COLUMN GENERATION

GERAD 25th Anniversary Series

- **Essays and Surveys in Global Optimization**
  Charles Audet, Pierre Hansen, and Gilles Savard, editors

- **Graph Theory and Combinatorial Optimization**
  David Avis, Alain Hertz, and Odile Marcotte, editors

- **Numerical Methods in Finance**
  Hatem Ben-Ameur and Michèle Breton, editors

- **Analysis, Control and Optimization of Complex Dynamic Systems**
  El-Kébir Boukas and Roland Malhamé, editors

- **Column Generation**
  Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors

- **Statistical Modeling and Analysis for Complex Data Problems**
  Pierre Duchesne and Bruno Rémillard, editors

- **Performance Evaluation and Planning Methods for the Next Generation Internet**
  André Girard, Brunilde Sansò, and Félisa Vázquez-Abad, editors

- **Dynamic Games: Theory and Applications**
  Alain Haurie and Georges Zaccour, editors

- **Logistics Systems: Design and Optimization**
  André Langevin and Diane Riopel, editors

- **Energy and Environment**
  Richard Loulou, Jean-Philippe Waaub, and Georges Zaccour, editors

COLUMN GENERATION

Edited by
GUY DESAULNIERS
GERAD and École Polytechnique de Montréal

JACQUES DESROSIERS
GERAD and HEC Montréal

MARIUS M. SOLOMON
GERAD and Northeastern University, Boston

Springer

Guy Desaulniers
GERAD & École Polytechnique de Montréal
Montréal, Canada

Jacques Desrosiers
GERAD and HEC
Montréal, Canada

Marius M. Solomon
GERAD and Northeastern University
Boston, MA, USA

Library of Congress Cataloging-in-Publication Data

Column generation / edited by Guy Desaulniers, Jacques Desrosiers, Marius M. Solomon.
p. cm. – (GERAD 25th anniversary series)
Includes bibliographical references.
ISBN-13: 978-0-387-25485-2 (acid-free paper)
ISBN-10: 0-387-25485-4 (acid-free paper)
ISBN-10: 0-387-25486-2 (e-book)
1. Production scheduling. I. Desaulniers, Guy. II. Desrosiers, Jacques. III. Solomon, Marius M. IV. Series.

TS 157.5.C64 2005
658.5'3—dc22

2005043438

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11053088

springeronline.com

Foreword

GERAD celebrates this year its 25th anniversary. The Center was created in 1980 by a small group of professors and researchers of HEC Montréal, McGill University and of the École Polytechnique de Montréal. GERAD's activities achieved sufficient scope to justify its conversion in June 1988 into a Joint Research Centre of HEC Montréal, the École Polytechnique de Montréal and McGill University. In 1996, the Université du Québec à Montréal joined these three institutions. GERAD has fifty members (professors), more than twenty research associates and post doctoral students and more than two hundreds master and Ph.D. students.

GERAD is a multi-university center and a vital forum for the development of operations research. Its mission is defined around the following four complementarily objectives:

- The original and expert contribution to all research fields in GERAD's area of expertise;
- The dissemination of research results in the best scientific outlets as well as in the society in general;
- The training of graduate students and post doctoral researchers;
- The contribution to the economic community by solving important problems and providing transferable tools.

GERAD's research thrusts and fields of expertise are as follows:

- Development of mathematical analysis tools and techniques to solve the complex problems that arise in management sciences and engineering;
- Development of algorithms to resolve such problems efficiently;
- Application of these techniques and tools to problems posed in related disciplines, such as statistics, financial engineering, game theory and artificial intelligence;
- Application of advanced tools to optimization and planning of large technical and economic systems, such as energy systems, transportation/communication networks, and production systems;
- Integration of scientific findings into software, expert systems and decision-support systems that can be used by industry.

One of the marking events of the celebrations of the 25th anniversary of GERAD is the publication of ten volumes covering most of the Center's research areas of expertise. The list follows: **Essays and Surveys in Global Optimization**, edited by C. Audet, P. Hansen and G. Savard; **Graph Theory and Combinatorial Optimization**, edited by D. Avis, A. Hertz and O. Marcotte; **Numerical Methods in Finance**, edited by H. Ben-Ameur and M. Breton; **Analysis, Control and Optimization of Complex Dynamic Systems**, edited by E.K. Boukas and R. Malhamé; **Column Generation**, edited by G. Desaulniers, J. Desrosiers and M.M. Solomon; **Statistical Modeling and Analysis for Complex Data Problems**, edited by P. Duchesne and B. Rémillard; **Performance Evaluation and Planning Methods for the Next Generation Internet**, edited by A. Girard, B. Sansò and F. Vázquez-Abad; **Dynamic Games: Theory and Applications**, edited by A. Haurie and G. Zaccour; **Logistics Systems: Design and Optimization**, edited by A. Langevin and D. Riopel; **Energy and Environment**, edited by R. Loulou, J.-P. Waaub and G. Zaccour.

I would like to express my gratitude to the Editors of the ten volumes, to the authors who accepted with great enthusiasm to submit their work and to the reviewers for their benevolent work and timely response. I would also like to thank Mrs. Nicole Paradis, Francine Benoît and Louise Letendre and Mr. André Montpetit for their excellent editing work.

The GERAD group has earned its reputation as a worldwide leader in its field. This is certainly due to the enthusiasm and motivation of GERAD's researchers and students, but also to the funding and the infrastructures available. I would like to seize the opportunity to thank the organizations that, from the beginning, believed in the potential and the value of GERAD and have supported it over the years. These are HEC Montréal, École Polytechnique de Montréal, McGill University, Université du Québec à Montréal and, of course, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

Georges Zaccour
Director of GERAD

Avant-propos

Le Groupe d'études et de recherche en analyse des décisions (GERAD) fête cette année son vingt-cinquième anniversaire. Fondé en 1980 par une poignée de professeurs et chercheurs de HEC Montréal engagés dans des recherches en équipe avec des collègues de l'Université McGill et de l'École Polytechnique de Montréal, le Centre comporte maintenant une cinquantaine de membres, plus d'une vingtaine de professionnels de recherche et stagiaires post-doctoraux et plus de 200 étudiants des cycles supérieurs. Les activités du GERAD ont pris suffisamment d'ampleur pour justifier en juin 1988 sa transformation en un Centre de recherche conjoint de HEC Montréal, de l'École Polytechnique de Montréal et de l'Université McGill. En 1996, l'Université du Québec à Montréal s'est jointe à ces institutions pour parrainer le GERAD.

Le GERAD est un regroupement de chercheurs autour de la discipline de la recherche opérationnelle. Sa mission s'articule autour des objectifs complémentaires suivants :

- la contribution originale et experte dans tous les axes de recherche de ses champs de compétence ;
- la diffusion des résultats dans les plus grandes revues du domaine ainsi qu'auprès des différents publics qui forment l'environnement du Centre ;
- la formation d'étudiants des cycles supérieurs et de stagiaires post-doctoraux ;
- la contribution à la communauté économique à travers la résolution de problèmes et le développement de coffres d'outils transférables.

Les principaux axes de recherche du GERAD, en allant du plus théorique au plus appliqué, sont les suivants :

- le développement d'outils et de techniques d'analyse mathématiques de la recherche opérationnelle pour la résolution de problèmes complexes qui se posent dans les sciences de la gestion et du génie ;
- la confection d'algorithmes permettant la résolution efficace de ces problèmes ;
- l'application de ces outils à des problèmes posés dans des disciplines connexes à la recherche opérationnelle telles que la statistique, l'ingénierie financière, la théorie des jeux et l'intelligence artificielle ;
- l'application de ces outils à l'optimisation et à la planification de grands systèmes technico-économiques comme les systèmes énergétiques, les réseaux de télécommunication et de transport, la logistique et la distributique dans les industries manufacturières et de service ;

- l'intégration des résultats scientifiques dans des logiciels, des systèmes experts et dans des systèmes d'aide à la décision transférables à l'industrie.

Le fait marquant des célébrations du 25^e du GERAD est la publication de dix volumes couvrant les champs d'expertise du Centre. La liste suit : **Essays and Surveys in Global Optimization**, édité par C. Audet, P. Hansen et G. Savard ; **Graph Theory and Combinatorial Optimization**, édité par D. Avis, A. Hertz et O. Marcotte ; **Numerical Methods in Finance**, édité par H. Ben-Ameur et M. Breton ; **Analysis, Control and Optimization of Complex Dynamic Systems**, édité par E.K. Boukas et R. Malhamé ; **Column Generation**, édité par G. Desaulniers, J. Desrosiers et M.M. Solomon ; **Statistical Modeling and Analysis for Complex Data Problems**, édité par P. Duchesne et B. Rémillard ; **Performance Evaluation and Planning Methods for the Next Generation Internet**, édité par A. Girard, B. Sansò et F. Vázquez-Abad ; **Dynamic Games : Theory and Applications**, édité par A. Haurie et G. Zaccour ; **Logistics Systems : Design and Optimization**, édité par A. Langevin et D. Riopel ; **Energy and Environment**, édité par R. Loulou, J.-P. Waaub et G. Zaccour.

Je voudrais remercier très sincèrement les éditeurs de ces volumes, les nombreux auteurs qui ont très volontiers répondu à l'invitation des éditeurs à soumettre leurs travaux, et les évaluateurs pour leur bénévolat et ponctualité. Je voudrais aussi remercier Mmes Nicole Paradis, Francine Benoît et Louise Letendre ainsi que M. André Montpetit pour leur travail expert d'édition.

La place de premier plan qu'occupe le GERAD sur l'échiquier mondial est certes due à la passion qui anime ses chercheurs et ses étudiants, mais aussi au financement et à l'infrastructure disponibles. Je voudrais profiter de cette occasion pour remercier les organisations qui ont cru dès le départ au potentiel et à la valeur du GERAD et nous ont soutenus durant ces années. Il s'agit de HEC Montréal, l'École Polytechnique de Montréal, l'Université McGill, l'Université du Québec à Montréal et, bien sûr, le Conseil de recherche en sciences naturelles et en génie du Canada (CRSNG) et le Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

Georges Zaccour
Directeur du GERAD

Contents

Contributing Authors

Hatem Ben Amor
École Polytechnique and GERAD, Canada
Hatem.Ben.Amor@gerad.ca

Marielle Christiansen
Norwegian University of Science and Technology, Norway
marielle.christiansen@iot.ntnu.no

Emilie Danna
Georgia Institute of Technology, USA
emilie.danna@isye.gatech.edu

José Valério de Carvalho
Universidade do Minho, Portugal
vc@dps.uminho.pt

Guy Desaulniers
École Polytechnique and GERAD, Canada
Guy.Desaulniers@gerad.ca

Jacques Desrosiers
HEC Montréal and GERAD, Canada
jacques.desrosiers@hec.ca

Sylvie Gélinas
École Polytechnique and GERAD, Canada
sylvieg@crt.umontreal.ca

Han Hoogeveen
Utrecht University, The Netherlands
slam@cs.uu.nl

Dennis Huisman
Erasmus University Rotterdam, The Netherlands
huisman@few.eur.nl

Stefan Irnich
RWTH Aachen University, Germany
SIrnich@or.rwth-aachen.de

Raf Jans
Erasmus University Rotterdam, The Netherlands
rjans@fbk.eur.nl

Brian Kallehauge
Technical University of Denmark
bk@ctt.dtu.dk

Diego Klabjan
University of Illinois at Urbana-Champaign, USA
klabjan@uiuc.edu

Jesper Larsen
Technical University of Denmark, Denmark
jla@imm.dtu.dk

Claude Le Pape
ILOG S.A., France
clepape@ilog.fr

Marco E. Lübbecke
Technische Universität Berlin, Germany
m.luebbecke@math.tu-berlin.de

Oli B.G. Madsen
Technical University of Denmark, Denmark
ogm@ctt.dtu.dk

Bjørn Nygreen
Norwegian University of Science and Technology, Norway
bjorn.nygreen@iot.ntnu.no

Marc Peeters
Place de L'Université 16, Belgium
Marc.Peeters2@electrabel.com

David M. Ryan
University of Auckland, New Zealand
d.ryan@auckland.ac.nz

François Soumis
École Polytechnique and GERAD, Canada
francois.soumis@gerad.ca

MIKKEL M. SIGURD
University of Copenhagen, Denmark
sigurd@diku.dk

MARIUS M. SOLOMON
Northeastern University, USA and
GERAD, Canada
m.solomon@neu.edu

NINA L. ULSTEIN
Norwegian University of Science and
Technology, Norway
nina.ulstein@iot.ntnu.no

FRANÇOIS VANDERBECK
Université Bordeaux 1, France
fv@math.u-bordeaux.fr

STEEF VAN DE VELDE
Erasmus University, The Netherlands
s.velde@fac.fbk.eur.nl

MARJAN VAN DEN AKKER
Utrecht University, The Netherlands
marjan@cs.uu.nl

ALBERT P.M. WAGELMANS
Erasmus University Rotterdam,
The Netherlands
wagelmans@few.eur.nl

Preface

GERAD is an Operations Research center founded in 1980 that brings together the top universities in Montréal: HEC Montréal, École Polytechnique de Montréal, McGill University, and Université du Québec à Montréal. It is organized across several research teams. The Gencol team is one of the oldest and best known. Led by Jacques Desrosiers and François Soumis, it also includes Jean-François Cordeau, Guy Desaulniers, Michel Gamache, Odile Marcotte, Gilles Savard, and Marius M. Solomon. The late Martin Desrochers was the first Ph.D. student. The group originally focused on the vehicle routing problem with time windows and then expanded their focus to more complex resource constrained vehicle routing and crew scheduling problems.

During these 25 years the team's efforts resulted in important academic, scientific, commercial, and industrial benefits. The academic spin-off consists of support offered to scores of Ph.D. and master students, analysts, and post-doctoral and visiting researchers. The scientific advances include numerous publications, many in premier journals and widely used survey papers. Overall, the Gencol group made significant advances in the integer programming column generation area. It is with the excitement of having participated in these developments and the modesty of being only a small part of the research community that we welcome the advancements described in the chapters of this book.

The book starts with *A Primer in Column Generation* by Jacques Desrosiers and Marco E. Lübbecke. It introduces the column generation technique in integer programming settings. The relevant theory and the more advanced ideas necessary to solve large-scale practical problems are illustrated with a variety of examples.

In the second chapter, *Shortest Path Problems with Resource Constraints*, Stefan Irnich and Guy Desaulniers offer a comprehensive survey of the problems used to cast the subproblem in most vehicle routing and crew scheduling applications solved by column generation. The following two chapters are dedicated to the *Vehicle Routing Problem with Time Windows.* Brian Kallehauge, Jesper Larsen, Oli B.G. Madsen, and Marius M. Solomon focus on the methodological evolution, including cutting planes, parallelism, acceleration strategies for the master problem and novel subproblem approaches. Emilie Danna and Claude Le Pape, *Branch-and-Price Heuristics: A Case Study on the Vehicle Routing Problem with Time Windows*, illustrate the benefits of using hybrid branch-and-price and heuristic solutions to rapidly produce good integer solutions. This technique along with stabilization methods proposed to improve the efficiency of the column generation process are revisited in the next chapter by Hatem Ben Amor and José M. Valério de Carvalho in the context of *Cutting Stock Problems.* The authors also explore the links between the extended Dantzig-Wolfe decomposition and the Gilmore-Gomory model.

The following three chapters deal with air and maritime transportation applications. *Large-scale Models in the Airline Industry* are presented by Diego Klabjan. He examines models involved in strategic business processes as well as operational processes. The former address schedule design and fleeting, aircraft routing, and crew scheduling, while the latter models cope with irregular operations. Then, Marielle Christiansen and Bjørn Nygreen describe *Robust Inventory Ship Routing by Column Generation.* They consider an actual integrated ship scheduling and inventory management problem where the transporter has the responsibility to keep the inventory level of a single product at all plants within predetermined limits without causing production stopages due to missed transportation opportunities or variability in sailing time. Next, Mikkel M. Sigurd, Nina L. Ulstein, Bjørn Nygreen, and David M. Ryan discuss the design of a sea-transport system for Norwegian companies who rely heavily on maritime transportation in *Ship Scheduling With Recurring Visits and Visit Separation Requirements.* The model determines the optimal fleet composition, including the potential investment in new ships, the ship routes and their visit-schedules when transport tonnage is pooled over all participating companies.

The next three chapters deal with production environments. First, Dennis Huisman, Raf Jans, Marc Peeters, and Albert P.M. Wagelmans propose *Combining Column Generation and Lagrangian Relaxation.* The authors focus on using Lagrangian relaxation to either directly solve the LP relaxation of the Dantzig-Wolfe master problem or to generate new columns. They illustrate their ideas with an application in lot-sizing and comment on applications in other areas. Second, Sylvie Gélinas and François Soumis propose *Dantzig-Wolfe Decomposition for Job Shop Scheduling.* They present a flexible formulation capable of handling several objectives. In this context, each subproblem is a single machine sequencing problem with time windows. Third, Marjan van den Akker, Han Hoogeveen, and Steef van de Velde survey *Applying Column Generation to Machine Scheduling.* In particular, they illustrate the success of column generation methods when the main objective is to divide the jobs across the machines.

The final chapter by François Vanderbeck is on *Implementing Mixed Integer Column Generation.* He reviews how to set-up the Dantzig-Wolfe reformulation, adapt standard MIP techniques to the column generation context (branching, preprocessing, primal heuristics), and deal with specific column generation issues (initialization, stabilization, column management strategies).

We think that this book offers an insightful overview of the state-of-the-art in integer programming column generation and its many applications, and we certainly hope that it will serve as a good reference for the novice as well as the experienced column generation user.

Finally, we would like to thank all the contributors for their fine work.

GUY DESAULNIERS
JACQUES DESROSIERS
MARIUS M. SOLOMON

Chapter 1

A PRIMER IN COLUMN GENERATION

Jacques Desrosiers
Marco E. Lübbecke

Abstract We give a didactic introduction to the use of the column generation technique in linear and in particular in integer programming. We touch on both, the relevant basic theory and more advanced ideas which help in solving large scale practical problems. Our discussion includes embedding Dantzig-Wolfe decomposition and Lagrangian relaxation within a branch-and-bound framework, deriving natural branching and cutting rules by means of a so-called compact formulation, and understanding and influencing the behavior of the dual variables during column generation. Most concepts are illustrated via a small example. We close with a discussion of the classical cutting stock problem and some suggestions for further reading.

1. Hands-on experience

Let us start right away by solving a constrained shortest path problem. Consider the network depicted in Figure 1.1. Besides a cost c_{ij} there is a resource consumption t_{ij} attributed to each arc $(i,j) \in A$, say a traversal time. Our goal is to find a shortest path from node 1 to node 6 such that the total traversal time of the path does not exceed 14 time units.

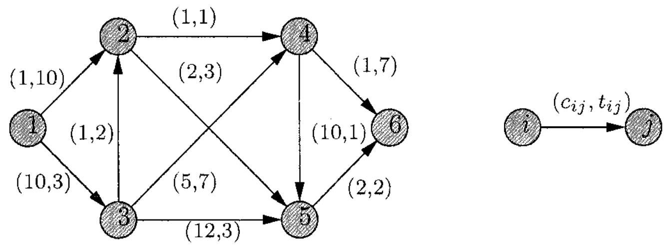

Figure 1.1. Time constrained shortest path problem, (p. 599 Ahuja et al., 1993).

One way to state this particular network flow problem is as the integer program (1.1)–(1.6). One unit of flow has to leave the source (1.2) and has to enter the sink (1.4), while flow conservation (1.3) holds at all other nodes. The time resource constraint appears as (1.5).

$$z^\star := \min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1.1}$$

$$\text{subject to} \quad \sum_{j:\,(1,j)\in A} x_{1j} = 1 \tag{1.2}$$

$$\sum_{j:\,(i,j)\in A} x_{ij} - \sum_{j:\,(j,i)\in A} x_{ji} = 0 \quad i = 2,3,4,5 \tag{1.3}$$

$$\sum_{i:\,(i,6)\in A} x_{i6} = 1 \tag{1.4}$$

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \leq 14 \tag{1.5}$$

$$x_{ij} = 0 \text{ or } 1 \quad (i,j) \in A \tag{1.6}$$

An inspection shows that there are nine possible paths, three of which consume too much time. The optimal integer solution is path 13246 of cost 13 with a traversal time of 13. How would we find this out? First note that the resource constraint (1.5) prevents us from solving our problem with a classical shortest path algorithm. In fact, no polynomial time algorithm is likely to exist since the resource constrained shortest path problem is $\mathcal{NP}$-hard. However, since the problem is *almost* a shortest path problem, we would like to exploit this embedded well-studied structure algorithmically.

1.1 An equivalent reformulation: Arcs vs. paths

If we ignore the complicating constraint (1.5), the easily tractable remainder is $X = \{x_{ij} = 0 \text{ or } 1 \mid (1.2)\text{–}(1.4)\}$. It is a well-known result in network flow theory that an extreme point $\mathbf{x}_p = (x_{pij})$ of the polytope defined by the convex hull of X corresponds to a path $p \in P$ in the network. This enables us to express any arc flow as a convex combination of path flows:

$$x_{ij} = \sum_{p\in P} x_{pij}\lambda_p \quad (i,j) \in A \tag{1.7}$$

$$\sum_{p\in P} \lambda_p = 1 \tag{1.8}$$

$$\lambda_p \geq 0 \quad p \in P. \tag{1.9}$$

If we substitute for $\mathbf{x}$ in (1.1) and (1.5) we obtain the so-called *master problem*:

$$z^{\star} = \min \sum_{p \in P} \left(\sum_{(i,j) \in A} c_{ij} x_{pij} \right) \lambda_p \tag{1.10}$$

$$\text{subject to} \sum_{p \in P} \left(\sum_{(i,j) \in A} t_{ij} x_{pij} \right) \lambda_p \leq 14 \tag{1.11}$$

$$\sum_{p \in P} \lambda_p = 1 \tag{1.12}$$

$$\lambda_p \geq 0 \quad p \in P \tag{1.13}$$

$$\sum_{p \in P} x_{pij} \lambda_p = x_{ij} \quad (i,j) \in A \tag{1.14}$$

$$x_{ij} = 0 \text{ or } 1 \quad (i,j) \in A. \tag{1.15}$$

Loosely speaking, the structural information X that we are looking for a path is hidden in "$p \in P$." The cost coefficient of λ_p is the cost of path p and its coefficient in (1.11) is path p's duration. Via (1.14) and (1.15) we explicitly preserve the linking of variables (1.7) in the formulation, and we may recover a solution $\mathbf{x}$ to our *original* problem (1.1)–(1.6) from a master problem's solution. Always remember that integrality must hold for the original $\mathbf{x}$ variables.

1.2 The linear relaxation of the master problem

One starts with solving the linear programming (LP) relaxation of the master problem. If we relax (1.15), there is no longer a need to link the $\mathbf{x}$ and $\boldsymbol{\lambda}$ variables, and we may drop (1.14) as well. There remains a problem with nine path variables and two constraints. Associate with (1.11) and (1.12) dual variables π_1 and π_0, respectively. For large networks, the cardinality of P becomes prohibitive, and we cannot even explicitly state all the variables of the master problem. The appealing idea of *column generation* is to work only with a sufficiently meaningful subset of variables, forming the so-called *restricted master problem* (RMP). More variables are added only when needed: Like in the simplex method we have to find in every iteration a promising variable to enter the basis. In column generation an iteration consists

a) of optimizing the restricted master problem in order to determine the current optimal objective function value $\bar{z}$ and dual multipliers $\boldsymbol{\pi}$, and

Table 1.1. BB0: The linear programming relaxation of the master problem

Iteration	Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
BB0.1	$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
BB0.2	$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
BB0.3	$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10	-4.8	13256	15	10
BB0.4	$\lambda_{1246} = \lambda_{13256} = 0.5$	9.0	30.00	-1.50	-2.5	1256	5	15
BB0.5	$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	7.0	35.00	-2.00	0			
Arc flows: $x_{12} = 0.8, x_{13} = x_{32} = 0.2, x_{25} = x_{56} = 1$								

b) of finding, if there still is one, a variable λ_p with *negative reduced cost*

$$\bar{c}_p = \sum_{(i,j)\in A} c_{ij}x_{pij} - \pi_1 \left(\sum_{(i,j)\in A} t_{ij}x_{pij} \right) - \pi_0 < 0. \tag{1.16}$$

The implicit search for a minimum reduced cost variable amounts to optimizing a *subproblem*, precisely in our case: A shortest path problem in the network of Figure 1.1 with a modified cost structure:

$$\bar{c}^\star = \min_{(1.2)-(1.4),(1.6)} \sum_{(i,j)\in A} (c_{ij} - \pi_1 t_{ij})x_{ij} - \pi_0. \tag{1.17}$$

Clearly, if $\bar{c}^\star \geq 0$ there is no improving variable and we are done with the linear relaxation of the master problem. Otherwise, the variable found is added to the RMP and we repeat.

In order to obtain integer solutions to our original problem, we have to embed column generation within a branch-and-bound framework. We now give full numerical details of the solution of our particular instance. We denote by BB$n.i$ iteration number i at node number n ($n = 0$ represents the root node). The summary in Table 1.1 for the LP relaxation of the master problem also lists the cost c_p and duration t_p of path p, respectively, and the solution in terms of the value of the original variables $\mathbf{x}$.

Since we have no feasible initial solution at iteration BB0.1, we adopt a big-M approach and introduce an artificial variable y_0 with a large cost, say 100, for the convexity constraint. We do not have any path variables yet and the RMP contains two constraints and the artificial variable. This problem is solved by inspection: $y_0 = 1, \bar{z} = 100$, and the dual variables are $\pi_0 = 100$ and $\pi_1 = 0$. The subproblem (1.17) returns path 1246 at reduced cost $\bar{c}^\star = -97$, cost 3 and duration 18. In iteration BB0.2, the RMP contains two variables: y_0 and λ_{1246}. An optimal solution with $\bar{z} = 24.6$ is $y_0 = 0.22$ and $\lambda_{1246} = 0.78$, which is still infeasible. The dual variables assume values $\pi_0 = 100$ and $\pi_1 = -5.39$. Solving the subproblem gives the feasible path 1356 of reduced cost -32.9, cost 24, and duration 8.

In total, four path variables are generated during the column generation process. In iteration BB0.5, we use 0.2 times the feasible path 13256 and 0.8 times the infeasible path 1256. The optimal objective function value is 7, with $\pi_0 = 35$ and $\pi_1 = -2$. The arc flow values provided at the bottom of Table 1.1 are identical to those found when solving the LP relaxation of the original problem.

1.3 Branch-and-bound: The reformulation repeats

Except for the integrality requirement (1.6) (or 1.15) all constraints of the original (and of the master) problem are satisfied, and a subsequent branch-and-bound process is used to compute an optimal integer solution. Even though it cannot happen for our example problem, in general the generated set of columns may not contain an integer feasible solution. To proceed, we have to start the reformulation and column generation again in each node.

Let us first explore some "standard" ways of branching on fractional variables, e.g., branching on $x_{12} = 0.8$. For $x_{12} = 0$, the impact on the RMP is that we have to remove path variables λ_{1246} and λ_{1256}, that is, those paths which contain arc $(1,2)$. In the subproblem, this arc is removed from the network. When the RMP is re-optimized, the artificial variable assumes a positive value, and we would have to generate new $\boldsymbol{\lambda}$ variables. On branch $x_{12} = 1$, arcs $(1,3)$ and $(3,2)$ cannot be used. Generated paths which contain these arcs are discarded from the RMP, and both arcs are removed from the subproblem.

There are also many strategies involving more than a single arc flow variable. One is to branch on the sum of all flow variables which currently is 3.2. Since the solution is a path, an integer number of arcs has to be used, in fact, at least three and at most five in our example. Our freedom of making branching decisions is a powerful tool when properly applied.

Alternatively, we branch on $x_{13} + x_{32} = 0.4$. On branch $x_{13} + x_{32} = 0$, we simultaneously treat two flow variables; impacts on the RMP and the subproblem are similar to those described above. On branch $x_{13} + x_{32} \geq 1$, this constraint is first added to the original formulation. We exploit again the path substructure X, go through the reformulation process via (1.7), and obtain a new RMP to work with. Details of the search tree are summarized in Table 1.2.

Table 1.2. Details of the branch-and-bound decisions

Iteration	Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
BB1: BB0 and $x_{13} + x_{32} = 0$									
BB1.1	$y_0 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33	–	0			
BB1.2	Big-M increased to 1000								
	$y_0 = 0.067, \lambda_{1256} = 0.933$	71.3	1000	-66.33	–	-57.3	12456	14	14
BB1.3	$\lambda_{12456} = 1$	**14**	1000	-70.43	–	0			
BB2: BB0 and $x_{13} + x_{32} \geq 1$									
BB2.1	$\lambda_{1246} = \lambda_{13256} = 0.5$	9	15	-0.67	3.33	0			
Arc flows:	$x_{12} = x_{13} = x_{24} = x_{25} = x_{32} = x_{46} = x_{56} = 0.5$								
BB3: BB2 and $x_{12} = 0$									
BB3.1	$\lambda_{13256} = 1$	**15**	15	0	0	-2	13246	13	13
BB3.2	$\lambda_{13246} = 1$	**13**	13	0	0	0			
BB4: BB2 and $x_{12} = 1$									
BB4.1	$y_0 = 0.067, \lambda_{1256} = 0.933$	111.3	100	-6.33	100	0			
Infeasible arc flows									

At node BB1, we set $x_{13} + x_{32} = 0$. In iteration BB1.1, paths 1356 and 13256 are discarded from the RMP, and arcs $(1,3)$ and $(3,2)$ are removed from the subproblem. The resulting RMP with $y_0 = 0.067$ and $\lambda_{1256} = 0.933$ is infeasible. The objective function assumes a value $\bar{z} = 11.3$, and $\pi_0 = 100$ and $\pi_1 = -6.33$. Given these dual multipliers, no column with negative reduced cost can be generated!

Here we face a drawback of the big-M approach. Path 12456 is feasible, its duration is 14, but its cost of 14 is larger than the current objective function value, computed as $0.067M + 0.933 \times 5$. The constant $M = 100$ is too small, and we have to increase it, say to 1000. (A different phase I approach, that is, minimizing the artificial variable y_0, would have easily prevented this.) Re-optimizing the RMP in iteration BB1.2 now results in $\bar{z} = 71.3$, $y_0 = 0.067$, $\lambda_{1256} = 0.933$, $\pi_0 = 1000$, and $\pi_1 = -66.33$. The subproblem returns path 12456 with a reduced cost of -57.3. In iteration BB1.3, the new RMP has an integer solution $\lambda_{12456} = 1$, with $\bar{z} = 14$, an upper bound on the optimal path cost. The dual multipliers are $\pi_0 = 1000$ and $\pi_1 = -70.43$, and no new variable is generated.

At node BB2, we impose $x_{13} + x_{32} \geq 1$ to the original formulation, and again, we reformulate these $\mathbf{x}$ variables in terms of the $\boldsymbol{\lambda}$ variables. The resulting new constraint (with dual multiplier π_2) in the RMP is $\sum_{p \in P}(x_{p13} + x_{p32})\lambda_p \geq 1$.From the value of $(x_{p13} + x_{p32})$ we learn how often arcs $(1,3)$ and $(3,2)$ are used in path p. The current problem at node BB2.1 is the following:

$$
\begin{array}{llllllll}
\min & 100y_0 + & 3\lambda_{1246} + & 24\lambda_{1356} + & 15\lambda_{13256} + & 5\lambda_{1256} & & \\
\text{subject to:} & & 18\lambda_{1246} + & 8\lambda_{1356} + & 10\lambda_{13256} + & 15\lambda_{1256} & \leq 14 & [\pi_1] \\
 & & & \lambda_{1356} + & 2\lambda_{13256} & & \geq 1 & [\pi_2] \\
 & y_0 + & \lambda_{1246} + & \lambda_{1356} + & \lambda_{13256} + & \lambda_{1256} & = 1 & [\pi_0] \\
 & & & y_0, & \lambda_{1246}, \lambda_{1356}, & \lambda_{13256}, \lambda_{1256} & \geq 0 &
\end{array}
$$

From solving this linear program we obtain an increase in the objective function $\bar{z}$ from 7 to 9 with variables $\lambda_{1246} = \lambda_{13256} = 0.5$, and dual multipliers $\pi_0 = 15, \pi_1 = -0.67$, and $\pi_2 = 3.33$. The new subproblem is given by

$$\bar{c}^\star = \min_{(1.2)-(1.4),\,(1.6)} \sum_{(i,j)\in A} (c_{ij} - \pi_1 t_{ij})x_{ij} - \pi_0 - \pi_2(x_{13} + x_{32}). \quad (1.18)$$

For these multipliers no path of negative reduced cost exists. The solution of the flow variables is $x_{12} = x_{13} = x_{24} = x_{25} = x_{32} = x_{46} = x_{56} = 0.5$.

Next, we arbitrarily choose variable $x_{12} = 0.5$ to branch on. Two iterations are needed when x_{12} is set to zero. In iteration BB3.1, path variables λ_{1246} and λ_{1256} are discarded from the RMP and arc $(1,2)$ is removed from the subproblem. The RMP is integer feasible with $\lambda_{13256} = 1$ at cost 15. Dual multipliers are $\pi_0 = 15, \pi_1 = 0$, and $\pi_2 = 0$. Path 13246 of reduced cost -2, cost 13 and duration 13 is generated and used in the next iteration BB3.2. Again the RMP is integer feasible with path variable $\lambda_{13246} = 1$ and a new best integer solution at cost 13, with dual multipliers $\pi_0 = 15, \pi_1 = 0$, and $\pi_2 = 0$ for which no path of negative reduced cost exists.

On the alternative branch $x_{12} = 1$ the RMP is optimal after a single iteration. In iteration BB4.1, variable x_{13} can be set to zero and variables λ_{1356}, λ_{13256}, and λ_{13246} are discarded from the current RMP. After the introduction of an artificial variable y_2 in the second row, the RMP is infeasible since $y_0 > 0$ (as can be seen also from the large objective function value $\bar{z} = 111.3$). Given the dual multipliers, no columns of negative reduced cost can be generated, and the RMP remains infeasible. The optimal solution (found at node BB3) is path 13246 of cost 13 with a duration of 13 as well.

2. Some theoretical background

In the previous example we already saw all the necessary building blocks for a column generation based solution approach to integer programs: (1) an original formulation to solve which acts as the control center to facilitate the design of natural branching rules and cutting planes; (2) a master problem to determine the currently optimal dual multipliers and to provide a lower bound at each node of the branch-and-bound tree; (3) a pricing subproblem which explicitly reflects an embedded structure we wish to exploit. In this section we detail the underlying theory.

2.1 Column generation

Let us call the following linear program the *master problem* (MP).

$$z^\star_{MP} := \min \sum_{j \in J} c_j \lambda_j \tag{1.19}$$

$$\text{subject to} \sum_{j \in J} \mathbf{a}_j \lambda_j \geq \mathbf{b} \tag{1.20}$$

$$\lambda_j \geq 0, \quad j \in J. \tag{1.21}$$

In each iteration of the simplex method we look for a non-basic variable to price out and enter the basis. That is, given the non-negative vector $\boldsymbol{\pi}$ of dual variables we wish to find a $j \in J$ which minimizes $\bar{c}_j := c_j - \boldsymbol{\pi}^t \mathbf{a}_j$. This explicit pricing is a too costly operation when $|J|$ is huge. Instead, we work with a reasonably small subset $J' \subseteq J$ of columns—the restricted master problem (RMP)—and evaluate reduced costs only by implicit enumeration. Let $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$ assume primal and dual optimal solutions of the current RMP, respectively. When columns $\mathbf{a}_j$, $j \in J$, are given as elements of a set $\mathcal{A}$, and the cost coefficient c_j can be computed from $\mathbf{a}_j$ via a function c then the subproblem

$$\bar{c}^\star := \min\{c(\mathbf{a}) - \boldsymbol{\pi}^t \mathbf{a} \mid \mathbf{a} \in \mathcal{A}\} \tag{1.22}$$

performs the pricing. If $\bar{c}^\star \geq 0$, there is no negative $\bar{c}_j$, $j \in J$, and the solution $\boldsymbol{\lambda}$ to the restricted master problem optimally solves the master problem as well. Otherwise, we add to the RMP the column derived from the optimal subproblem solution, and repeat with re-optimizing the RMP. The process is initialized with an artificial, a heuristic, or a previous ("warm start") solution. In what regards convergence, note that each $\mathbf{a} \in \mathcal{A}$ is generated at most once since no variable in an optimal RMP has negative reduced cost. When dealing with some finite set $\mathcal{A}$ (as is practically always true), the column generation algorithm is exact. In addition, we can make use of bounds. Let $\bar{z}$ denote the optimal objective function value to the RMP. When an upper bound $\kappa \geq \sum_{j \in J} \lambda_j$ holds for the optimal solution of the master problem, we have not only an upper bound $\bar{z}$ on $z^\star_{MP}$ in each iteration, but also a lower bound: we cannot reduce $\bar{z}$ by more than κ times the smallest reduced cost $\bar{c}^\star$:

$$\bar{z} + \kappa \bar{c}^\star \leq z^\star_{MP} \leq \bar{z}. \tag{1.23}$$

Thus, we may verify the solution quality at any time. In the optimum of (1.19), $\bar{c}^\star = 0$ for the basic variables, and $\bar{z} = z^\star_{MP}$.

2.2 Dantzig-Wolfe decomposition for integer programs

In many applications we are interested in optimizing over a discrete set X. For $X = \{\mathbf{x} \in \mathbb{Z}^n_+ \mid D\mathbf{x} \geq \mathbf{d}\} \neq \varnothing$ we have the special case of integer linear programming. Consider the following (*original* or *compact*) program:

$$z^\star := \min \mathbf{c}^t \mathbf{x} \tag{1.24}$$

$$\text{subject to } A\mathbf{x} \geq \mathbf{b} \tag{1.25}$$

$$\mathbf{x} \in X. \tag{1.26}$$

Replacing X by $\operatorname{conv}(X)$ in (1.24) does not change $z^\star$ which we assume to be finite. The Minkowski and Weyl theorems (see Schrijver, 1986) enable us to represent each $\mathbf{x} \in X$ as a convex combination of extreme points $\{\mathbf{x}_p\}_{p \in P}$ plus a non-negative combination of extreme rays $\{\mathbf{x}_r\}_{r \in R}$ of $\operatorname{conv}(X)$, i.e.,

$$\mathbf{x} = \sum_{p \in P} \mathbf{x}_p \lambda_p + \sum_{r \in R} \mathbf{x}_r \lambda_r, \quad \sum_{p \in P} \lambda_p = 1, \qquad \boldsymbol{\lambda} \in \mathbb{R}^{|P|+|R|}_+ \tag{1.27}$$

where the index sets P and R are finite. Substituting for $\mathbf{x}$ in (1.24) and applying the linear transformations $c_j = \mathbf{c}^t \mathbf{x}_j$ and $\mathbf{a}_j = A\mathbf{x}_j$, $j \in P \cup R$ we obtain an equivalent *extensive formulation*

$$z^\star := \min \sum_{p \in P} c_p \lambda_p + \sum_{r \in R} c_r \lambda_r \tag{1.28}$$

$$\text{subject to } \sum_{p \in P} \mathbf{a}_p \lambda_p + \sum_{r \in R} \mathbf{a}_r \lambda_r \geq \mathbf{b} \tag{1.29}$$

$$\sum_{p \in P} \lambda_p = 1 \tag{1.30}$$

$$\boldsymbol{\lambda} \geq \mathbf{0} \tag{1.31}$$

$$\sum_{p \in P} \mathbf{x}_p \lambda_p + \sum_{r \in R} \mathbf{x}_r \lambda_r = \mathbf{x} \tag{1.32}$$

$$\mathbf{x} \in \mathbb{Z}^n_+. \tag{1.33}$$

Equation (1.30) is referred to as the *convexity constraint.* When we relax the integrality of $\mathbf{x}$, there is no need to link $\mathbf{x}$ and $\boldsymbol{\lambda}$, and we may also relax (1.32). The columns of this special master problem are defined by the extreme points and extreme rays of $\operatorname{conv}(X)$. We solve the master by column generation to get its optimal objective function value $z^\star_{MP}$. Given an optimal dual solution $\boldsymbol{\pi}$ and π_0 to the current RMP, where variable π_0 corresponds to the convexity constraint, the subproblem is to determine $\min_{j\in P}\{c_j - \boldsymbol{\pi}^t\mathbf{a}_j - \pi_0\}$ and $\min_{j\in R}\{c_j - \boldsymbol{\pi}^t\mathbf{a}_j\}$. By our previous linear transformation and since π_0 is a constant, this results in

$$\bar{c}^\star := \min\{(\mathbf{c}^t - \boldsymbol{\pi}^t A)\mathbf{x} - \pi_0 \mid \mathbf{x} \in X\}. \tag{1.34}$$

This subproblem is an integer linear program. When $\bar{c}^\star \geq 0$, there is no negative reduced cost column, and the algorithm terminates. When $\bar{c}^\star < 0$ and finite, an optimal solution to (1.34) is an extreme point $\mathbf{x}_p$ of $\operatorname{conv}(X)$, and we add the column $[\mathbf{c}^t\mathbf{x}_p, (A\mathbf{x}_p)^t, 1]^t$ to the RMP. When $\bar{c}^\star = -\infty$ we identify an extreme ray $\mathbf{x}_r$ of $\operatorname{conv}(X)$ as a solution $\mathbf{x} \in X$ to $(\mathbf{c}^t - \boldsymbol{\pi}^t A)\mathbf{x} = 0$, and add the column $[\mathbf{c}^t\mathbf{x}_r, (A\mathbf{x}_r)^t, 0]^t$ to the RMP.

From (1.23) together with the convexity constraint we obtain in each iteration

$$\bar{z} + \bar{c}^\star \leq z^\star_{MP} \leq \bar{z}, \tag{1.35}$$

where $\bar{z} = \boldsymbol{\pi}^t\mathbf{b} + \pi_0$ is again the optimal objective function value of the RMP. Since $z^\star_{MP} \leq z^\star$, $\bar{z} + \bar{c}^\star$ is also a lower bound on $z^\star$. In general, $\bar{z}$ is not a valid upper bound on $z^\star$, except if the current $\mathbf{x}$ variables are integer. The algorithm is exact and finite as long as finiteness is ensured in optimizing the RMP.

The original formulation is the starting point to obtain integer solutions in the $\mathbf{x}$ variables. Branching and cutting constraints are added there, the reformulation as in Section 1.1 is re-applied, and the process continues with an updated master problem. It is important to see that it is our choice as to whether the additional constraints remain in the master problem (as in the previous section) or go into the subproblem (as we will see later).

Pricing out the original x variables. Assume that in (1.24) we have a linear subproblem $X = \{\mathbf{x} \in \mathbb{R}^n_+ \mid D\mathbf{x} \geq \mathbf{d}\} \neq \varnothing$. Column generation then essentially solves the linear program

$$\min \mathbf{c}^t\mathbf{x} \quad \text{subject to } A\mathbf{x} \geq \mathbf{b}, \quad D\mathbf{x} \geq \mathbf{d}, \qquad \mathbf{x} \geq \mathbf{0}.$$

We obtain an optimal primal solution $\mathbf{x}$ but only the dual multipliers $\boldsymbol{\pi}$ associated with the constraint set $A\mathbf{x} \geq \mathbf{b}$. However, following an idea of Walker (1969) we can also retrieve the dual variables $\boldsymbol{\sigma}$ associated with

$D\mathbf{x} \geq d$: It is the vector obtained from solving the linear subproblem in the last iteration of the column generation process. This full dual information allows for a pricing of the original variables, and therefore a possible elimination of some of them. Given an upper bound on the integer optimal objective function value of the original problem, one can eliminate an x variable if its reduced cost is larger than the optimality gap.

In the general case of a linear integer or even non-linear pricing subproblem, the above procedure does not work. Poggi de Aragão and Uchoa (2003) suggest to directly use the extensive formulation: If we keep the coupling constraint (1.32) in the master problem, it suffices to impose the constraint $\mathbf{x} \geq \boldsymbol{\epsilon}$, for a small $\boldsymbol{\epsilon} > \mathbf{0}$, at the end of the process. The shadow prices of these constraints are the reduced costs of the $\mathbf{x}$ vector of original variables. Note that there is no need to apply the additional constraints to already positive variables. Computational experiments underline the benefits of this procedure.

Block diagonal structure. For practical problems Dantzig-Wolfe decomposition can typically exploit a block diagonal structure of D, i.e.,

$$D = \begin{pmatrix} D^1 & & & \\ & D^2 & & \\ & & \ddots & \\ & & & D^\kappa \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}^1 \\ \mathbf{d}^2 \\ \vdots \\ \mathbf{d}^\kappa \end{pmatrix}. \tag{1.36}$$

Each $X^k = \{D^k\mathbf{x}^k \geq \mathbf{d}^k, \mathbf{x}^k \geq \mathbf{0} \text{ and integer}\}$, $k \in K := \{1, \ldots, \kappa\}$, gives rise to a representation as in (1.27). The decomposition yields κ subproblems, each with its own convexity constraint and associated dual variable:

$$\bar{c}^{k\star} := \min\{(\mathbf{c}^{kT} - \boldsymbol{\pi}^t A^k)\mathbf{x}^k - \pi_0^k \mid \mathbf{x}^k \in X^k\}, \quad k \in K. \tag{1.37}$$

The superscript k to all entities should be interpreted in the canonical way. The algorithm terminates when $\bar{c}^{k\star} \geq 0$, for all $k \in K$. Otherwise, extreme points and rays identified in (1.37) give rise to new columns to be added to the RMP. By linear programming duality, $\bar{z} = \boldsymbol{\pi}^t\mathbf{b} + \sum_{k=1}^{\kappa} \pi_0^k$, and we obtain the following bounds, see Lasdon (1970):

$$\bar{z} + \sum_{k=1}^{\kappa} \bar{c}^{k\star} \leq z_{MP}^\star \leq \bar{z}. \tag{1.38}$$

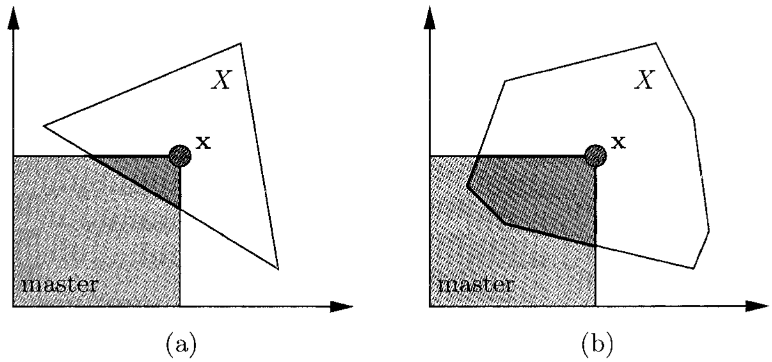

Figure 1.2. Schematic illustration of the domains of master and subproblem X.

2.3 Useful working knowledge

When problems get larger and computationally much more difficult than our small constrained shortest path problem it is helpful to know more about mechanisms, their consequences, and how to exploit them.

Infeasible paths. One may wonder why we kept infeasible paths in the RMP during column generation. Here, as for the whole process, we cannot overemphasize the fact that knowledge about the *integer* solution usually does not help us much in solving the *linear relaxation* program. Figure 1.2 illustrates the domain of the RMP (shaded) and the domain X of the subproblem. In part a), the optimal solution $\mathbf{x}$, symbolized by the dot, is uniquely determined as a convex combination of the three extreme points of the triangle X, even though all of them are not feasible for the intersection of the master and subproblem. In our example, in iteration BB0.5, any convex combination of feasible paths which have been generated, namely 13256 and 1356, has cost larger than 7, i.e., is suboptimal for the linear relaxation of the master problem. Infeasible paths are removed only if needed during the search for an integer solution. In Figure 1.2 (a), $\mathbf{x}$ can be integer and no branch-and-bound search is needed.

In part b) there are many ways to express the optimal solution as a convex combination of three extreme points. This is a partial explanation of the slow convergence (*tailing off*) of linear programming column generation.

Lower and upper bounds. Figure 1.3 gives the development of upper ($\bar{z}$) and lower ($\bar{z} + \bar{c}^\star$) bounds on $z^\star_{MP}$ in the root node for our

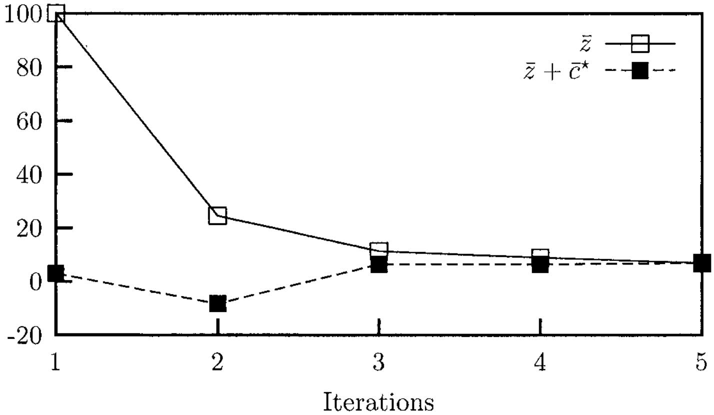

Figure 1.3. Development of lower and upper bounds on z_{MP} in BB0.

Table 1.3. Multipliers σ_i for the flow conservation constraints on nodes $i \in N$.

node i	1	2	3	4	5	6
σ_i	29	8	13	5	0	-6

small constrained shortest path example. The values for the lower bound are 3.0, -8.33, 6.6, 6.5, and finally 7. While the upper bound decreases monotonically (as expected when minimizing a linear program) there is no monotony for the lower bound. Still, we can use these bounds to evaluate the quality of the current solution by computing the optimality gap, and could stop the process when a preset quality is reached. Is there any use of the bounds beyond that?

Note first that $UB = \bar{z}$ is not an upper bound on $z^\star$. The currently (over all iterations) best lower bound LB, however, is a lower bound on $z^\star_{MP}$ and on $z^\star$. Even though there is no direct use of LB or UB in the master problem we can impose the additional constraints $LB \leq \mathbf{c}^t\mathbf{x} \leq UB$ to the subproblem structure X if the subproblem consists of a single block. Be aware that this cutting modifies the subproblem structure X, with all algorithmic consequences, that is, possible complications for a combinatorial algorithm. In our constrained shortest path example, two generated paths are feasible and provide upper bounds on the optimal integer solution z^*. The best one is path 13256 of cost 15 and duration 10. Table 1.3 shows the multipliers σ_i, $i = 1, \ldots, 6$ for the flow conservation constraints of the path structure X at the last iteration of the column generation process.

Therefore, given the optimal multiplier $\pi_1 = -2$ for the resource constraint, the reduced cost of an arc is given by $\bar{c}_{ij} = c_{ij} - \sigma_i + \sigma_j - t_{ij}\pi_1$, $(i,j) \in A$. The reader can verify that $\bar{c}_{34} = 3 - 13 + 5 - (7)(-2) = 11$. This is the only reduced cost which exceeds the current optimality gap which equals to $15 - 7 = 8$. Arc $(3,4)$ can be permanently discarded from the network and paths 1346 and 13456 will never be generated.

Integrality property. Solving the subproblem as an integer program usually helps in closing part of the integrality gap of the master problem (Geoffrion, 1974), except when the subproblem possesses the *integrality property.* This property means that solutions to the pricing problem are naturally integer when it is solved as a linear program. This is the case for our shortest path subproblem and this is why we obtained the value of the linear relaxation of the original problem as the value of the linear relaxation of the master problem.

When looking for an integer solution to the original problem, we need to impose new restrictions on (1.1)–(1.6). One way is to take advantage of a new X structure. However, if the new subproblem is still solved as a linear program, $z^\star_{MP}$ remains 7. Only solving the new X structure as an integer program may improve $z^\star_{MP}$.

Once we understand that we can modify the subproblem structure, we can devise other decomposition strategies. One is to define the X structure as

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \leq 14, \quad x_{ij} \text{ binary}, \qquad (i,j) \in A \tag{1.39}$$

so that the subproblem becomes a knapsack problem which does not possess the integrality property. Unfortunately, in this example, $z^\star_{MP}$ remains 7. However, improvements can be obtained by imposing more and more constraints to the subproblem. An example is to additionally enforce the selection of one arc to leave the source (1.2) and another one to enter the sink (1.4), and impose constraint $3 \leq \sum_{(i,j)\in A} x_{ij} \leq 5$ on the minimum and maximum number of selected arcs. Richer subproblems, as long as they can be solved efficiently and do not possess the integrality property, may help in closing the integrality gap.

It is also our decision how much branching and cutting information (ranging from none to all) we put into the subproblem. This choice depends on where the additional constraints are more easily accommodated in terms of algorithms and computational tractability. Branching decisions imposed on the subproblem can reduce its solution space and may turn out to facilitate a solution as integer program. As an illus-

Table 1.4. Lower bound cut added to the subproblem at the end of the root node.

Iteration	Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p	UB	LB
BB0.6	$\lambda_{13256} = 1$	**15**	15	0	-2	13246	13	13	15	13
BB0.7	$\lambda_{13246} = 1$	**13**	13	0					13	13

tration we describe adding the lower bound cut in the root node of our small example.

Imposing the lower bound cut $\mathbf{c}^t\mathbf{x} \geq 7$. Assume that we have solved the relaxed RMP in the root node and instead of branching, we impose the lower bound cut on the X structure, see Table 1.4. Note that this cut would not have helped in the RMP since $\bar{z} = \mathbf{c}^t\mathbf{x} = 7$ already holds. We start modifying the RMP by removing variables λ_{1246} and λ_{1256} as their cost is smaller than 7. In iteration BB0.6, for the re-optimized RMP $\lambda_{13256} = 1$ is optimal at cost 15; it corresponds to a feasible path of duration 10. UB is updated to 15 and the dual multipliers are $\pi_0 = 15$ and $\pi_1 = 0$. The X structure is modified by adding constraint $\mathbf{c}^t\mathbf{x} \geq 7$. Path 13246 is generated with reduced cost -2, cost 13, and duration 13. The new lower bound is $15-2 = 13$. On the downside of this significant improvement is the fact that we have destroyed the pure network structure of the subproblem which we have to solve as an integer program now. We may pass along with this circumstance if it pays back a better bound.

We re-optimize the RMP in iteration BB0.7 with the added variable λ_{13246}. This variable is optimal at value 1 with cost and duration equal to 13. Since this variable corresponds to a feasible path, it induces a better upper bound which is equal to the lower bound: Optimality is proven. There is no need to solve the subproblem.

Note that using the dynamically adapted lower bound cut right from the start has an impact on the solution process. For example, the first generated path 1246 would be eliminated in iteration BB0.3 since the lower bound reaches 6.6, and path 1256 is never generated. Additionally adding the upper bound cut has a similar effect.

Acceleration strategies. Often acceleration techniques are key elements for the viability of the column generation approach. Without them, it would have been almost impossible to obtain quality solutions to various applications, in a reasonable amount of computation time. We sketch here only some strategies, see e.g., Desaulniers et al. (2001) for much more.