberg, M. [2000a] "Packing small boxes into a big box", to appear in *Mathematical Methods of Operations Research*, 52 (2000).
- Padberg, M. [2000b] "Approximating separable nonlinear functions via mixed zero-one programs", to appear in *Operations Research Letters*, 2000.
- Padberg, M. and M. Grötschel [1985] "Polyhedral computations", Chapter 9 in Lawler, E.L. et al (eds), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, Chichester.
- Padberg, M. and S. Hong [1980] "On the symmetric travelling salesman problem: a computational study", *Mathematical Programming Study* 12 78–107.
- Padberg, M. and M.R. Rao [1979] "The Russian method for linear inequalities and linear optimization", November 1979, Revised June 1980, Graduate School of Business Administration, New York University, New York.
- Padberg, M. and M.R. Rao [1979] "The Russian method for linear inequalities II: Approximate arithmetic", January 1980, Graduate School of Business Administration, New York University, New York.
- Padberg, M. and M.R. Rao [1981] "The Russian method for linear inequalities III: Bounded integer programming", Preprint, New York University, New York.
- Padberg, M. and M.R. Rao [1982] "Odd Minimum Cut-Sets and  $b$ -Matchings", *Mathematics of Operations Research* 7 67–80.
- Padberg, M. and G. Rinaldi [1987] "Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut", *Operations Research Letters* 6 1–7.
- Padberg, M. and G. Rinaldi [1990] "An efficient algorithm for the minimum capacity cut problem", *Mathematical Programming* 47 19–36.
- Padberg, M. and G. Rinaldi [1990] "Facet identification for the symmetric traveling salesman polytope", *Mathematical Programming* 47 219–257.
- Padberg, M. and G. Rinaldi [1991] "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems", *SIAM Review* 33 60–100.
- Padberg, M. and Ting-Yi Sung [1991] "An analytical comparison of different formulations of the travelling salesman problem", *Mathematical Programming* 52 315–357.
- Padberg, M. and Ting-Yi Sung [1997] "An analytic symmetrization of max flow-min cut", *Discrete Mathematics*, 165/166 531–545.
- Padberg, M., T. Van Roy and L. Wolsey [1985] "Valid linear inequalities for fixed charge problems", *Operations Research*, 33 (1985), 842–861.
- Padberg, M. and M. Wilczak [1999] "Optimal project selection when borrowing and lending rates differ", *Mathematical and Computer Modelling*, 29, 63–78.
- Perold, A.F. and G.B. Dantzig [1978] "A basis factorization method for block triangular linear programs", in Duff and Stewart (eds) *Proceedings of the Symposium on Sparse Matrix Computations*, Knoxville 283–312.
- Reid, J.K. [1982] "A sparsity exploiting variant of the Bartels-Golub decomposition for linear programming basis", *Mathematical Programming* 24 55–69.
- Reinelt, G. [1991] "TSPLIB – A traveling salesman problem library", *ORSA Journal on Computing* 3 376–384.
- Reinelt, G. [1994] *Contributions to practical traveling salesman problem solving*, Lecture Notes in Scientific Computing, Springer Verlag, Berlin..
- Renegar, J. [1988] "A polynomial-time algorithm, based on Newton's method, for linear programming", *Mathematical Programming* 40 59–93.

- Robinson, S.M. [1963] "Bounds for error in the solution set of a perturbed linear program", *Linear Algebra and its Applications* 6 69–81.
- Rockafellar, T. [1970] *Convex Analysis*, Princeton University Press, Princeton.
- Roehrkas, R.C. and G.C. Hughes [1990] "Crisis analysis", *OR/MS Today* Dec. 1990 22–27.
- Saaty, T.L. and S.I. Gass [1954] "The parametric objective function. Part I", *Operations Research* 2 316–319.
- Savelsbergh, M. [1994] "Preprocessing and probing techniques for mixed integer programming problems", *ORSA Journal on Computing* 6 445–454.
- Sakarovitch, M. [1971] *Notes on Linear Programming*, Van Nostrand Reinhold, New York.
- Saunders, M.A. [1976] "A fast, stable implementation of the simplex method using Bartels-Golub updating", in Bunch and Rose (eds) *Sparse Matrix Computations*, Academic Press, New York 213–226.
- Schlesinger, K. [1934] "Über die Produktionsgleichungen der ökonomischen Wert-lehre", *Ergebnisse eines mathematischen Kolloquiums* 6 10–11.
- Schrader, R. [1982] "Ellipsoid methods", in D. Korte (ed) *Modern Applied Mathematics – Optimization and Operations Research*, North-Holland, Amsterdam, 265–311.
- Schrijver, A. [1986] *Theory of Linear and Integer Programming*, Wiley, Chichester.
- Schuppe, Th. [1991] "OR goes to war", *OR/MS Today* April 1991 36–44.
- Shaw (Xiao), D. and D. Goldfarb [1991] "A path following projective interior point method for linear programming", Manuscript, Columbia University, New York, to appear in *SIAM Journal on Optimization*.
- Shor, N.Z. [1970] "Utilization of the operation of space dilatation in the minimization of convex functions", *Cybernetics* 6 7–15.
- Shor, N.Z. [1970] "Convergence rate of the gradient method with dilatation of the space", *Cybernetics* 6 102–108.
- Shor, N.Z. [1977] "Cut-off method with space extension in convex programming problems", *Cybernetics* 13 94–96.
- Simmonard, M. [1966] *Linear Programming*, Prentice-Hall, Englewood Cliffs.
- Skolnik, H. [1973] "Outline of a new algorithm for linear programming", Paper presented at the Stanford Mathematical Programming Symposium, August 1973.
- Smale, S. [1983] "The problem of the average speed of the simplex method", in Bachem et al (eds) *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin 530–539.
- Sperner, E. [1951] *Einführung in die Analytische Geometrie und Algebra*, Zweiter Teil, Vandenhoeck & Ruprecht, Göttingen.
- von Stackelberg, H. [1933] "Zwei kritische Bemerkungen zur Preistheorie Gustav Cassels", *Zeitschrift für Nationalökonomie* 4 456–472.
- Stiemke, E. [1915] "Über positive Lösungen homogener linearer Gleichungen", *Mathematische Annalen* 76 340–342.
- Stigler, G.J. [1945] "The cost of subsistence", *Journal of Farm Economics* 27 303–314.
- Stoer, J. and C. Witzgall [1970] *Convexity and Optimization in Finite Dimensions I*, Springer, Berlin.
- Stuckey, P. [1991] "Incremental linear constraint solving and detection of implicit equalities", *ORSA Journal on Computing* 3 269–274.
- Suhl, U. and L. Suhl [1990] "Computing sparse LU factorizations for large-scale linear programming bases", *ORSA Journal on Computing* 4 325–335.
- Symmonds, G.H. [1955] *Linear Programming: The Solution of Refinery Problems*, Esso Standard Oil Co., New York.
- Tewarson, R.R. [1973] *Sparse Matrices*, Academic Press, New York.
- Tietz, H. [1962] *Vorlesung über Analytische Geometrie*, Aschendorffsche Verlagsbuchhandlung, Münster/Westfalia.
- Todd, M.J. and Burrell, B.P. [1986] "An extension of Karmarkar's algorithm for linear programming using dual variables", *Algorithmica* 1 409–424.

- Tomlin, J.A. [1971] "An improved branch-and-bound method for integer programming", *Operations Research* 19 1070–1074.
- Tomlin, J.A. [1972] "Modifying triangular factors of the basis in the simplex method", in Rose and Willoughby (eds) *Sparse Matrices and Their Applications*, Plenum, New York 77–85.
- Tucker, A.W. [1956] "Dual systems of homogeneous linear equations", in Kuhn and Tucker (eds) *Linear Inequalities and Related Systems*, Princeton U. Press, Princeton, 3–18.
- Vajda, S. [1956] *The Theory of Games and Linear Programming*, Wiley, New York.
- Vanderbei, R.J. [1993] "ALPO: Another linear programming optimizer", *ORSA Journal on Computing* 5 134–146.
- Vanderbei, R.J., M.S. Meketon and B.A. Freedman [1986] "A modification of Karmarkar's linear programming algorithm", *Algorithmica* 4 395–407.
- Van Roy, T. and L.A. Wolsey [1987] "Solving mixed integer programming problems using automatic reformulation" *Operations Research* 35 45–57.
- Veinott, R. and G.B. Dantzig [1968] "Integral extreme points", *SIAM Review* 10 371–372.
- Wagner, H.M. [1958] "The dual simplex algorithm for bounded variables", *Naval Research Logistics Quarterly* 5 257–261.
- Wagner, H.M. [1969] *Principles of Operations Research*, Prentice-Hall, Englewood Cliffs.
- Wald, A. [1934] "Über die eindeutige positive Lösbarkeit der Produktionsgleichungen", *Ergebnisse eines mathematischen Kolloquiums* 6 12–18.
- Wald, A. [1935] "Über die Produktionsgleichungen der ökonomischen Wertlehre", *Ergebnisse eines mathematischen Kolloquiums* 7 1–6.
- Weingartner, H.M. [1967] *Mathematical Programming and the Analysis of Capital Budgeting Problems*, Markham Publ. Co., Chicago.
- Weyl, H. [1935] "Elementare Theorie der konvexen Polyeder", *Commentarii Mathematici Helvetici* 7 290–306. [Translation in: Kuhn, H. and A. Tucker (eds) *Contributions to the Theory of Games I*, Princeton University Press, Princeton, 3–18.]
- Wilkinson, J.H. [1971] "Modern error analysis", *SIAM Review* 13 548–568.
- Williams, H.P. [1985] *Model Building in Mathematical Programming*, 2nd edition, Wiley, New York.
- Witzgall, C., Boggs, P.T. and P.D. Domich [1990] "On the convergence behavior of trajectories for linear programming", *Contemporary Mathematics* 114 161–187.
- Wolfe, Ph. [1955] "Reduction of systems of linear relations", in Antosiewicz (ed) *Linear Programming* Volume II, National Bureau of Standards, Washington, 449–451.
- Wolfe, Ph. [1965] "Errors in the solution of linear programming problems" in Rau (ed) *Error in Digital Computation* Volume II, Wiley, New York 271–284.
- Wolfe, Ph. [1963] "A technique for resolving degeneracy in linear programming", *Journal of the Society for Industrial and Applied Mathematics* 11 205–211.
- Wolfe, Ph. [1978] Personal communication.
- Wolfe, P. [1980] "A bibliography for the ellipsoid algorithm", IBM Research Center, Yorktown Heights, NY.
- Wolsey, L. [1989] "Strong formulations for mixed integer programming: A Survey", *Mathematical Programming* 45 173–191.
- Wood, M.K. and G.B. Dantzig [1949] "Programming of interdependent activities I: General discussion", *Econometrica* 17 193–199.
- Yemelichev, V.A., M.M. Kovalev and M.K. Kravtsov [1984] *Polytopes, Graphs and Optimisation* translated by G.H. Lawden, Cambridge University Press, Cambridge.
- Yudin, D.B. and Nemirovskii, A.S. [1976] "Informational complexity and efficient methods for the solution of convex extremal problems" *Matekon* 13 3–25.
- Zeuthen, F. [1933] "Das Prinzip der Knappheit, technische Kombination und ökonomische Qualität", *Zeitschrift für Nationalökonomie* 7 1–24.

- Zhang, S. [1991] "On anti-cycling pivoting rules for the simplex method", *Operations Research Letters* 10 189-192.
- Zionts, S. [1974] *Linear and Integer Programming*, Prentice-Hall, Englewood Cliffs.
- Zurmühl, R. [1958] *Matrizen*, Springer Verlag, Berlin, 2<sup>nd</sup> edition.

# Index

- $\ell_1$ -norm, 126
- $\ell_2$ -norm, 126
- $\ell_\infty$ -norm, 126, 280
- $\varepsilon$ -optimal set, 287
- $\varepsilon$ -optimal solution, 287
- $\varepsilon$ -solidification, 280
- $v_0$ -polar, 135
- adjacent extreme points, 131, 132
- affine:
  - combination, 125
  - hull, 125
  - rank, 125
  - subspace, 125
  - transformation, 132
- affinely:
  - dependent, 125
  - independent, 125
- algorithm
  - best approximation, 284, 309
  - binary numbers, 78
  - binary search 1, 143
  - binary search 2, 144
  - branch-and-bound, 324
  - branch-and-cut, 325
  - cutting plane, 325
  - double description, 136, 172
    - all-integer, 139, 179
    - basis, 179, 182
    - MDDA, 139, 179
    - modified, 139, 179
  - ellipsoid
    - basic, 270
    - CCS, 288
    - DCS, 273, 297
  - Euclidean, 137
  - Gaussian elimination, 149
    - division free, 151, 191
    - with Euclidean reduction, 194
  - iterative scheme, 228, 255
  - LIST-and-CHECK, 264
  - Newtonian, 229, 259
  - projective, 221, 248
    - basic, 205, 232
- simplex
  - BV, 90
  - dual, 96, 108
  - Dual BV, 118
  - dynamic, 98, 113
  - primal, 63, 69
- all-integer
  - Gaussian elimination, 152
  - inversion routine, 139
- apex, 132
- applications to/examples in
  - automatized production, 399
  - cash budgeting, 362
  - cash management, 359
  - discrete-valued variables, 345
  - job shop scheduling, 121
  - nonlinear function approximation, 346
  - operations management, 371
  - packing boxes, 355
  - production, 371
  - project selection, 350
- arithmetic mean, 155
- asymptotic cone, 130, 331
- backward substitution, 150
- ball, 153
- barrier function, 224
  - geometric, 224
  - logarithmic, 224, 226
- basic ellipsoid algorithm, 270, 273
- basic projective algorithm, 205
  - approximate problem, 203
    - convergence of the iterates, 205
    - solution, 204
  - initialization of the, 206
  - step complexity of, 206
- basis, 47
  - dual, 96
  - feasible, 47
  - notation, 48
  - of a subspace, 126
  - position in, 48
- basis
  - change of, 55

optimal, 55  
 basis algorithm, 138  
 best approximation, 282  
     algorithm, 284  
     problem, 284  
 binary search, 143  
     algorithm 1, 143  
     algorithm 2, 144  
 block pivots, 148  
 blow-up factor, 154  
     in the ellipsoid algorithm, 264  
 branch-and-bound, 324  
     finiteness for MIP, 331  
 branch-and-cut, 265, 325  
  
 Carathéodory's theorem, 131  
 Cauchy-Schwarz inequality, 126  
 center of gravity, 223  
 center of the quadric, 153  
 centering direction, 228  
 central cut ellipsoid algorithm, 288  
 central path, 227  
 centroid, 223  
 changing elements of the coefficient matrix, 98  
 characteristic cone, *see* asymptotic cone  
 choice rules:  
     for pivot column, 64, 147, 150  
     for pivot row, 65, 150  
 Cholesky factorization, 150  
     integer, 151  
 combinatorial optimization, 323, 399  
 complementary slackness, 94  
 complete formulation, 328  
 cone, 130  
     extreme ray of, 130  
     lineality space of, 130  
     pointed, 130  
     polyhedral, 130  
 conical hull, 131  
 constraint identification problem, 145  
 continued fraction, 285  
 convex hull, 131  
 correctness:  
     of CCS ellipsoid algorithm, 289  
     of DCS ellipsoid algorithm, 277

of double description algorithm, 136  
 of dual simplex algorithm, 97  
 of ellipsoid algorithm, 270  
 of simplex algorithm, 65  
 of the Newtonian algorithm, 229  
 of the projective algorithm, 222  
 Cramer's rule, 153  
 cross multiplication, 150  
 cross ratio, 215  
  
 deep cut, 272  
 definition  
      $\varepsilon$ -solidification, 280  
     CO: cone, 130  
     cross ratio, 216  
     DI: dimension, etc., 125  
     EP: extreme point, 128  
     FA: face, 128  
     FC: facet complexity, 141  
     Hadamard product, 226  
     HU: hulls, 131  
     linear optimization problem, 289  
     linear program, 39  
         canonical form, 39  
         standard form, 39  
     P1: polyhedron, 127  
     P2: polyhedron, 133  
     PI: perfect etc., 335  
     polyhedral separation problem, 289  
     Schur complement, 40  
     TU: totally unimodular, 335  
     VE: valid equation, 332  
     VI: valid inequality, 332  
 degeneracy, 148  
 description  
     finite linear, 126  
     ideal, 129  
     minimal complete, 129  
     quasi-unique, 129  
     digital size, 141  
     dimension, 125  
     displaced asymptotic cone, 132  
     displaced subspace, 125  
     displacement, *see* translation  
     distance, 125  
     distance function, 125

- division free, 151
- division free Gaussian algorithm, 152
- double description algorithm, 136
- dual simplex algorithm, 96, 108
- duality
  - strong
    - theorem of, 94
- dynamic simplex algorithm, 98, 99, 113
- ellipsoid, 154
  - "halving", 266
  - volume of an, 155
- ellipsoidal norm, 154
- Euclidean:
  - algorithm, 137
  - norm, 126
  - reduction, 137
- extremal directions, 131
- extreme point, 128
- facet complexity, *see* rational polyhedron
- feasibility direction, 228
- feasibility problem, 143
- feasible triplet, 229
- fill-in, 149
- finite generator, 126
- finiteness:
  - of basic projective algorithm, 206
  - of branch-and-bound, 331
  - of CCS ellipsoid algorithm, 289
  - of DCS ellipsoid algorithm, 277
  - of dual simplex algorithm, 97
  - of ellipsoid algorithm, 270
  - of simplex algorithm, 65
  - of the Newtonian algorithm, 229
  - of the projective algorithm, 222
- flat, 129
- forward substitution, 150
- Frobenius norm, 265
- full dimensional, 125
- fundamental theorem of linear programming,
  - 206
- gamma function, 155
- Gaussian elimination
  - division free
    - iterative step of, 151
- iterative step of, 149
- generation
  - column, 99
  - row, 99
- geometric center, 224
- geometric mean, 155, 224
- geometric/arithmetic mean inequality, 156
- Gram-Schmidt orthogonalization, 127
- greatest common divisor (g.c.d.), 137
- Hadamard inequality, 127
- Hadamard product, 226
- halfline, 125
- halfspace, 126
  - open, 126
- homogenization, 131
- hyperplane, 126
- Hölder's inequality, 282, 308
- improper face, 128
- improper point, 210
- inhomogeneous linear equation, 126
- integrality property, 335
- interior point algorithm, 201
- large step, 272
- length, 126
- lexicographically maximal point, 145
- line search, 272
- linearity space, 127
- linear hull, 125
- linear optimization problem, 142, 289
- linear program
  - dual, 93
  - large-scale, 64, 98
  - primal, 93
  - structured, 39
- linear transformation, 132
- log-central path, 227
- logarithmic center, 224
- matrix
  - balanced, 336
  - Cholesky factorization, 150
    - integer, 151
  - eigenvalue of a, 154
  - eigenvector of, 154

- ideal, 335
- indefinite, 153
- inversion of, 40
- partitioned, 39
- perfect, 335
- permutation, 150
- positive definite, 153
  - orthonormal, 154
- positive semi-definite, 153
- Schur complement of, 40
- totally unimodular, 335
- triangular factors of, 150
- method of centers, 224
- Minkowski's theorem, 132
- MIP, *see* mixed-integer programming
- MIP formulation, 323
  - complete, 328
  - existence, 330
  - facet complexity, 331
  - vertex complexity, 331
- discrete mixed set, 327
- extremal characterization, 334
- facets, 333
  - characterization, 334
- ideal, 332, 333
- preprocessing, 325
- valid equation, 332
- valid<sup>#</sup> inequality, 332
- mixed-integer linear program, 323
- modified double description Algorithm, 139
- Newton's method, 227
- Newton:
  - direction, 227
  - step, 227
- Newtonian algorithm, 228
- norm:, 125
  - dual, 282
  - Frobenius, 265
  - of column, 65
  - of matrix, 265
  - polytopal, 282
- orthogonal:
  - complement, 126
  - decomposition, 128
  - projection, 126, 135
- properties of, 204
- vectors, 126
- outer inclusion principle, 146, 278
- parallel vectors, 126
- parametric:
  - objective function, 112
  - right-hand-side, 97
- perturbation, 145
- pivot:
  - column, 63
  - operation, 63
  - row, 63
  - rules, 64, 65
- polyhedral separation problem, 289
- polyhedron, 127, 133
  - blunt, 128
  - canonical generator of, 135
  - equivalence of definitions, 133
  - extreme point of, 128
  - extreme rays of, 131
  - face of, 128
  - facet of, 128
  - full dimensional, 128
  - image of, 132
  - integral, 334
  - line free, 127
  - linear description of, 127
  - minimal face of, 128
  - minimal generator of, 133
  - pointed, 128
  - rational, *see* rational polyhedron
  - solid, 129
- polytope, 131
  - dual, 282
- position of variable in basis, 48
- post-optimality, 97
- pricing, 64
- principal axis transformation, 155
- principal minors, 154
- projective
  - curve, 211
- projective algorithm, 221
  - geometry of, 211
  - initialization of the, 222
  - optimization problem, 202

- convergence of the iterates, 220
- solution in  $x$ -space, 207
- solution in  $y$ -space, 209, 214
- the iterative step of the, 203, 207, 220
- projective images, 215
- projective space  $\mathcal{P}_n$ , 213, 214
- projective transformation, 201
- proper face, 128
- pure integer program, 323
- purging, 99
- quadratic form, 153
- quasi-uniqueness
  - of pointwise description, 135
- rank-one update, 55
- rational polyhedron, 140
  - facet complexity of, 141
  - vertex complexity of, 141
- rational rounding, 282
- recession cone, *see* asymptotic cone
- reduced cost, 49
- redundant inequality, 95
- reflection on a circle, 218
- regular polyhedra, 127
- relative interior, 129
- restricted feasibility problem, 142
- scaling
  - a vector, 40
- separation problem, 145
  - polyhedral, 289
- separator
  - most violated, 280
  - optimal, 280
- simplex algorithm, 63, 90, 118
  - two-phase method of, 64
- simplex paths, 147
- sliding objective, 272
- solid, 129
- solution
  - $\varepsilon$ -feasible, 273
  - $\varepsilon$ -optimal, 273
  - basic feasible, 47
  - degenerate, 47
  - bounded, 47
  - feasible, 47
  - finite, 47
  - infeasible, 47
  - optimal, 47
- sphere, 153
  - unit, 154
- steepest descent direction, 228
- steepest edge criteria, 147
- subspace, 125
- system of equations
  - equivalent, 47
  - solving a, 153
- system of linear inequalities
  - solvable, 95
- translation, 125
- traveling salesman problem, 399
  - (refined) comb inequalities, 405
  - examples, 407, 409
  - formulation, 401
  - progress in exact solution, 407
- triangle inequality, 125
- unique:
  - minimal generator, 134
  - optimizer, 48
- valid equation, 129, 332
- valid<sup>#</sup> inequality, 332
- variable:
  - entering the basis, 56
  - leaving the basis, 56
- Weyl's theorem, 133

# Universitext

---

- Aksoy, A.; Khamsi, M. A.: Methods in Fixed Point Theory  
Anderson, M.: Topics in Complex Analysis  
Aoki, M.: State Space Modeling of Time Series  
Aupetit, B.: A Primer on Spectral Theory  
Bachem, A.; Kern, W.: Linear Programming Duality  
Benedetti, R.; Petronio, C.: Lectures on Hyperbolic Geometry  
Berger, M.: Geometry I, and II  
Bliedtner, J.; Hansen, W.: Potential Theory  
Boltyanski, V.; Martini, H.; Soltan, P. S.: Excursion into Combinatorial Geometry  
Booss, B.; Bleecker, D. D.: Topology and Analysis  
Borkar, V. S.: Probability Theory  
Carleson, L.; Gamelin, T.: Complex Dynamics  
Cecil, T. E.: Lie Sphere Geometry: With Applications of Submanifolds  
Chae, S. B.: Lebesgue Integration  
Chandrasekharan, K.: Classical Fourier Transform  
Charlap, L. S.: Bieberbach Groups and Flat Manifolds  
Chern, S.: Complex Manifolds without Potential Theory  
Chorin, A. J.; Marsden, J. E.: Mathematical Introduction to Fluid Mechanics  
Cohn, H.: A Classical Invitation to Algebraic Numbers and Class Fields  
Curtis, M. L.: Abstract Linear Algebra  
Curtis, M. L.: Matrix Groups  
Dalen, D. van: Logic and Structure  
Das, A.: The Special Theory of Relativity: A Mathematical Exposition  
Devlin, K. J.: Fundamentals of Contemporary Set Theory  
DiBenedetto, E.: Degenerate Parabolic Equations  
Diener, F.; Diener, M.: Nonstandard Analysis in Practice  
Dimca, A.: Singularities and Topology of Hypersurfaces  
DoCarmo, M. P.: Differential Forms and Applications  
Edwards, R. E.: A Formal Background to Higher Mathematics Ia, and Ib  
Edwards, R. E.: A Formal Background to Higher Mathematics IIa, and IIb  
Emery, M.: Stochastic Calculus in Manifolds  
Endler, O.: Valuation Theory  
Erez, B.: Galois Modules in Arithmetic  
Foulds, L. R.: Graph Theory Applications  
Frauenthal, J. C.: Mathematical Modeling in Epidemiology  
Fuks, D. B.; Rokhlin, V. A.: Beginner's Course in Topology  
Fuhrmann, P. A.: A Polynomial Approach to Linear Algebra  
Gabisch, G.; Lorenz, H.-W.: Business Cycle Theory  
Gallot, S.; Hulin, D.; Lafontaine, J.: Riemannian Geometry  
Gardiner, C. F.: A First Course in Group Theory  
Gårding, L.; Tambour, T.: Algebra for Computer Science  
Godbillon, C.: Dynamical Systems on Surfaces  
Goldblatt, R.: Orthogonality and Spacetime Geometry  
Gouvêa, F. Q.:  $p$ -Adic Numbers  
Gustafson, K. E.; Rao, D. K. M.: Numerical Range. The Field of Values of Linear Operators and Matrices  
Hahn, A. J.: Quadratic Algebras, Clifford Algebras, and Arithmetic Witt Groups  
Hájek, P.; Havránek, T.: Mechanizing Hypothesis Formation  
Hlawka, E.; Schoißengeier, J.; Taschner, R.: Geometric and Analytic Number Theory  
Holmgren, R. A.: A First Course in Discrete Dynamical System  
Howe, R., Tan, E. Ch.: Non-Abelian Harmonic Analysis  
Howes, N. R.: Modern Analysis and Topology  
Humi, M., Miller, W.: Second Course in Ordinary Differential Equations for Scientists and Engineers

- Hurwitz, A.; Kritikos, N.: Lectures on Number Theory
- Iversen, B.: Cohomology of Sheaves
- Jennings, G. A.: Modern Geometry with Applications
- Jones, A.; Morris, S. A.; Pearson, K. R.: Abstract Algebra and Famous Impossibilities
- Jost, J.: Riemannian Geometry and Geometric Analysis
- Jost, J.: Compact Riemann Surfaces
- Kannan, R.; Krueger, C. K.: Advanced Analysis on the Real Line
- Kelly, P.; Matthews, G.: The Non-Euclidean Hyperbolic Plane
- Kempf, G.: Complex Abelian Varieties and Theta Functions
- Kloeden, P. E.; Platen, E.; Schurz, H.: Numerical Solution of SDE Through Computer Experiments
- Kostrikin, A. I.: Introduction to Algebra
- Krasnoselskii, M. A.; Pokrovskii, A. V.: Systems with Hysteresis
- Luecking, D. H., Rubel, L. A.: Complex Analysis. A Functional Analysis Approach
- Ma, Zhi-Ming; Roeckner, M.: Introduction to the Theory of (non-symmetric) Dirichlet Forms
- Mac Lane, S.; Moerdijk, I.: Sheaves in Geometry and Logic
- Marcus, D. A.: Number Fields
- Mc Carthy, P. J.: Introduction to Arithmetical Functions
- Meyer, R. M.: Essential Mathematics for Applied Field
- Meyer-Nieberg, P.: Banach Lattices
- Mines, R.; Richman, F.; Ruitenberg, W.: A Course in Constructive Algebra
- Moise, E. E.: Introductory Problem Courses in Analysis and Topology
- Montesinos-Amilibia, J. M.: Classical Tesselations and Three Manifolds
- Nikulin, V. V.; Shafarevich, I. R.: Geometries and Groups
- Morris, P.: Introduction to Game Theory
- Oden, J. J.; Reddy, J. N.: Variational Methods in Theoretical Mechanics
- Øksendal, B.: Stochastic Differential Equations
- Porter, J. R.; Woods, R. G.: Extensions and Absolutes of Hausdorff Spaces
- Ramsay, A.; Richtmeyer, R. D.: Introduction to Hyperbolic Geometry
- Rees, E. G.: Notes on Geometry
- Reisel, R. B.: Elementary Theory of Metric Spaces
- Rey, W. J. J.: Introduction to Robust and Quasi-Robust Statistical Methods
- Rickart, C. E.: Natural Function Algebras
- Rotman, J. J.: Galois Theory
- Rubel, L. A.: Entire and Meromorphic Functions
- Rybakowski, K. P.: The Homotopy Index and Partial Differential Equations
- Sagan, H.: Space-Filling Curves
- Samelson, H.: Notes on Lie Algebras
- Schiff, J. L.: Normal Families
- Sengupta, J. K.: Optimal Decisions under Uncertainty
- Séroul, R.: Programming for Mathematicians
- Shapiro, J. H.: Composition Operators and Classical Function Theory
- Simonnet, M.: Measures and Probabilities
- Smith, K. T.: Power Series from a Computational Point of View
- Smyrski, C.: Logical Number Theory I. An Introduction
- Smyrski, C.: Self-Reference and Modal Logic
- Stanisic, M. M.: The Mathematical Theory of Turbulence
- Stichtenoth, H.: Algebraic Function
- Stillwell, J.: Geometry of Surfaces
- Stroock, D. W.: An Introduction to the Theory of Large Deviations
- Sunada, T.: The Fundamental Group and Laplacian (to appear)
- Sunder, V. S.: An Invitation to von Neumann Algebras
- Tamme, G.: Introduction to Étale Cohomology
- Tondeur, P.: Foliations on Riemannian Manifolds
- Verhulst, F.: Nonlinear Differential Equations and Dynamical Systems
- Zaanen, A.C.: Continuity, Integration and Fourier Theory
- Zong, C.: Strange Phenomena in Convex and Discrete Geometry