A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems

Nonconvex Optimization and Its Applications

Volume 31

The titles published in this series are listed at the end of this volume.

A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems

by

Hanif D. Sherali

Department of Industrial and Systems Engineering,
Virginia Polytechnic Institute and State University,
Blacksburg, Virginia, U.S.A.

and

Warren P. Adams

Department of Mathematical Sciences,
Clemson University,
Clemson, North Carolina, U.S.A.

SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-1-4419-4808-3 ISBN 978-1-4757-4388-3 (eBook)
DOI 10.1007/978-1-4757-4388-3

Printed on acid-free paper

Dedicated to:

Our Parents

and to

Azeem, Stephanie, and Cristina:

The Next Generation.

CONTENTS

PREFACE

This book deals with the theory and applications of the **Reformulation-Linearization/Convexification Technique (RLT)** for solving nonconvex optimization problems. A unified treatment of discrete and continuous nonconvex programming problems is presented using this approach. In essence, the bridge between these two types of nonconvexities is made via a polynomial representation of discrete constraints. For example, the binariness on a 0-1 variable x_j can be equivalently expressed as the polynomial constraint $x_j(1 - x_j) = 0$. The motivation for this book is the role of tight linear/convex programming representations or relaxations in solving such discrete and continuous nonconvex programming problems. The principal thrust is to commence with a model that affords a useful representation and structure, and then to further strengthen this representation through automatic reformulation and constraint generation techniques.

As mentioned above, the focal point of this book is the development and application of RLT for use as an automatic reformulation procedure, and also, to generate strong valid inequalities. The RLT operates in two phases. In the *Reformulation Phase*, certain types of additional implied polynomial constraints, that include the aforementioned constraints in the case of binary variables, are appended to the problem. The resulting problem is subsequently linearized, except that certain convex constraints are sometimes retained in particular special cases, in the *Linearization/Convexification Phase*. This is done via the definition of suitable new variables to replace each distinct variable-product term. The higher dimensional representation yields a linear (or convex) programming relaxation. In fact, by employing higher order polynomial constraints in the reformulation phase, this technique can be applied to generate a hierarchy of relaxations. A noteworthy feature is that in several applications in the context of location-allocation, network design, economics, and engineering design, it has been demonstrated that even the first level relaxation in this hierarchy generates very tight relaxations. As a result, this has frequently enabled the solution of difficult nonconvex problems to near optimality, often via a single linear programming (LP) problem. With advances in LP technology and the widespread availability of related commercial software, this affords the hope and opportunity for solving practical nonconvex programming problems to acceptable limits of accuracy with relative ease. Moreover, the theory of RLT provides a unifying mechanism for generating valid inequalities and constructing algebraic convex hull representations in special cases.

We commence our discussion with an introductory chapter that motivates this work and provides a summary of related concepts and methodologies that comprise this book. The remainder of this book is organized into three parts. **Part I** contains 5 chapters that deal with discrete nonconvex programs. Chapter 2 considers linear mixed-integer zero-one programming problems that arise very frequently in real-world applications, and provides the basic, fundamental elements of the RLT methodology. The generation of a hierarchy of relaxations spanning the spectrum from the linear programming to the convex hull representation is presented. The approach is also shown to naturally extend to the class of multilinear polynomial programming problems.

Chapter 3 further extends and strengthens the RLT approach via the use of generalized factors in constructing a reformulation of a given mixed-integer zero-one problem. This modified approach is especially designed to exploit inherent special structures such as generalized and variable upper bounding constraints, partitioning or packing constraints, and also problem sparsity. Some insightful conditional logic based enhancements of RLT constraints are also discussed. Several illustrative examples, some computations, as well as a new tighter model for the Traveling Salesman Problem are presented to demonstrate the efficacy of this approach. In essence, this chapter constitutes the heart of the RLT methodology for solving discrete optimization problems.

The next three chapters deal with various theoretical extensions and applications of RLT. Chapter 4 presents an RLT approach for general discrete (not necessarily zero-one) mixed-integer problems. It is shown that, in concept, a parallel development can be designed through the use of Lagrange interpolating polynomial factors in the reformulation phase of the procedure, in lieu of the simple bound factors employed for the zero-one case. The analytical construct developed provides insights into translating classes of valid inequalities for 0-1 problems to the case of general discrete variables. Chapter 5 illustrates the use of RLT to generate strong valid inequalities and facets, as well as enhanced model representations for various special cases. Particular applications to the Boolean quadric polytope, the GUB-constrained knapsack polytope, and the set partitioning problem are considered. Finally, Chapter 6 deals with a novel application of RLT that facilitates the identification of variables that take on zero-one values in a continuous RLT relaxation, to *persist* in taking on these same values in an optimal solution. Hence, using such *persistency* results, we can *a priori* optimally fix certain variables based on the solution obtained for a particular RLT relaxation, even though this overall solution is not feasible to the underlying discrete problem. We derive such results for both unconstrained polynomial as well as certain constrained problems, that include the well-known vertex packing problem as a special case. Extensions to the development of other types of relaxations that possess this persistency properly are also discussed.

Part II of this book develops an RLT approach for solving *continuous* polynomial programming problems of the type that arise in various economics, chemical equilibrium, process design, and engineering design contexts. The situation becomes a little different here because the partitioning process as well as the reformulated representation must deal with nonlinearities and recognize that variables can assume continuous values. As a result, the design of RLT relaxations needs to be coordinated with the partitioning mechanism in order to induce (infinite) global convergence to an optimal solution. Chapter 7 discusses the basic RLT concepts and the development of related algorithms for polynomial programs that might have integral or rational variable exponents in the product terms. Chapter 8 presents several fundamental insights into the structure and properties of the RLT process, and their application to construct approximations for (and in some special cases, exact representations of) the closure convex hull of feasible solutions, where the objective function is accommodated into the constraints. This discussion also provides useful tools for using RLT to derive convex envelopes of several important classes of nonconvex functions. For the popular nonconvex quadratic programming problem, a detailed description of the design, convergence, and implementation of an RLT algorithm is described. Included in this development is a discussion on various algorithmic strategies such as constraint-selection, range-reductions, and the solution of relaxations via Lagrangian dual approaches. These can be gainfully employed to significantly enhance the computational performance of the algorithm. Extensive computational results are presented to demonstrate the effectiveness of the proposed methodology. The results of this chapter lie at the heart of Part II of the book.

The extension of many of these algorithmic strategies and concepts to general polynomial programming problems is discussed in Chapter 9. While such problems can be equivalently quadrified, it is theoretically proven that this can weaken ensuing RLT relaxations, even if all alternative equivalent quadratic formulations are *simultaneously* represented within the model. This chapter also presents special classes of RLT constraints based on convex bounds, grid factors, constraint factors, and Lagrange interpolating polynomials that can be generated to significantly tighten relaxations. Furthermore, range-reduction and constraint filtering strategies are also designed and embedded in a branch-and-bound algorithm. Computational results on various practical process and engineering design problems from the literature exhibit the efficacy of this procedure.

Part III of this book is devoted to special successful applications for which we have designed tailored RLT relaxations and embedded these into algorithms that have substantially advanced the state-of-the-art in solving these problems. Chapter 10 is concerned with discrete optimization applications of this type, including zero-one quadratic and mixed-integer bilinear programming problems, and various miscellaneous applications such as the discrete location-allocation problem, the quadratic assignment problem, the airline gate assignment problem, the traveling salesman problem, and several telecommunication network design problems. Chapter 11 deals with applications related to continuous nonconvex optimization problems. Included here is a discussion on the squared-Euclidean distance location-allocation problem, the linear complementarity problem, and various miscellaneous applications such as the rectilinear and the ℓp distance capacitated location-allocation problems, and the water distribution pipe network design problem.

This book is intended for researchers and practitioners who work in the area of discrete or continuous nonconvex optimization problems, as well as for students who are interested in learning about techniques for solving such problems. We have attempted to lay the foundations here of an idea that we have found to be very stimulating, and that has served to enhance the solution capability of many challenging problems in the field. Several problems exist for which specialized RLT designs can be developed, as demonstrated in this book, for generating improved representations, strong valid inequalities, and effective, powerful heuristics and exact algorithms. As we continue to explore this vast territory, we hope that this book provides an impetus for other researchers to join us on this opportunistic and exciting journey.

Hanif D. Sherali
Warren P. Adams

August, 1998

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the *National Science Foundation* under Grant Nos. DMI-9121419 and DMI9521398 and the *Air Force Office of Scientific Research* under Grant Nos. AFOSR 90-0191 and F49620-96-1-0274 in conducting this research, and in the publication of this book. Many graduate students have contributed scholarly assistance through long hours of investigation and computational implementation of RLT and its various applications. Alphabetically, these include Amine Alammedine, Eric Brown, Gyunghyun Choi, Patrick Driscoll, Terry Johnson, Ravi Krishnamurthy, Julie Lassiter, Youngho Lee, Robin Lougee-Heimer, Sridhar Ramachandran, Price Smith, and Cihan Tuncbilek. All of these students are now established professionals in government, academia, and industry. They continue to make strong contributions to society and we are very proud of them. We also appreciate related research collaborations of Professor Faiz Al-Khayyal at Georgia Tech, Professor Seong-in Kim at Korea University, and Professor Fred Glover at the University of Colorado. Last, but certainly not the least, a very special thanks to Sandy Dalton for her laborious, painstaking word processing of this work. Without her talents and efficiency, we would have been at a loss!

COPYRIGHT PERMISSIONS

This book is largely composed from a collection of papers written by the authors over the past decade. The following list acknowledges these publications and their respective journals and publishers on a chapter-by-chapter basis. Portions of these papers have been adapted with permission from the publishers as required. All rights are reserved as stipulated by the various copyright agreements.

Chapter 1:

Sherali, H. D. and W. P. Adams, Computational Advances on Using the Reformulation-Linearization Technique (RLT) to Solve Various Discrete and Continuous Nonconvex Programming Problems, *OPTIMA - The Mathematical Programming Society Newsletter*, **49**, 1-7, March, 1996. Copyright © 1996 by, and with kind permission from, *OPTIMA - Mathematical Programming Society.*

Chapter 2:

Sherali, H. D. and W. P. Adams, "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems," *SIAM Journal on Discrete Mathematics, 3*(3), 411-430, 1990. Copyright © 1990 by, and with kind permission from, *The Society for Industrial and Applied Mathematics.*

Chapter 3:

Sherali, H. D., W. P. Adams and P. Driscoll, "Exploiting Special Structures in Constructing a Hierarchy of Relaxations for 0-1 Mixed Integer Problems," *Operations Research,* **46**(3), 396-405, 1998. . Copyright © 1998 by, and with kind permission from, *The Institute for Operations Research and the Management Sciences,* 901 Elkridge Landing Road, Suite 400, Linthicum, MD 21090-2909.

Chapter 5:

Sherali, H. D. and Y. Lee, "Sequential and Simultaneous Liftings of Minimal Cover Inequalities for GUB Constrained Knapsack Polytopes," *SIAM Journal on Discrete Mathematics,* **8**(1), 133-153, 1995. Copyright © 1995 by, and with kind permission from, *The Society for Industrial and Applied Mathematics.*

Sherali H. D. and Y. Lee, "Tighter Representations for Set Partitioning Problems Via a Reformulation-Linearization Approach," *Discrete Applied Mathematics* **68**, 153-167, 1996. Copyright © 1995 by, and with kind permission from, *Elsevier Science-NL,* Sara Burgerhartstraat 25 1055 KV, Amsterdam, The Netherlands.

Sherali, H. D., Y Lee and W. P. Adams, "A Simultaneous Lifting Strategy for Identifying New Classes of Facets for the Boolean Quadric Polytope," *Operations Research Letters,* **17**(1), 19-26, 1995. Copyright © 1996 by, and with kind permission from, *Elsevier Science-NL,* Sara Burgerhartstraat 25 1055 KV, Amsterdam, The Netherlands.

Chapter 6:

Adams, W. P., J. B. Lassiter and H. D. Sherali, "Persistency in 0-1 Polynomial Programming," *Mathematics of Operations Research,* **23**, 359-389, 1998 . Copyright © 1998 by, and with kind permission from, *The Institute for Operations Research and the Management Sciences (INFORMS),* 901 Elkridge Landing Road, Suite 400, Linthicum, MD 21090-2909.

Chapter 7:

Sherali, H. D. and C. H. Tuncbilek, "A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation-Linearization Technique," *Journal of Global Optimization,* **2**, 101-112, 1992. Copyright © 1992 by, and with kind permission from, *Kluwer Academic Publishers.*

Sherali, H. D., "Global Optimization of Nonconvex Polynomial Programming Problems Having Rational Exponents, *Journal of Global Optimization,* **12**(3), 267-283 1998. Copyright © 1998 by, and with kind permission from, *Kluwer Academic Publishers.*

Chapter 8:

Sherali, H. D. and A. Alameddine, "A New Reformulation-Linearization Technique for Solving Bilinear Programming Problems," *Journal of Global Optimization,* **2**, 379-410, 1992. Copyright © 1992 by, and with kind permission from, *Kluwer Academic Publishers.*

Sherali, H. D. and C. H. Tuncbilek, "A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems," *Journal of Global Optimization*, **7**, 1-31, 1995. Copyright © 1995 by, and with kind permission from, *Kluwer Academic Publishers*.

Chapter 9:

Sherali, H. D. and C. H. Tuncbilek, "Comparison of Two Reformulation-Linearization Technique Based Linear Programming Relaxations for Polynomial Programming Problems," *Journal of Global Optimization*, **10**, 381-390, 1997. Copyright © 1997 by, and with kind permission from, *Kluwer Academic Publishers*.

Sherali, H. D. and C. H. Tuncbilek, "New Reformulation-Linearization Convexification Relaxations for Univariate and Multivariate Polynomial Programming Problems," *Operations Research Letters*, **21(1)**, 1-10, 1997. Copyright © 1997 by, and with kind permission from, *Elsevier Science-NL*, Sara Burgerhartstraat 25 1055 KV, Amsterdam, The Netherlands.

Chapter 10:

Adams, W. P. and H. D. Sherali, "A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems," *Management Science*, **32**(10), 1274-1290, 1986. Copyright © 1993 by, and with kind permission from, *The Institute of Management Sciences* (currently INFORMS), 2 Charles Street, Suite 300, Providence, Rhode Island 02904, USA.

Adams, W. P. and H. D. Sherali, "Mixed-Integer Bilinear Programming Problems," *Mathematical Programming*, **59**(3), 279-306, 1993. Copyright © 1993 by, and with kind permission from, *Elsevier Science-NL*, Sara Burgerhartstraat 25 1055 KV, Amsterdam, The Netherlands.

Chapter 11:

Sherali, H. D. and C. H. Tuncbilek, "A Squared-Euclidean Distance Location-Allocation Problem," *Naval Research Logistics*, **39**, 447-469, 1992. . Copyright © 1992 by, and with kind permission from, *John Wiley & Sons*, Inc.

Sherali, H. D., R. S. Krishnamurthy and F. A. Al-Khayyal, "A Reformulation-Linearization Technique Based Enumeration Approach for Linear Complementarity Problems," *Journal of Optimization Theory and Applications*, **99**(2), 481-507, November, 1998. Copyright © 1993 by, and with kind permission from, *Plenum Publishing Corp.*, 233 Spring Street, New York, NY 10013.

1 INTRODUCTION

Discrete and continuous nonconvex programming problems arise in a host of practical applications in the context of production, location-allocation, distribution, economics and game theory, process design, and engineering design situations. Several recent advances have been made in the development of branch-and-cut algorithms for discrete optimization problems and in polyhedral outer-approximation methods for continuous nonconvex programming problems. At the heart of these approaches is a sequence of linear programming problems that drive the solution process. The success of such algorithms is strongly linked to the strength or tightness of the linear programming representations employed.

This book addresses the role played by tight linear programming (LP) representations that are generated via automatic reformulation techniques in solving discrete and continuous nonconvex programming problems. Particularly, we focus on the elements of a new **Reformulation Linearization Technique (RLT)** in generating such linear (or sometimes convex) programming representations, that can be used not only to construct exact solution algorithms, but also to design powerful heuristic procedures. Hence, this enables one to derive good quality solutions to problems of practical sizes that arise in the aforementioned applications. Also, the RLT procedure is capable of generating representations of increasing degrees of strength, but with an accompanied increase in problem size. Coupled with the recent advances in LP technology, this permits one to incorporate tighter RLT based representations within the context of exact or heuristic methods.

1.1. Discrete Linear and Nonlinear Mixed-Integer Problems

The concept and importance of having tight linear programming relaxations to enhance the effectiveness of any algorithm for solving integer programming problems, in particular, has long been recognized. Most of the attention has focused on pure and mixed zero-one programming problems because of the wide variety of applications which these types of problems model. The basic idea is to try to construct a formulation whose continuous relaxation is "tight" in that it closely approximates the convex hull of feasible integer solutions, at least in the vicinity of optimal solutions.

To illustrate, consider the following class of fixed-charge location problems wherein up to m potential facilities having respective supplies $s_1, \ldots, s_m$ can be constructed to serve n customers, having respective demands $d_1, \ldots, d_n$. There is a fixed (amortized) cost f_i for constructing service facility i, and there is a variable distribution cost of c_{ij}

per unit shipped from facility i to customer j, for $i = 1, \ldots, m$, $j = 1, \ldots, n$. To formulate this problem, let us define decision variables

$$y_i = \begin{cases} 1 & \text{if facility } i \text{ is constructed} \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, \ldots, m$$

and

x_{ij} = quantity shipped from facility i to customer j, for $i = 1, \ldots, m$, and $j = 1, \ldots, n$.

Then, a "mathematically correct" representation of this problem can be stated as follows.

$$\text{Minimize} \quad \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} \le s_i y_i \quad \text{for } i = 1, \ldots, m \tag{1.1a}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \text{for } j = 1, \ldots, n \tag{1.1b}$$

$$x_{ij} \ge 0 \quad \text{for } i = 1, \ldots, m, \ j = 1, \ldots, n \tag{1.1c}$$

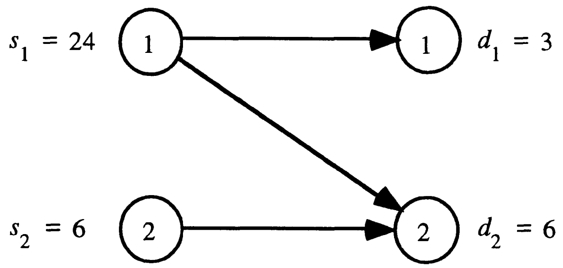
$$y_i \text{ binary for } i = 1, \ldots, m. \tag{1.1d}$$

Observe that the "correctness" of this formulation follows from the fact that constraints (1.1a) permit the supply s_i to be used only from facilities i that have been constructed $(y_i = 1)$.

Using formulation (1.1), one can solve moderately sized problems having about 20 potential facilities and 20 customers. However, this solution capability can be amplified more than ten-fold by incorporating some simple disaggregated constraints of the following type to replace (1.1c):

$$0 \le x_{ij} \le y_i \min\{s_i, d_j\} \text{ for } i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{1.2}$$

Note that $x_{ij} \le \min\{s_i, d_j\}$ (or even $x_{ij} \le s_i y_i$) $\forall\, i, j$ are implied by (1.1a-1.1c). However, while (1.2) is implied in the discrete 0-1 sense for this reason, the restrictions (2.1) are not necessarily implied in the continuous sense when $d_j < s_i$. Hence, this serves to strengthen the continuous or linear programming relaxation in which (1.1d) is replaced by $0 \le y_i \le 1$ for $i = 1, \ldots, m$. For example, consider an instance in which we have $m = 2$, $n = 2$ and the possible distribution paths as shown below. (Here, c_{21} is virtually infinite, so that facility 2 is disallowed from serving customer 1, and $c_{11} = c_{12} = c_{22} = 1$.)



If the finite fixed cost f_1 is sufficiently large in relation to $f_2 (f_1 > 4f_2)$, an optimal solution to the continuous relaxation of (1.1) will be to let $x_{11} = 3$, $x_{22} = 6$, $y_1 = 3/24 = 1/8$, and $y_2 = 6/6 = 1$. However, if we were to include (1.2) in the model, then we would enforce the constraint $x_{11} \le 3y_1$, while (1b) would require $x_{11} = 3$, leading to $y_1 = 1$ in the linear programming solution, (along with $y_2 = 0$, and $x_{12} = 6$). Hence, the continuous solution would solve the underlying mixed-integer program in this case. It is important to note that while it might appear in this simple example that an ordinary logical test based on the sparse structure of the permissible distribution pattern could have been used *a priori* to arrive at the conclusion that y_1 must be unity, the role of (1.2) turns out to be similar to that illustrated above in larger, denser problems as well. The reason being that for a fixed y, since the underlying problem in x is a transportation problem that admits an optimum for which the distribution paths that carry positive flows constitute a tree (or a forest) graph, this tendency along with (1.2) serves to tighten the problem representation to quite an extent in general.

The foregoing "disaggregation" concept can be used in various forms in order to enhance solution capability by constructing "*good*" models, that is, models that have tight underlying linear programming representations, rather than simply "*mathematically correct*" models (see Nemhauser and Wolsey, 1988, Parker and Rardin, 1988, Rardin and Choe, 1979, and Sherali and Adams, 1984, for example). While several other such strategies exist, and these comprise the "art" of constructing good model formulations, one can further embellish a given model through a reformulation process that augments it via the inclusion of appropriate additional variables and constraints, or that generates suitable valid, implied inequalities. Hopefully, such inequalities or cutting planes will have some partial convex-hull facet defining properties, such as being a facet of the knapsack polytope defined by some single identified constraint. The latter constraint might either be an original model constraint, or might be a suitable surrogate of the defining constraints of the model. This rationale has led to some crucial and critical research on the model construction and formulation process as in Balas (1985), Jeroslow (1980, 1984a, 1984b, 1985), Jeroslow and Lowe (1984, 1985), Johnson (1989), Meyer (1975, 1976, 1981), Meyer *et al.* (1980), Sherali and Adams (1989, 1990), and Williams (1985). Much work has also been done in converting classes of separable or polynomial nonlinear integer programming problems into equivalent linear integer programs, and for generating tight, valid inequalities for such problems, as in Adams and Sherali (1986, 1987a,b), Balas and Mazzola (1984a,b), Glover (1975), Glover and Woolsey (1973, 1974), Ibaraki (1976), Pardalos and Rosen (1984), Peterson (1971), Sherali and Adams (1989, 1990), Watters (1967), and Zangwill (1965).

Given a certain formulation of a problem, a preprocessing in the form of standard logical tests is prudent in order to possibly fix some binary variables or range restrict continuous variables (see Nemhauser and Wolsey, 1988, Parker and Rardin, 1988, or Taha, 1975, for example). Sometimes, a special variable redefinition technique may be applicable based on its structure as in Martin (1987) and Eppen and Martin (1985). In this approach, a linear transformation is defined on the variables to yield an equivalent formulation that tightens the continuous relaxation by constructing a partial convex hull of a specially structured subset of constraints. A more generally applicable technique is to augment the formulation through the addition of valid or implied inequalities that typically provide some partial characterization for the convex hull of feasible solutions. Some cutting plane generation schemes in this vein include the ones described in Martin and Schrage

(1981, 1983), Nemhauser and Wolsey (1990), Padberg (1973, 1979, 1980), Van Roy and Wolsey (1983), and Wolsey (1975, 1976). Automatic reformulation procedures utilizing such constraint generation schemes within a branch-and-bound or branch-and-cut framework are presented in Crowder *et al.* (1983), Hoffman and Padberg (1985, 1989, 1991), Johnson and Suhl (1980), Johnson *et al.* (1985), Nemhauser *et al.* (1991), Oley and Sjouquist (1982), Padberg and Rinaldi (1991), Spielberg and Suhl (1980), and Van Roy and Wolsey (1984, 1987). Besides these studies, ample evidence is available in the literature on the efficacy of providing tight linear programming relaxations for pure and mixed zero-one programming problems as in Adams and Sherali (1986, 1987a,b), Barany *et al.* (1983), Crowder and Padberg (1986), Geoffrion and Graves (1974), Geoffrion and McBryde (1979), Magnanti and Wong (1981), McDaniel and Devine (1977), Rardin and Choe (1979), and Williams (1974), among many others.

In particular, in a series of papers, Crowder *et al.* (1983) and Hoffman and Padberg (1985, 1989a,b, 1991) recommend and test various automatic reformulation or preprocessing strategies. A typical schema of this type commences with a classification of problem constraints into appropriate sets such as:

(i) Specially ordered sets (SOS) of constraints of the type $\sum_{j \in L} x_j + \sum_{j \in H} \bar{x}_j \leq 1$, where $\bar{x}_j \equiv (1 - x_j)$ and x_j, $j \in L \cup H$ are binary variables.

(ii) Invariant knapsacks having nonzero coefficients of values ± 1 but which cannot be transformed into SOS constraints.

(iii) Plant location constraints of the type $\sum_{j \in P} x_j \le \theta x_p$ where x_j, $j \in P \bigcup \{p\}$ are binary and θ is some (supply type of) coefficient.

(iv) Ordinary knapsack constraints that do not possess any of the foregoing structures.

Various logical tests are then conducted to tighten individual constraint coefficients and right-hand sides via Euclidean and coefficient reductions, as well as to detect blatant infeasibilities and redundancies and to *a priori* fix variables at deducible binary values. The SOS and plant location constraints are used in concert with the knapsack classes of constraints to conduct such logical tests (see Hoffman and Padberg, 1991). In addition, after solving the resultant linear program, if a fractional solution is obtained, then a further strengthening of this LP representation is conducted by fixing variables based on reduced-cost coefficients along with the objective value of a derived heuristic solution, and additional constraints that delete the fractional LP solution are generated. The generated constraints are valid for the original problem, and might be based on the special structure of the problem, or for general unstructured problems, these are typically lifted minimal covers derived after possibly projecting out certain variables at values of 0 and 1, as necessary (see Hoffman and Padberg, 1989, 1991). Experimental results indicate that this constraint generation process is a crucial and indispensable step of the algorithm. This combination of preprocessing, constraint generation, and the derivation of a heuristic solution in concert with the LP solution process is conducted at each node of the enumeration tree in the context of a branch-and-cut algorithm. Large, unstructured, and sparse real-world problems (up to 2757 variables, and 756 constraints, including 353

SOS rows) have been successfully solved (1 hr of cpu time on a VAX 8800 machine with coding in FORTRAN) using such an approach. Further theoretical and computational enhancements on generating various types of strong lifted inequalities for ordinary as well as for generalized upper bounded (GUB) knapsack polytopes are presented in Glover *et al.* (1996), Gu *et al.* (1995a,b), Sherali and Lee (1995), and Wolsey (1990).

In Part I of this book, we shall be discussing in detail the Reformulation-Linearization Technique (RLT) of Sherali and Adams (1989, 1990, 1994) along with its several enhancements and extensions. This procedure is also an automatic reformulation technique that can be used to derive tight LP representations as well as to generate strong valid inequalities. Consider a mixed-integer zero-one linear programming problem whose feasible region X is defined in terms of some inequalities and equalities in binary variables $x = (x_1, \ldots, x_n)$ and a set of bounded continuous variables $y = (y_1, \ldots, y_m)$. Given a value of $d \in \{1, \ldots, n\}$, this RLT procedure constructs various polynomial factors of degree d comprised of the product of some d binary variables x_j or their complements $(1 - x_j)$. These factors are then used to multiply each of the constraints defining X (including the variable bounding restrictions), to create a (nonlinear) polynomial mixed-integer zero-one programming problem. Using the relationship $x_j^2 = x_j$ for each binary variable x_j, $j = 1, \ldots, n$, substituting a variable w_J and v_{Jk}, respectively, in place of each nonlinear term of the type $\pi_{j \in J} x_j$, and $y_k \pi_{j \in J} x_j$, and relaxing integrality, the nonlinear polynomial problem is re-linearized into a higher dimensional polyhedral set X_d defined in terms of the original variables *(x, y)* and the new variables *(w, v)*. For X_d to be equivalent to *X*, it is only necessary to enforce *x* to be binary valued, with the remaining variables treated as continuous valued, since the binariness on the *x*-variables is shown to automatically enforce the required product relationships on the *w*- and *v*-variables. Denoting the projection of X_d onto the space of the original *(x, y)*-variables as X_{Pd}, it is shown (see Chapter 2) that as *d* varies from 1 to *n*, we get,

$$X_{P0} \supseteq X_{P1} \supseteq X_{P2} \supseteq \cdots \supseteq X_{Pn} \equiv \text{conv}(X),$$

where X_{P0} is the ordinary linear programming relaxation, and conv*(X)* represents the convex hull of X. The projection process also produces an algebraic representation of $X_{Pn} \equiv \text{conv}(X)$ which has a structure that can be exploited to derive facets for various classes of special combinatorial optimization problems, as we shall see in Chapter 5. As mentioned above, this can play an important role in devising powerful algorithms for such problems. The extension of these concepts to general discrete (as opposed to zero-one) mixed-integer problems is presented in Chapter 4 based on the work of Adams and Sherali (1996), while Chapter 6 develops various persistency results (see Adams, Lassiter, and Sherali, 1995) that permit the optimal fixing of variables at values realized by any of the foregoing relaxations. Indeed, the general RLT constructs can be modified to produce related linearizations that possess desirable persistency properties.

Sherali and Adams (1989) also demonstrate the relationship between their hierarchy of relaxations and that which can be generated through disjunctive programming techniques. Balas (1985) has shown how a hierarchy spanning the spectrum from the linear programming relaxation to the convex hull of feasible solutions can be generated for linear mixed-integer zero-one programming problems by inductively representing the feasible region at each stage as a conjunction of disjunctions, and then taking its hull relaxation. This hull relaxation amounts to constructing the intersection of the individual convex hulls of the different disjunctive sets, and hence yields a relaxation. Sherali and Adams show that their hierarchy produces a different, stronger set of intermediate relaxations that lead to the underlying convex hull representation. To view their relaxations as Balas' hull relaxations requires manipulating the representation at any stage d to write it as a conjunction of a *non-standard* set of disjunctions. Moreover, by the nature of the RLT approach, Sherali and Adams also show (see Chapter 2) how one can readily construct a hierarchy of *linear* relaxations leading to the convex hull representation for mixed-integer zero-one *polynomial* programming problems having no cross-product terms among continuous variables, an important result not shared by the earlier disjunctive work.

In connection with the final comment above, we remark that Boros *et al.* (1989) have independently developed a similar hierarchy for the special case of the unconstrained, quadratic pseudo-Boolean programming problem. For this case, they construct a standard linear programming relaxation that coincides with our relaxation at level $d = 1$, and then show in an *existential* fashion how a hierarchy of relaxations indexed by $d = 1, \ldots, n$ leading up to the convex hull representation at level n can be generated. This is done by including at level d, constraints corresponding to the extreme directions of the cone of nonnegative quadratic pseudo-Boolean functions that involve at most d of the n-variables. Each such relaxation can be viewed as the projection of one of our *explicitly* stated higher-order relaxations onto the variable space of the first level for this special case. Moreover, our approach also permits one to consider general pseudo-Boolean polynomials, constrained problems, as well as mixed-integer situations.

Lovasz and Shrijver (1989) have also independently proposed a similar hierarchy of relaxations for linear, *pure* 0-1 programming problems, which essentially amounts to deriving a modified version of X_1 from X_0, finding the projection X_{P1}, and then repeating this step by replacing X_0 with X_{P1}. Continuing in this fashion, they show that in n steps, conv(X) is obtained. However, from a practical viewpoint, while the relaxations $X_1, X_2, \ldots$ of Sherali and Adams are explicitly available and directly implementable, the projections required by Lovasz and Shrijver are computationally burdensome, necessitating the potentially exponential task of vertex enumeration. Moreover, extensions to mixed-integer or to nonlinear zero-one problems are not evident using this development.

Another hierarchy along the same lines has been proposed by Balas *et al.* (1993). In this hierarchy, the set X_1 is generated as in Sherali and Adams (1989), but using factors involving only one of the binary variables, say, x_1. Projecting the resulting formulation onto the space of the original variables produces the convex hull of feasible solutions to the original LP relaxation with the added restriction that x_1 is binary valued. This follows from Sherali and Adams (1989) since it is equivalent to treating only x_1 as binary valued and the remaining variables as continuous, and then generating the convex hull representation. Using the fact that x_1 is now binary valued at all vertices, this process is then repeated using another binary variable, say, x_2, in order to determine the convex hull of feasible vertices at which both x_1 and x_2 are binary valued. Continuing with the remaining binary variables $x_3, \ldots, x_n$ in this fashion, produces the convex hull representation at the final stage. Based on this construct, which amounts to a specialized application of Sherali and Adams' hierarchy, Balas *et al.* describe and test a cutting plane algorithm. Encouraging computational results are reported, despite the fact that this specialization produces weaker relaxations than the original Sherali-Adams' relaxations. Since this cutting plane generation scheme is based simply on the first level of Sherali and Adams' hierarchy in which bininess is enforced on only a single variable at a time, the prospect of enhancing computational performance by considering multiple binary variables at a time appears to be promising.

As far as the use of RLT as a practical computational aid is concerned, one may simply work with the relaxation X_1 itself, which has frequently proven to be beneficial. Using variations on the first-order implementation of RLT (a partial generation of X_1 enhanced by additional constraint products), Adams and Johnson (1994), Adams and Sherali (1984, 1986, 1990, 1991, 1993), Sherali and Alameddine (1992), Sherali and Brown (1994), Sherali, Krishnamurthy, and Al-Khayyal (1996), Sherali, Ramachandran, and Kim (1994), Sherali and Smith (1995), and Sherali and Tuncbilek (1995, 1996) have shown how highly effective algorithms can be constructed for various classes of problems and applications (see Chapters 10 and 11). A study of special cases of this type has provided insights into useful implementation strategies based on first-order products. Additionally, techniques for generating tight valid inequalities that are implied by higher order relaxations may be devised, or explicit convex hull representations or facetial inequalities could be generated by applying the highest order RLT scheme to various subsets of sparse constraints that involve a manageable number of variables. The lattermost strategy would be a generalization of using facets of the knapsack polytope from minimal covers (see Balas and Zemel, 1978, and Zemel, 1987), so successfully implemented by Crowder *et al.* (1983) and Hoffman and Padberg (1991). An alternative would be to apply a higher order scheme using the binary variables that turn out to be fractional in the initial linear programming solution. Letting x_j, $j \in J_f$, represent such a subset, it follows from Sherali and Adams (1989) that by deriving $X_{|J_f|}$ based on this subset, which is manageable if $|J_f|$ is relatively small, the resulting linear program will yield binary values for x_j, $\forall j \in J_f$. This would be akin to employing judicious partial convex hull representations. The development and testing of many such implementation strategies are discussed in Chapter 10, and enhancements are continuing, even as of this writing.

1.2. Continuous Nonconvex Polynomial Programming Problems

Although the Reformulation-Linearization Technique (RLT) was originally designed to employ factors involving *zero-one* variables in order to generate zero-one (mixed-integer) polynomial programming problems that are subsequently re-linearized, as we shall see in Part 2 of this book, the approach can be extended in concept to continuous, bounded variable polynomial programming problems as well. Problems of this type have the following form, involving the optimization of a polynomial objective function subject to polynomial constraints in a set of continuous, bounded variables.

Minimize $\{\phi_0(x) : \quad \phi_r(x) \geq \beta_r \text{ for } r = 1, \ldots, R_1, \ \phi_r(x) = \beta_r \text{ for}$

$r = R_1 + 1, \ldots, R, \ x \in \Omega\}$

where,

$$\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[\pi_{j \in J_{rt}} x_j^{a_{jrt}} \right] \text{ for } r = 0, 1, \ldots, R,$$

and where

$$\Omega \equiv \{x : \quad 0 \leq \ell_j \leq x_j \leq u_j < \infty, \text{ for } j = 1, \ldots, n\}.$$

Here, x is an n-dimensional vector of decision variables, T_r is an index set for the terms defining $\phi_r(\bullet)$, α_{rt} are real coefficients for the polynomial terms defined by the index sets J_{rt}, $t \in T_r$, and a_{jrt} are natural numbers $\forall\ (j, r, t)$. (Problems having rational exponents are also analyzed in Chapter 7.) Such polynomial programming problems find a wide range of applications in production planning, location, and distribution problems (see Horst and Tuy, 1990, and Vaish, 1974), risk management problems (see Alameddine, 1990), and various engineering design problems (see Duffin *et al.*, 1969, Peterson, 1976, and Floudas and Pardalos, 1987). In particular, Floudas and Pardalos (1987) discuss applications and provide test examples for chemical and mechanical engineering problems, Floudas and Visweswaran (1990a,b), and Lasdon *et al.* (1979) address applications to chemical engineering, and pooling and blending problems, and Shor (1989) discusses a related application to water treatment and distribution. Various algorithms employing piecewise-linear convex envelopes or minorants (polyhedral under-estimating functions), or polyhedral outer-approximations (see Al-Khayyal and Falk, 1983, Floudas and Visweswaran, 1991, Horst and Tuy, 1990, Pardalos and Rosen, 1987, Sherali and Alameddine, 1992, Sherali and Smith, 1995, and Sherali and Tuncbilek, 1995) have been proposed for special cases of such problems. Again, it has been computationally reiterated that tighter approximations of this type naturally benefit the solution algorithms.

Sherali and Tuncbilek (1992) have proposed an algorithm based on an RLT scheme in which products of factors $(u_j - x_j)$ and $(x_j - \ell_j)$, $j = 1, \ldots, n$, are taken δ at a time and are restricted to be nonnegative, where δ is the highest degree of any polynomial term appearing in the problem. The resulting problem is then linearized by substituting a variable X_J for each product term $\pi_{j \in J} x_j$, where J might contain repeated indices. Using this linear programming relaxation, and partitioning the problem based on splitting the bounding interval for that variable x_p which produces the highest discrepancy between $X_{J \cup p}$ and X_J over all J in the linear programming solution, a convergent branch-and-bound algorithm is designed. Chapter 7 discusses this procedure along with an extension due to Sherali (1996) to handle rational exponents on the polynomial functions, instances of which arise in many engineering design problems (see

Floudas and Pardalos, 1987, and Grossmann, 1996). Again, various transformations and (partial) constraint and bound factor products can be gainfully employed to judiciously construct an appropriate linear programming relaxation. For example, Shor (1990) and Floudas and Visweswaran (1991) have suggested a successive quadratic variable substitution strategy to transform a given polynomial programming problem to one of equivalently minimizing a quadratic objective function subject to quadratic constraints. Hence, we can apply this transformation before using the RLT scheme, as compared with using relaxations produced by applying RLT directly to the polynomial program itself. We remark here that while the former might be convenient with respect to the size of the resulting reformulation as compared with directly applying RLT to the original polynomial program, Sherali and Tuncbilek (1995) have shown that this will produce a weaker linear programming relaxation, even if *all* possible quadrification transformations of this type are *simultaneously* incorporated within the derived quadratic polynomial program. Hence, a suitable compromise needs to be made in this connection. Insights into various implementation strategies, as well as computational experience on many classes of problems, including bilinear programming, indefinite quadratic programming, and various location-allocation and engineering design problems, are presented in Chapters 8, 9, and 11.

1.3. Coping with Large-Scale Representations

While the RLT process leads to tight linear programming relaxations for the underlying discrete or continuous nonconvex problems being solved as discussed above, one has to contend with the repeated solutions of such large-scale linear programs. By the nature of the RLT process, these linear programs possess a special structure induced by the replicated products of the original problem constraints (or its subset) with certain designated variables. At the same time, this process injects a high level of degeneracy in the problem since blocks of constraints automatically become active whenever the factor expression that generated them turns out to be zero at any feasible solution, and the condition number of the bases can become quite large. As a result, simplex-based procedures and even interior-point methods experience difficulty in coping with such reformulated linear programs (see Adams and Sherali, 1993, for some related computational experience). On the other hand, a Lagrangian duality based scheme can not only exploit the inherent special structures, but can quickly provide near optimal primal and dual solutions that serve the purpose of obtaining tight lower and upper bounds. However, for a successful use of this technique, there are two critical issues. First, an appropriate formulation of the underlying Lagrangian dual must be constructed (see Shapiro, 1979, and Fisher, 1981). Sherali and Myers (1989) also discuss and test various strategies and provide guidelines for composing suitable Lagrangian dual formulations. Second, an appropriate nondifferentiable optimization technique must be employed to solve the Lagrangian dual problem. Cutting plane or outer linearization methods (see Bazaraa and Goode, 1979) can quickly accumulate far too many constraints in the master program, and thereby get bogged down. The first-order subgradient methods of Poljak (1967, 1969), and Held *et al.* (1974) are slow to converge, and can stall far from optimality. Higher-order bundle methods or aggregate subgradient methods (see Kiwiel, 1983, 1985, 1990, 1991, Lemarechal, 1978, 1982, Lemarechal *et al.*, 1980, and Mifflin, 1982), or quasi-Newton analogies developed for nondifferentiable optimization as in Lemarechal (1975), and the method of space-dilation in the direction of the difference of two subgradients as proposed by Shor (1985), have far better convergence characteristics. However, they are suitable only for moderately sized problems. For the size of problems that are encountered by us in the context of RLT, it appears imperative to use conjugate subgradient methods as in Camerini *et al.* (1975), Wolfe (1975), Sherali and Ulular (1989), and Sherali *et al.* (1995, 1996), that employ higher-order information, but in a manner involving minor additional effort and storage over traditional subgradient algorithms. Since these types of algorithms are not usually dual adequate (Geoffrion, 1972), for algorithms that require primal solutions for partitioning purposes, some extra work becomes necessary. For this purpose, one can either use powerful LP solvers such as CPLEX (1990) on suitable surrogate versions of the problem based on the derived dual solution, or apply primal solution recovery procedures as in Sherali and Choi (1995) along with primal penalty function techniques as in Sherali and Ulular (1989).

We have discussed in this chapter our motivation for this book, stressing the role of tight linear programming representations in successfully solving large-scale mixed-integer zero-one problems as well as continuous, nonlinear polynomial programming problems. While such problems are very difficult to solve, particularly the nonlinear instances, they do find a wide range of applications in many production, distribution, process design, and engineering design contexts. Hence, the open challenge is to devise solution procedures that are capable of handling realistically sized instances of problems arising in such applications.

In this respect, we anticipate that automatic reformulation techniques, such as the RLT scheme discussed in this book, will play a crucial role over the next decade, and beyond, in enhancing problem-solving capability. Ongoing advances in solving large-scale linear programming problems will provide a further impetus to such techniques, which typically tend to derive tighter representations in higher dimensions and with additional constraints. Furthermore, such RLT and constraint generation concepts can be specialized to exploit particular structures inherent in the nonconvex models for different classes of problems. It is through such a combination of general strategies for generating tight LP representations, along with specializations for particular structures, that substantial improvements can be made in the solution capability for many of these difficult nonconvex problems.

PART I

DISCRETE NONCONVEX PROGRAMS

2

RLT HIERARCHY FOR MIXED-INTEGER ZERO-ONE PROBLEMS

Consider a linear mixed-integer zero-one programming problem whose (nonempty) feasible region is given as follows:

$$X = \{(x, y): \sum_{j=1}^{n} \alpha_{rj} x_j + \sum_{k=1}^{m} \gamma_{rk} y_k \geq \beta_r \text{ for } r = 1, \ldots, R,$$

$$0 \leq x \leq e_n, \ x \text{ integer}, \ 0 \leq y \leq e_m\}, \tag{2.1}$$

where e_n and e_m are, respectively, column vectors of n and m entries of 1, and where the continuous variables y_k are assumed to be bounded and appropriately scaled to lie in the interval [0, 1] for $k = 1, \ldots, m$. (Upper bounds on the continuous variables are imposed here only for convenience in exposition, as we comment on later in the discussion.) Note that any equality constraints present in the formulation can be accommodated in a similar manner as are the inequalities in the following derivation, and we omit writing them explicitly in (2.1) only to simplify the presentation. However, we will show later that the equality constraints can be treated in a special manner which, in fact, might sometimes encourage the writing of the R inequalities in (2.1) as equalities by using slack variables.

In this chapter, we present the basic construction process of the *Reformulation-Linearization Technique* (RLT). For the region described in (2.1), given any level $d \in \{0, \ldots, n\}$, this technique first converts the constraint set into a polynomial mixed-integer zero-one set of restrictions by multiplying the constraints with some suitable d-degree polynomial factors involving the n binary variables and their complements, and subsequently linearizes the resulting problem through appropriate variable transformations. For each level d, this produces a higher dimensional representation of the feasible region (2.1) in terms of the original variables x and y, and some new variables (w, v) that are defined to linearize the problem. Relaxing integrality, the projection, or "shadow," of this higher dimensional polyhedral set on the original variable space produces a tighter envelope for the convex hull of feasible solutions to (2.1), than does its ordinary linear programming (LP) or continuous relaxation. In fact, as d varies from 0 to n, we obtain a hierarchy of such relaxations or shadows, each nested within the previous one, spanning the spectrum from the ordinary linear programming relaxation to the convex hull of feasible solutions. As mentioned earlier, the first level relaxation has itself proven to be sufficiently tight to benefit solution algorithms for several classes of problems. Moreover, as we shall see later in Chapter 5, the explicit algebraic representation that RLT provides for the convex hull of feasible solutions permits the derivation of classes of strong valid inequalities and facets for specially structured combinatorial optimization problems. Indeed, this explicit representation also provides a direct link between continuous and discrete sets, promoting the persistency results of Chapter 6.

The remainder of this chapter is organized as follows. In Section 2.1, we describe the basic RLT procedure and illustrate it with some numerical examples. The validity of the hierarchy and the proof that the convex hull is obtained at the highest level is presented in Section 2.2. Some insights into the structure of the polyhedral representations produced by RLT, and the characterization of facets are highlighted in Sections 2.3 and 2.4, respectively. Finally, we discuss two special extensions in Section 2.5, one dealing with the treatment of equality constraints, and the other dealing with nonlinear, polynomial mixed-integer zero-one programming problems. As we shall see, the RLT process is equally applicable to this latter, more general, class of problems, and generates a similar hierarchy of nested polyhedral relaxations, leading to the convex hull of feasible solutions.

2.1. Basic RLT Process for Linear Mixed-Integer 0-1 Problems

We now describe the RLT process for constructing a relaxation X_d of the region X defined in (2.1) corresponding to any level $d \in \{0, 1, \ldots, n\}$. For $d = 0$, the relaxation X_0 is simply the LP relaxation obtained by deleting the integrality restrictions on the x-variables. In order to construct the relaxation for any level $d \in \{1, \ldots, n\}$, let us consider the *bound-factors* $x_j \geq 0$ and $(1 - x_j) \geq 0$ for $j = 1, \ldots, n$ and let us compose *bound-factor products of degree* (*or order*) *d* by selecting some *d* distinct variables from the set $x_1, \dots, x_n$, and by using either the bound-factor x_j or $(1 - x_j)$ for each selected variable in a product of these terms. Mathematically, for any $d \in \{1, \dots, n\}$, and for each possible selection of d distinct variables, these (nonnegauve polynomial) *bound factor products of degree* (*or order*) *d* are given by

$$F_d(J_1, J_2) = \left[\prod_{j \in J_1} x_j\right]\left[\prod_{j \in J_2} (1 - x_j)\right] \text{for each } J_1, J_2 \subseteq N \equiv \{1, \dots, n\}$$

$$\text{such that } J_1 \cap J_2 = \emptyset \text{ and } |J_1 \cup J_2| = d. \tag{2.2}$$

Any (J_1, J_2) satisfying the conditions in (2.2) will be said to be of *order d*. For example, for $n = 3$ and $d = 2$, these factors are $x_1x_2, x_1x_3, x_2x_3, x_1(1 - x_2)$, $x_1(1 - x_3)$, $x_2(1 - x_1)$, $x_2(1 - x_3)$, $x_3(1 - x_1)$, $x_3(1 - x_2)$, $(1 - x_1)(1 - x_2)$, $(1 - x_1)(1 - x_3)$, and $(1 - x_2)(1 - x_3)$. In general, there are $\binom{n}{d}2^d$ such factors. For convenience, we will consider the single factor of degree zero to be $F_0(\emptyset, \emptyset) \equiv 1$, and accordingly assume products over null sets to be unity (so that the aforementioned set X_0 is the linear programming relevation). Using these factors, let us construct a relaxation X_d of X, for any given $d \in \{0, \dots, n\}$, using the following two steps that comprise the *Reformulation-Linearization Technique* (RLT).

Step 1 (Reformulation Phase): Multiply each of the inequalities in (2.1), including $0 \le x \le e_n$ and $0 \le y \le e_m$, by each of the factors $F_d(J_1, J_2)$ of degree *d* as defined in (2.2). Upon using the identity $x_j^2 \equiv x_j$ (and so $x_j(1 - x_j) = 0$) for each binary variable x_j, $j = 1,\ldots,n$, this gives the following set of additional, implied, nonlinear constraints

$$\left[\sum_{j\in J_1}\alpha_{rj} - \beta_r\right]F_d(J_1,J_2) + \sum_{j\in N-(J_1\cup J_2)}\alpha_{rj}F_{d+1}(J_1+j,J_2) + \sum_{k=1}^{m}\gamma_{rk}y_kF_d(J_1,J_2) \geq 0$$

for $r = 1,\ldots,R$ and for each (J_1, J_2) of order d, (2.3a)

$F_D(J_1,J_2) \geq 0$ for each (J_1,J_2) of order $D \equiv \min\{d+1,n\}$ (2.3b)

$$F_d(J_1,J_2) \geq y_kF_d(J_1,J_2) \geq 0$$

for $k = 1,\ldots,m$, and for each (J_1,J_2) of order d. (2.3c)

Step 2 (Linearization Phase): Viewing the constraints in (2.3) in expanded form as a sum of monomials, linearize them by substituting the following variables for the corresponding nonlinear terms for each $J \subseteq N$:

$$w_J = \prod_{j\in J}x_j \text{ and } v_{Jk} \equiv y_k\prod_{j\in J}x_j, \text{ for } k = 1,\ldots,m. \tag{2.4a}$$

We will assume the notation that

$$w_j \equiv x_j \text{ for } j = 1,\ldots,n,\; w_\varnothing \equiv 1, \text{ and } v_{\varnothing k} \equiv y_k \text{ for } k = 1,\ldots,m. \tag{2.4b}$$

Denoting by $f_d(J_1,J_2)$ and $f_d^k(J_1,J_2)$ the respective linearized forms of the polynomial expressions $F_d(J_1,J_2)$ and $y_kF_d(J_1,J_2)$ under such a substitution, we obtain the following polyhedral set X_d whose projection onto the (x,y) space is claimed to yield a relaxation for X:

$X_d = \{(x, y, w, v):$

$$\left[\sum_{j\in J_1} \alpha_{rj} - \beta_r\right] f_d(J_1, J_2) + \sum_{j\in N-(J_1\cup J_2)} \alpha_{rj} f_{d+1}(J_1 + j, J_2) + \sum_{k=1}^{m} \gamma_{rk} f_d^k(J_1, J_2) \geq 0$$

for $r = 1, \ldots, R$ and for each (J_1, J_2) of order *d*, (2.5a)

$f_D(J_1, J_2) \geq 0$ for each (J_1, J_2) of order $D \equiv \min\{d + 1, n\}$, (2.5b)

$f_d(J_1, J_2) \geq f_d^k(J_1, J_2) \geq 0$ for $k = 1, \ldots m$,

and for each (J_1, J_2) of order $d\}$. (2.5c)

Let us denote the projection of X_d onto the space of the original variables (x, y) by

$$X_{Pd} = \{(x, y): \ (x, y, w, v) \in X_d\} \text{ for } d = 0, 1, \ldots, n. \tag{2.6}$$

Then, the main result of this chapter is that

$$X_{P0} \equiv X_0 \supseteq X_{P1} \supseteq X_{P2} \supseteq \cdots \supseteq X_{Pn} \equiv conv(X) \tag{2.7}$$

where $X_{P0} \equiv X_0$ (for $d = 0$) denotes the ordinary linear programming relaxation, and $conv(X)$ denotes the convex hull of *X*.

Notice that the nonlinear product constraints generated by this RLT process are implied by the original constraints, and were it not for the fact that we have explicitly imposed $x_j^2 = x_j$ (or $x_j(1 - x_j) = 0$) for each binary variable, this process would have simply produced a relaxation that is equivalent to the ordinary LP relaxation. Hence, the key to the tightening lies in the recognition of binariness of x_j in replacing x_j^2 by x_j for each

$j = 1, \ldots n$. (The reader may wish to reflect here on the following question, which will be answered in Chapter 4: What is the corresponding step if x_j is some general discrete variable that can realize a finite set of possible values?)

It is of interest to note that the initial research into the automatic generation of higher-dimensional linear inequalities based on the products of 0-1 factors with inequality restrictions appeared in Adams (1985) and Adams and Sherali (1986, 1990). These earlier works formulated tight linearizations for pure and mixed 0-1 programs having quadratic terms in the objective function, and coincide with the level-1 RLT relaxation, where the quadratic objective terms are linearized using the exact operations of (2.4).

Example 2.1. Consider the following mixed-integer 0-1 constraint region.

$$X = \{(x, y): \; x + y \le 2, \; -x + y \le 1, \; 2x - 2y \le 1, \; x \text{ binary and } y \ge 0\}. \tag{2.8}$$

The corresponding LP relaxation of X, X_0, is depicted in Figure 2.1. Note that no explicit upper bound on the continuous variable y is included in (2.8), although the other defining constraints of X imply the boundedness of y. The RLT process can be applied directly to X as above, without creating any explicit upper bound on y.

Notice that for this instance, we have $n = 1$, and so by our foregoing discussion, the relaxation X_1 at level $d = 1$ should produce the convex hull representation. Let us verify this fact.

For $d = 1$, the bound-factor (products) of order 1 are simply x and $(1 - x)$. Multiplying the constraints of X by x and by $(1 - x)$, and using $x^2 = x$ along with the linearizing substitution $v = xy$ as given by (2.4a), we get the following constraints in the higher dimensional space of the variables (x, y, v). This represents the set X_1.

$$x + y \le 2 \quad \begin{array}{ll} * \, x \Rightarrow & -x \le -v \\ * \, (1 - x) \Rightarrow & 2x + y \le 2 + v \end{array}$$

$$-x + y \le 1 \quad \begin{array}{ll} * \, x \Rightarrow & -2x \le -v \\ * \, (1 - x) \Rightarrow & x + y \le 1 + v \end{array}$$

$$2x - 2y \le 1 \quad \begin{array}{ll} * \, x \Rightarrow & x \le 2v \\ * \, (1 - x) \Rightarrow & x - 2y \le 1 - 2v \end{array}$$

$$y \ge 0 \quad \begin{array}{ll} * \, x \Rightarrow & v \ge 0 \\ * \, (1 - x) \Rightarrow & -y \le -v \end{array}$$

along with $0 \le x \le 1$.

To examine the projection X_{P1} of X_1 onto the space of the (x, y) variables, let us rewrite X_1 as follows.

$$X_1 = \{(x, y, v): \; v \ge 2x + y - 2, \; v \ge x + y - 1, \; v \ge x / 2, \; v \ge 0,$$

$$v \le x, \; v \le 2x, \; v \le (1 - x + 2y) / 2, \; v \le y\}.$$

This yields its projection (using Fourier-Motzkin elimination) as

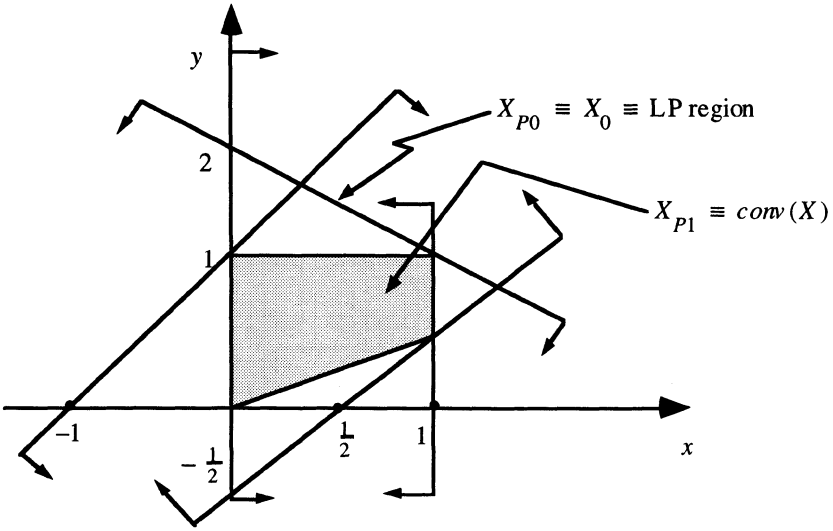


Figure 2.1. Illustration of RLT relaxations for Example 2.1.

$$X_{P1} = \{(x, y): \ \max\{2x + y - 2, \ x + y - 1, x / 2, \ 0\} \leq \min\{x, 2x, \ (1 - x + 2y) / 2, \ y\}\}.$$

Writing out the corresponding equivalent linear inequalities and dropping redundant constraints, we obtain

$$X_{P1} \equiv X_{Pn} = \{(x, y): \ x \leq 2y, \ 0 \leq x \leq 1, \ y \leq 1\}$$

which describes $conv(X)$ as seen in Figure 2.1.

Example 2.2. Consider the set

$$X = \{(x, y): \ \alpha_1 x_1 + \alpha_2 x_2 + \gamma_1 y_1 + \gamma_2 y_2 \geq \beta, \ 0 \leq x \leq e_2, x \text{ integer}, \ 0 \leq y \leq e_2\}.$$

Hence, we have $n = m = 2$. Let us consider $d = 2$, so that $D \equiv \min\{n+1, d\} = 2$ as well. The various sets (J_1, J_2) of order 2 and the corresponding factors are given below:

(J_1, J_2)	$(\{1,2\},\emptyset)$	$(\{1\},\{2\})$	$(\{2\},\{1\})$	$(\emptyset,\{1,2\})$
$F_2(J_1, J_2)$	x_1x_2	$x_1(1-x_2)$	$x_2(1-x_1)$	$(1-x_1)(1-x_2)$
$f_2(J_1, J_2)$	w_{12}	$x_1 - w_{12}$	$x_2 - w_{12}$	$1-(x_1+x_2)+w_{12}$
$f_2^k(J_1, J_2), k=1,2$	v_{12k}	$v_{1k} - v_{12k}$	$v_{2k} - v_{12k}$	$y_k - (v_{1k}+v_{2k}) + v_{12k}$

Hence, we obtain the following constraints (2.9a)-(2.9c) corresponding to (2.5a)-(2.5c), respectively, where v_{Jk} has been written as $v_{J,k}$ for clarity:

$X_2 = \{(x, y, w, v)$:

$$\left.\begin{aligned}
&(\alpha_1 + \alpha_2 - \beta)w_{12} + \gamma_1 v_{12,1} + \gamma_2 v_{12,2} \geq 0, \\
&(\alpha_1 - \beta)[x_1 - w_{12}] + \gamma_1(v_{1,1} - v_{12,1}) + \gamma_1(v_{1,2} - v_{12,2}) \geq 0, \\
&(\alpha_2 - \beta)[x_2 - w_{12}] + \gamma_1(v_{2,1} - v_{12,1}) + \gamma_2(v_{2,2} - v_{12,2}) \geq 0, \\
&\beta[-1 + (x_1 + x_2) - w_{12}] + \gamma_1[y_1 - (v_{1,1} + v_{2,1}) + v_{12,1}] \\
&\qquad + \gamma_2[y_2 - (v_{1,2} + v_{2,2}) + v_{12,2}] \geq 0,
\end{aligned}\right\} \tag{2.9a}$$

$$w_{12} \geq 0, x_1 - w_{12} \geq 0, x_2 - w_{12} \geq 0, 1 - (x_1 + x_2) + w_{12} \geq 0, \tag{2.9b}$$

$$\left.\begin{aligned}
&w_{12} \geq v_{12,k} \geq 0, (x_1 - w_{12}) \geq (v_{1,k} - v_{12,k}) \geq 0,\\
&(x_2 - w_{12}) \geq (v_{2,k} - v_{12,k}) \geq 0,\\
&\text{and } [1 - (x_1 + x_2) + w_{12}] \geq [y_k - (v_{1,k} + v_{2,k}) + v_{12,k}] \geq 0,\\
&\quad \text{for } k = 1, 2\}.
\end{aligned}\right\} \tag{2.9c}$$

Some comments and illustrations are in order at this point. First, note that we could have computed inter-products of other defining constraints in constructing the relaxation at level *d*, so long as the degree of the intermediate polynomial program generated at the Reformulation step remains the same and no nonlinearities are created with respect to the *y*-variables, so that linearity would be preserved upon using the substitution (2.4). While the additional inequalities thus generated would possibly yield tighter relaxations for levels $1, \ldots, n-1$, by our convex hull assertion in (2.7), these constraints would be implied by those defining X_n. Hence, our hierarchy results can be directly extended to include such additional constraints, but for simplicity, we omit these types of constraints. Nonetheless, we address the issue of the validity of including such constraints, among others, at the end of Section 2.3, and note that one may include them in a computational scheme employing the sets X_d, $d < n$.

Second, as alluded to earlier, note that for the case $d = 0$, using the fact that $f_0(\varnothing, \varnothing) \equiv 1$, that $f_0^k(\varnothing, \varnothing) \equiv y_k$ for $k = 1, \ldots, m$, and that $f_1(j, \varnothing) \equiv x_j$ and $f_1(\varnothing, j) \equiv (1 - x_j)$ for $j = 1, \ldots, n$, it follows that X_0 given by (2.5) is precisely the continuous (LP) relaxation of X in which the integrality restrictions on the

x-variables are dropped. Finally, for $d = n$, note that the inequalities (2.5b) are implied by (2.5c) and can therefore be omitted from the representation X_n.

The following section establishes the fact that for $d = 0, 1, \ldots, n$, the sets X_d represent a sequence of nested, valid relaxations leading up to the convex hull representation.

2.2. Validity of the Hierarchy of Relaxations and the Convex Hull Representation

This section establishes the main result of the chapter that is embodied in Equation (2.7). The lemma given below first sets up a hierarchy of implications with respect to constraints (2.5b) and (2.5c) via a surrogation process.

Lemma 2.1. *For any* $d \in \{0, \ldots, n-1\}$, *the constraints* $f_{d+1}(J_1, J_2) \geq 0$ *for all* (J_1, J_2) *of order* $(d+1)$ *imply that* $f_d(J_1, J_2) \geq 0$ *for all* (J_1, J_2) *of order d. Similarly, the constraints* $f_{d+1}(J_1, J_2) \geq f^k_{d+1}(J_1, J_2) \geq 0$ *for all* $k = 1, \ldots, m$, *and* (J_1, J_2) *of order* $(d+1)$ *imply that* $f_d(J_1, J_2) \geq f^k_d(J_1, J_2) \geq 0$ *for all* $k = 1, \ldots, m$, *and* (J_1, J_2) *of order d.*

Proof. Consider any (J_1, J_2) of order d with $0 \leq d < n$ and any $k \in \{1, \ldots, m\}$, and let $t \in N - (J_1 \cup J_2)$. Then we have,

$$\begin{aligned} &F_{d+1}(J_1 + t, J_2) + F_{d+1}(J_1, J_2 + t) = F_d(J_1, J_2), \\ &y_k F_{d+1}(J_1 + t, J_2) + y_k F_{d+1}(J_1, J_2 + t) = y_k F_d(J_1, J_2). \end{aligned} \tag{2.10}$$

It is readily seen that these equations are preserved upon using the substitution (2.4) so that we also have,

$$\begin{aligned} &f_{d+1}(J_1 + t, J_2) + f_{d+1}(J_1, J_2 + t) = f_d(J_1, J_2), \\ &f_{d+1}^k(J_1 + t, J_2) + f_{d+1}^k(J_1, J_2 + t) = f_d^k(J_1, J_2). \end{aligned} \tag{2.11}$$

The required result now follows from (2.11), and the proof is complete. □

The equivalence of X to X_d for any $d \in \{0, \ldots, n\}$ under integrality restrictions on the x-variables, and the hierarchy among the relaxations are established next.

Theorem 2.1. *Let X_{Pd} denote the projection of the set X_d onto the space of the (x, y) variables as defined by (2.6), for $d = 0, 1, \ldots, n$. Then*

$$conv(X) \subseteq X_{Pn} \subseteq X_{P(n-1)} \subseteq \cdots \subseteq X_{P1} \subseteq X_{P0} \equiv X_0. \tag{2.12}$$

In particular, $X_{Pd} \cap \{(x, y)$: x binary$\} \equiv X$ for all $d = 0, 1, \ldots, n$.

Proof. Consider any $d \in \{1, \ldots, n\}$, and let $\{(x, y, w, v)\} \in X_d$. We will show that this same solution (using the components which appear in X_{d-1}) satisfies X_{d-1}, hence implying that $X_{Pd} \subseteq X_{P(d-1)}$. By Lemma 2.1, we have that the constraints (2.5b) and (2.5c) defining X_{d-1} are satisfied, and hence let us show by a similar surrogation process that the constraints (2.5a) are also satisfied. Toward this end, consider any (J_1, J_2) of order $(d-1)$, and any $r \in \{1, \ldots, R\}$. For any $t \notin (J_1 \cup J_2)$, by summing the two inequalities in (2.5a) corresponding to the sets $(J_1, J_2 + t)$ and $(J_1 + t, J_2)$ of order d, and using (2.11), we obtain the constraint (2.5a) for X_{d-1} corresponding to the set (J_1, J_2) of order $(d-1)$. Hence, $X_{Pn} \subseteq X_{P(n-1)} \subseteq \ldots \subseteq X_0$.

Next, let us show that $conv(X) \subseteq X_{Pn}$. If $X = \emptyset$, this is trivial. Otherwise, given any $(x, y) \in X$, define w_J and v_{Jk} for all $J \subseteq N$, $k = 1, \ldots, m$, as in (2.4). Then, by construction, $(x, y, w, v) \in X_n$. Hence $X \subseteq X_{Pn}$, and since X_{Pn} is convex, we have $conv(X) \subseteq X_{Pn}$, and so (2.12) holds true. Finally, since $X \equiv conv(X) \cap \{(x, y)\colon\ x \text{ binary}\} \equiv X_0 \cap \{(x, y)\colon\ x \text{ binary}\}$, it follows from (2.12) that $X_{Pd} \cap \{(x, y)\colon\ x \text{ binary}\} \equiv X$ for all $d = 0, 1, \ldots, n$, and this completes the proof. □

Hence, by Theorem 2.1, we see that for any $d \in \{0, 1, \ldots, n\}$, the set X_{Pd} is a polyhedral relaxation of *X* in that it contains *X* and is equivalent to *X* if the *x*-variables are enforced to be binary valued. Moreover, the sets X_{Pd} are all nested, one within the previous set as *d* varies from 0 to *n*, beginning with the ordinary continuous relaxation $X_{P0} \equiv X_0$. In fact, as stated in (2.7) and as shown in the sequel, the final relaxation X_{Pn} coincides with $conv(X)$. But first, let us introduce the following transformation which we shall find useful throughout this analysis.

Lemma 2.2. *Consider the affine transformation:* $\{w_J, J \subseteq N\} \to \{U_J^0, J \subseteq N\}$ *defined by*

$$U_J^0 = f_n(J, \bar{J}) \equiv \sum_{J' \subseteq \bar{J}} (-1)^{|J'|} w_{J \cup J'} \quad \textit{for all } J \subseteq N, \tag{2.13a}$$

where $\bar{J} = N - J$ *for* $J \subseteq N$. *This transformation is nonsingular with inverse*

$$w_J = \sum_{J' \subseteq \bar{J}} U^0_{J \cup J'} \text{ for all } J \subseteq N, \tag{2.13b}$$

where as defined in (2.4b), $w_\varnothing \equiv 1$, *and* $w_j \equiv x_j$ *for* $j = 1, \dots, n$.

Similarly, for each $k = 1, \dots, m$, *consider the linear transformation:* $\{v_{Jk}, J \subseteq N\} \to \{U^k_J, J \subseteq N\}$ *defined by*

$$U^k_J = f^k_n(J, \bar{J}) \equiv \sum_{J' \subseteq \bar{J}} (-1)^{|J'|} v_{(J \cup J')k} \text{ for all } J \subseteq N. \tag{2.14a}$$

Then this defines a nonsingular transformation with inverse

$$v_{Jk} = \sum_{J' \subseteq \bar{J}} U^k_{J \cup J'} \text{ for all } J \subseteq N, \tag{2.14b}$$

where as defined in (2.4b), $v_{\varnothing k} \equiv y_k$ *for* $k = 1, \dots, m$.

In particular, under (2.13) and (2.14), we have

$$x_j = \sum_{J \subseteq N : j \in J} U^0_J \text{ for } j = 1, \dots, n \text{ and } y_k = \sum_{J \subseteq N} U^k_J \text{ for } k = 1, \dots, m. \tag{2.15}$$

Proof. Note that from (2.13a), where the expression for $f_n(J, \bar{J})$ follows easily from (2.2), the sum in (2.13b) is given by

$$\sum_{J' \subseteq \bar{J}} U^0_{J \cup J'} = \sum_{J' \subseteq \bar{J}} \sum_{K \subseteq \overline{J \cup J'}} (-1)^{|K|} w_{J \cup J' \cup K}$$

$$= \sum_{H \subseteq \bar{J}} \left[\sum_{K \subseteq H} (-1)^{|K|} \right] w_{J \cup H} = w_J. \tag{2.16}$$

The last step follows because the sum $\sum_{K \subseteq H} (-1)^{|K|}$ in (2.16) equals 0 whenever $H \neq \emptyset$, and equals 1 when $H = \emptyset$. Hence, given the system (2.13a), we see from (2.16) that the system (2.13b) must be satisfied, yielding a unique solution. This proves the assertion involving (2.13).

In an identical fashion, the system (2.14a) is equivalent to (2.14b). Finally, noting that (2.15) simply rewrites (2.13b) for $J = \{j\}, j = 1, \ldots, n$, and (2.14b) for $J = \emptyset$, $k = 1, \ldots, m$, the proof is complete. □

Example 2.3. To illustrate Lemma 2.2, consider a situation with $n = 3$ and let us verify the transformation (2.14), for example. Then for any $k \in \{1, \ldots, m\}$, the system (2.14a) is of the form

$$U^k_{123} = v_{123k},\ U^k_{12} = v_{12k} - v_{123k},\ U^k_{13} = v_{13k} - v_{123k},\ U^k_{23} = v_{23k} - v_{123k},$$

$$U^k_1 = v_{1k} - (v_{12k} + v_{13k}) + v_{123k},\ U^k_2 = v_{2k} - (v_{12k} + v_{23k}) + v_{123k},$$

$$U^k_3 = v_{3k} - (v_{13k} + v_{23k}) + v_{123k},$$

$$U^k_\emptyset = y_k - (v_{1k} + v_{2k} + v_{3k}) + (v_{12k} + v_{13k} + v_{23k}) - v_{123k}$$

The inverse transformation (2.14) is as follows:

$$v_{123k} = U^k_{123},\ v_{12k} = U^k_{12} + U^k_{123},\ v_{13k} = U^k_{13} + U^k_{123},\ v_{23k} = U^k_{23} + U^k_{123},$$

$$v_{1k} = U^k_1 + U^k_{12} + U^k_{13} + U^k_{123},\ v_{2k} = U^k_2 + U^k_{12} + U^k_{23} + U^k_{123},$$

$$v_{3k} = U^k_3 + U^k_{13} + U^k_{23} + U^k_{123},$$

$$v_{\varnothing k} \equiv y_k = U^k_{\varnothing} + U^k_1 + U^k_2 + U^k_3 + U^k_{12} + U^k_{13} + U^k_{23} + U^k_{123}.$$

The following theorem now provides the desired convex hull characterization.

Theorem 2.2. *Let the polyhedral relaxation X_{Pn} of X be as defined by (2.5) and (2.6). Then $X_{Pn} \equiv conv(X)$.*

Proof. By Theorem 2.1, we need to show that $X_{Pn} \subseteq conv(X)$. If $X_{Pn} = \varnothing$, then this result is trivial, and so we assume that $X_{Pn} \neq \varnothing$. Since X_{Pn} is bounded, with $X_{Pn} \subseteq X_0$ by Theorem 2.1, we only need to show that x is binary valued at all extreme points (x, y) of X_{Pn}. Equivalently, we need to show that the linear program

$$\text{LP: maximize } \left\{ \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{m} d_k y_k : \ (x, y) \in X_{Pn} \right\} \tag{2.17a}$$

has an optimal solution at which x is binary for any objective function $(cx + dy)$. Noting the definition of X_{Pn} given via (2.5) and (2.6) with $d = n$, and recalling the redundancy of (2.5b) when $d = n$, we may write (2.17a) as follows.

LP: Maximize $\displaystyle\sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{m} d_k y_k$

subject to $$\left[\sum_{j \in J} \alpha_{rj} - \beta_r \right] f_n(J, \bar{J}) + \sum_{k=1}^{m} \gamma_{rk} f^k_n(J, \bar{J}) \geq 0$$

$$\text{for all } r = 1, \ldots, R, \ J \subseteq N, \tag{2.17b}$$

$$f_n(J, \bar{J}) \geq f^k_n(J, \bar{J}) \geq 0 \quad \text{for all } k = 1, \ldots, m, \ J \subseteq N.$$

Now, consider the nonsingular transformations given by (2.13) and (2.14). Noting (2.15), and using Lemma 2.2, the linear program LP given in (2.17b) gets equivalently transformed into the following problem.

$$\textbf{LP:}\ \text{Maximize}\quad \sum_{J \subseteq N} c_J^0 U_J^0 + \sum_{J \subseteq N} \sum_{k=1}^{m} d_k U_J^k \tag{2.18a}$$

$$\text{subject to}\quad \sum_{k=1}^{m} \gamma_{rk} U_J^k \geq \delta_{Jr} U_J^0 \quad \text{for all } r = 1, \ldots, R,\ J \subseteq N, \tag{2.18b}$$

$$\sum_{J \subseteq N} U_J^0 = 1, \tag{2.18c}$$

$$0 \leq U_J^k \leq U_J^0 \quad \text{for all } k = 1, \ldots, m,\ J \subseteq N, \tag{2.18d}$$

where

$$c_J^0 \equiv \sum_{j \in J} c_j \qquad \text{for all } J \subseteq N \text{ with } c_{\varnothing}^0 = 0,$$

$$\delta_{Jr} \equiv \beta_r - \sum_{j \in J} \alpha_{rj} \quad \text{for all } r = 1, \ldots, R,\ J \subseteq N, \tag{2.18e}$$

and where (2.13b) with $|J| = 1$ and (2.14b) with $J = \varnothing$ give the optimal solution to (2.17) that corresponds to an optimum for (2.18). (Above, note that (2.18c) corresponds to (2.13b) for $J = \varnothing$.)

Now, projecting onto the space of the variables U_J^0, $J \subseteq N$, and defining

$$S_0 = \left\{ U^0 \equiv (U_J^0, J \subseteq N) : \sum_{J \subseteq N} U_J^0 = 1,\ U^0 \geq 0 \right\} \tag{2.19a}$$

and for all $J \subseteq N$, letting

$$\Delta_J \equiv \max\{\sum_{k=1}^{m} d_k U_J^k: \sum_{k=1}^{m} \gamma_{rk} U_J^k \geq \delta_{Jr} \text{ for } r = 1, \ldots, R,$$

$$0 \leq U_J^k \leq 1 \text{ for } k = 1, \ldots, m\}, \tag{2.19b}$$

where $\Delta_J \equiv -\infty$ if (2.19b) is infeasible, it is readily seen that problem (2.18) is equivalent to the problem

$$\text{maximize} \left\{ \sum_{J \subseteq N} (c_J^0 + \Delta_J) U_J^0: \ U^0 \in S_0 \right\}. \tag{2.20}$$

(The equivalence follows by noting that for a fixed $U^0 \in S_0$, the problem (2.18) decomposes into separable problems over $J \subseteq N$, with each such problem being given by (2.19b) in which all the right-hand sides are multiplied by the corresponding scalar U_J^0.)

Now, since $X_{Pn} \neq \emptyset$ by assumption, $\Delta_J > -\infty$ for at least some $J \subseteq N$ in (2.19b). Noting (2.19a), we have at optimality in (2.20), that $U_{J*}^0 = 1$ for some $J* \subseteq N$, and $U_J^0 = 0$ for $J \subseteq N$, $J \neq J*$. Accordingly, from (2.18), $U_J^k = 0$ for $k = 1, \ldots, m$ for all $J \subseteq N$, $J \neq J*$, while U_{J*}^k, $k = 1, \ldots, m$, are given at optimality by the solution U_{J*}^{*k}, $k = 1, \ldots, m$, to the problem in (2.19b) for $J = J*$. Hence, from (2.15), we obtain at optimality for LP that

$$x_j = \begin{cases} 1 & \text{if } j \in J* \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } j = 1, \ldots, n \text{ and}$$

$$y_k = U_{J*}^{*k} \text{ for } k = 1, \ldots, m. \tag{2.21}$$

Since x is binary valued at optimality, this completes the proof. □

Equation (2.7) has thus been established. A few remarks are pertinent at this point.

Remark 2.1. Observe that the upper bounds on the y-variables are not needed for (2.7) to hold true, and that the foregoing analysis holds by simply eliminating any RLT product constraints that correspond to nonexistent bounding restrictions. Given feasibility of the underlying mixed-integer program, the proof of Theorem 2.2 asserts that x is binary valued at an optimum to any linear program $\max\{cx + dy\colon \ (x, y) \in X_{Pn}\}$ for which there exists an optimal solution, that is, for which this LP is not unbounded. Hence, by an identical argument, we have that (2.7) holds true in this case as well. □

Remark 2.2. A direct consequence of the convex hull result stated in (2.7) is that the convex hull representation over subsets of the x-variables can be computed in an identical fashion. In particular, suppose that the first $p \le n$ variables are treated as binary in the set X of (2.1), with the remaining $(n - p)$ x-variables relaxed to be continuous. Then it directly follows that the p^{th} level linearization of this relaxed problem produces $\text{conv}\{X_0 \cap \{(x, y)\colon \ x_j \text{ is binary } \forall\ j = 1, \dots, p\}\}$. The special case in which $p = 1$ precisely recovers the main step in the convexification argument of Balas *et al.* (1993) mentioned in Chapter 1. □

2.3. Further Insights Into the Structure of the Relaxations

In this section, we begin by presenting two lemmas which show that if x is binary in X_d, then the identity (2.4) holds precisely due to the constraints (2.5b) and (2.5c) defining X_d, while the remaining constraints serve to contain such a corresponding solution within X. Along with Theorem 2.2, this enables us to present a more complete characterization of the vertices of X_n, and permits us to suggest further strategies for tightening the intermediate relaxations while maintaining the hierarchy. We begin by focusing in the Lemma below on inequalities (2.5c).

Lemma 2.3. *For any* $d \in (1, \ldots, n\}$, *define the following set composed from the constraints (2.5c).*

$$Z_d = \{(x, y, w, v)\colon\ f_d(J_1, J_2) \geq f_d^k(J_1, J_2) \geq 0 \ \textit{for } k = 1, \ldots, m$$

$$\textit{and for each } (J_1, J_2) \textit{ of order } d\}. \tag{2.22}$$

Let $\hat{x}$ *be any binary vector. Then* $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in Z_d$ *if and only if*

$$0 \leq \hat{y} \leq e_m \tag{2.23a}$$

and

$$\hat{w}_J = \prod_{j \in J} \hat{x}_j \ \textit{ and } \ \hat{v}_{Jk} = \hat{y}_k \prod_{j \in J} \hat{x}_j,$$

$$\textit{for all } k = 1, \ldots, m \textit{ and } J \subseteq N \textit{ with } |J| = 1, \ldots, d. \tag{2.23b}$$

Proof. If $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ satisfies (2.23) with $\hat{x}$ binary, then the values of $f_d(J_1, J_2)$ and $f_d^k(J_1, J_2)$ match those of $F_d(J_1, J_2)$ and $y_k F_d(J_1, J_2)$, respectively, and so $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in Z_d$ with $\hat{x}$ binary. Conversely, let $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in Z_d$. Let us show by induction on d that (2.23) holds true. Note that for $d = 1$, the set Z_1 has constraints

$$x_j \geq v_{jk} \geq 0 \text{ and } (1 - x_j) \geq (y_k - v_{jk}) \geq 0 \text{ for } j = 1, \ldots, n,\ k = 1, \ldots, m.$$

Hence, $0 \leq \hat{y}_k \leq 1$, and for any $j = 1, \ldots, n$, if $\hat{x}_j = 0$, then $\hat{v}_{jk} = 0 = \hat{y}_k \hat{x}_j$, for $k = 1, \ldots, m$, while if $\hat{x}_j = 1$, then $\hat{v}_{jk} = \hat{y}_k = \hat{y}_k \hat{x}_j$ for $k = 1, \ldots, m$. Moreover, since $\hat{w}_j \equiv \hat{x}_j$, $j = 1, \ldots, n$ from (2.4b), we have that (2.23) holds true. Therefore, the result is true for Z_1. Now, assume that it is true for $Z_1, \ldots, Z_{d-1}$, and consider the set Z_d for any $d \in \{2, \ldots, n\}$.

Observe by Lemma 2.1, that the set Z_d enforces the constraints $f_{d-1}(J_1, J_2) \geq f_{d-1}^k(J_1, J_2) \geq 0$ for all (J_1, J_2) of order $(d - 1)$, and so by the induction hypothesis, (2.23a) holds true, and moreover, (2.23b) holds true for all $J \subseteq N$ such that $|J| \in \{1, \ldots, d - 1\}$. Hence, consider any $J \subseteq N$ with $|J| = d$, and let us show that (2.23b) holds true for this case as well.

Note that for any distinct $s, t \in J$, since $F_d(J - t, t) = (1 - x_t)\prod_{j \in J - t} x_j$ and $F_d(J - s - t, \{s, t\}) = (1 - x_s - x_t + x_s x_t)\prod_{j \in J - \{s,t\}} x_j$, we have from the constraint $f_d(J - t, t) \geq 0$ that $w_J \leq w_{J-t}$, and from the constraint $f_d(J - s - t, \{s, t\}) \geq 0$ that $w_J \geq w_{J-s} + w_{J-t} - w_{J-s-t}$. Using the induction hypothesis, this means that

$$\hat{w}_J \le \prod_{j \in J-t} \hat{x}_j \text{ and } \hat{w}_J \ge \prod_{j \in J-s} \hat{x}_j + \prod_{j \in J-t} \hat{x}_j - \prod_{j \in J-s-t} \hat{x}_j \text{ for all } s, t \in J. \tag{2.24}$$

Now, suppose that $\prod_{j \in J} \hat{x}_j = 0$. Then from the first inequality in (2.24) and the restriction that $w_J \equiv f_d(J, \varnothing) \ge f_d^k(J, \varnothing) \equiv v_{Jk} \ge 0$, we have that $\hat{w}_J = \hat{v}_{Jk} = 0$ for $k = 1, \ldots, m$, and so (2.23b) holds true. On the other hand, suppose that $\prod_{j \in J} \hat{x}_j = 1$. Then from (2.24), we have $\hat{w}_J = 1$. Moreover, for any $t \in J$ and $k \in \{1, \ldots, m\}$, the constraint $f_d(J-t, t) \ge f_d^k(J-t, t) \ge 0$ in (2.22) yields $(\hat{w}_{J-t} - \hat{w}_J) \ge (\hat{v}_{(J-t)k} - \hat{v}_{Jk}) \ge 0$. Since $\hat{w}_{J-t} = \hat{w}_J = 1$, we have $\hat{v}_{Jk} = \hat{v}_{(J-t)k}$. But by the induction hypothesis, since (2.23b) holds true for $(J-t)$ as $|J-t| = d-1$, we have, $\hat{v}_{(J-t)k} = \hat{y}_k \prod_{j \in J-t} \hat{x}_j = \hat{y}_k$, and so $\hat{v}_{Jk} = \hat{y}_k$. Therefore, if $\prod_{j \in J} \hat{x}_j = 1$, then $\hat{w}_J = 1$ and $\hat{v}_{Jk} = \hat{y}_k$ for $k = 1, \ldots, m$, and so again (2.23b) holds true. This completes the proof. □

Lemma 2.4. *Consider any* $d \in \{1, \ldots, n\}$, *and let* $\hat{x}$ *be any binary vector. Then* $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in X_d$ *if and only if*

$$(\hat{x}, \hat{y}) \in X, \hat{w}_J = \prod_{j \in J} \hat{x}_j \text{ for all } J \subseteq N \text{ with } |J| = 1, \ldots, D \equiv \min\{d+1, n\}$$

and

$$\hat{v}_{Jk} = \hat{y}_k \prod_{j \in J} \hat{x}_j \text{ for all } k = 1, \ldots, m \text{ and } J \subseteq N \text{ with } |J| = 0, 1, \ldots, d. \tag{2.25}$$

Proof. For any $d \in \{1, \ldots, n\}$, and $\hat{x}$ binary, if $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in X_d$, then since $(\hat{x}, \hat{y}) \in X_{Pd}$, $X_{Pd} \subseteq X_0$ by Theorem 2.1, and $X \equiv X_0 \cap \{(x, y): \ x \text{ binary}\}$, we have that $(\hat{x}, \hat{y}) \in X$. Moreover, noting (2.5b) he type (2.5c), we have from Lemma

2.3 that the other conditions in (2.25) hold true as well. Conversely, if (2.25) holds true, then $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in X_d$ by construction, and the proof is complete. □

Theorem 2.3. *The solution* $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ *is a vertex of* X_n *if and only if* $\hat{x}$ *is binary valued,* $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ *for all* $J \subseteq N$, $J \neq \emptyset$, $\hat{v}_{Jk} = \hat{y}_k \prod_{j \in J} \hat{x}_j$ *for all* $J \subseteq N$, $J \neq \emptyset$, $k = 1, \ldots, m$, *and* $(\hat{y}_1, \ldots, \hat{y}_m)$ *is an extreme point of the set* $\hat{Y} \equiv \{y : (\hat{x}, y) \in X\}$.

Proof. Let $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ be a vertex of X_n. Then there exists a linear objective function defined on the (x, y, w, v) space such that the maximum of this function over X_n occurs uniquely at $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$. Now, following the approach in Theorem 2.2, under the transformations (2.13) and (2.14) of Lemma 2.2, the foregoing linear program can be put into the form (2.18) with appropriately defined objective coefficients. Consequently, from (2.21), the unique optimum $\hat{x}$ must be binary valued. Hence, from Lemma 2.4, $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$, $J \neq \emptyset$, and $\hat{v}_{Jk} = \hat{y}_k \prod_{j \in J} \hat{x}_j$ for all $k = 1, \ldots, m$, $J \subseteq N$, $J \neq \emptyset$. Furthermore, from (2.19b) and (2.21), the (unique) optimum $\hat{y}$ is obtained as the solution U_{J*}^{*k}, $k = 1, \ldots, m$ to (2.19b) for some $J = J*$. Noting in (2.19b) that $\delta_{J*r} \equiv \beta_r - \sum_{j \in J*} \alpha_{rj} = \beta_r - \sum_{j=1}^{n} \alpha_{rj} \hat{x}_j$ from (2.18e) and (2.21), we obtain $\hat{y}$ as the unique solution to a linear program over the polyhedron $Y \equiv \{y: \sum_{k=1}^{m} \gamma_{rk} y_k \geq (\beta_r - \sum_{j=1}^{n} \alpha_{rj} \hat{x}_j), \ 0 \leq y \leq e_m\}$. Hence, $\hat{y}$ is a vertex of $\hat{Y}$.

Conversely, suppose that we are given $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ satisfying the conditions stated in Theorem 2.3. By Lemma 2.4, $(\hat{x}, \hat{y}, \hat{w}, \hat{v}) \in X_n$, and in particular, $(\hat{x}, \hat{y}) \in X_{Pn}$. It is sufficient to show that $(\hat{x}, \hat{y})$ is an extreme point of X_{Pn} since by Lemma 2.4, for feasibility to X_n, we must uniquely have $w = \hat{w}$ and $v = \hat{v}$ as the completion to this vertex. To accomplish this, we show that for $(\hat{x}, \hat{y})$ to satisfy $(\hat{x}, \hat{y}) = \lambda(\bar{x}, \bar{y}) + (1 - \lambda)(\tilde{x}, \tilde{y})$ for some $\lambda \in (0,1)$, $(\bar{x}, \bar{y}) \in X_{Pn}$ and $(\tilde{x}, \tilde{y}) \in X_{Pn}$, we must have $(\hat{x}, \hat{y}) = (\bar{x}, \bar{y}) = (\tilde{x}, \tilde{y})$. Observe that since $X_{Pn} \subseteq X_0$ by Theorem 2.1, we have $0 \leq \bar{x} \leq e_n$ and $0 \leq \tilde{x} \leq e_n$. Since $\hat{x}$ is binary, for any $\lambda \in (0,1)$ we get $\hat{x} = \bar{x} = \tilde{x}$. Using this result along with the supposition that $\hat{y}$ is an extreme point of $\hat{Y}$, we deduce that $\hat{y} = \bar{y} = \tilde{y}$, and the proof is complete. □

Note that Theorem 2.3 essentially asserts that in projecting the set X_n from the (x, y, w, v) space to the set X_{Pn} in the (x, y) space, all extreme points are preserved. If $(\hat{x}, \hat{y})$ is a vertex of X_{Pn}, then by Theorem 2.2 and Lemma 2.4, $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ as defined by the theorem is a vertex of X_n. Conversely, if $(\hat{x}, \hat{y}, \hat{w}, \hat{v})$ is a vertex of X_n, then it satisfies the conditions stated in the theorem and, as shown in the proof, yields $(\hat{x}, \hat{y})$ as a vertex of X_{Pn}.

Remark 2.4. In addition to lending insight into the vertices of X_n, Lemmas 2.3 and 2.4 provide the essential ingredients for extending the RLT constructs to accommodate mixed 0-1 polynomial programs in the variables (x, y), provided cross-product terms do not exist in the continuous variables y esult, a hierarchy of mixed-integer linear relaxations satisfying (2.7) and Theorem 2.3 for such programs can be readily obtained, provided that the product factors of the type (2.5b) and (2.5c) are computed to a sufficiently high-level d so as to linearize all the nonlinear terms. Such nonlinear terms maye present in the set X and/or the objective function. Indeed, level-1 type linearizations were considered by Adams (1985) and Adams and Sherali (1986, 1990, 1993) for programs having quadratic objective functions, and by Adams and Johnson (1994) and Johnson (1992) specialized for the quadratic assignment problem. More discussion is found in Section 2.5.

In concluding this section, let us comment on the situation in which there exist certain constraints from the first set of inequalities in (2.1) which involve only the x-variables. As mentioned in Section 2.1, one can multiply such constraints with the factors y_k and $(1 - y_k)$ for $k = 1, \ldots, m$, and then linearize the resulting constraints using (2.4) as with the other constraints (2.3). By Lemma 2.4, these constraints are implied when x is binary in any feasible solution, and moreover, since they serve to tighten the continuous relaxation, Theorem 2.1 continues to hold. Furthermore, since X_n has x binary for all vertices by Theorem 2.3, and X_n is bounded, these constraints are implied by the other constraints defining X_n by Lemma 2.4. Consequently, Theorems 2.2 and 2.3 also continue to hold with the inclusion of such constraints. In a likewise fashion, inequalities defining X that involve only the x-variables can be used in the same spirit as the factors $x_j \geq 0$ and $(1 - x_j) \geq 0$ to generate products of inequalities up to a given order level in order to further tighten intermediate level relaxations. Of course, by virtue of Lemma 2.4 and Theorem 2.3, such constraints are again all implied by the constraints defining X_n.

2.4. Characterization of the Facets of the Convex Hull of Feasible Solutions

We will now derive a characterization for the facets of $X_{Pn} \equiv conv(X)$ using a projection operation. Since under a nonsingular linear transformation, the extreme points, facets, and the boundedness of a polyhedron are preserved, we will conveniently use the form (2.18b)-(2.18d) obtained under the transformation (2.13) and (2.14) of Lemma 2.2 to represent the set X_{Pn}. Noting (2.15), we may therefore write

$X_{Pn} = \{(x, y):$

$$x_j = \sum_{J \subseteq N: j \in J} U_J^0 \text{ for } j = 1, \ldots, n, \tag{2.26a}$$

$$y_k = \sum_{J \subseteq N} U_J^k \text{ for } k = 1, \ldots, m, \tag{2.26b}$$

$$\sum_{k=1}^{m} \gamma_{rk} U_J^k \geq \delta_{Jr} U_J^0 \text{ for } r = 1, \ldots, R,\ J \subseteq N, \tag{2.26c}$$

$$\sum_{J \subseteq N} U_J^0 = 1, \tag{2.26d}$$

$$0 \leq U_J^k \leq U_J^0 \text{ for } k = 1, \ldots, m,\ J \subseteq N\}. \tag{2.26e}$$

Associating Lagrange multipliers π_j, $j = 1, \ldots, n$, λ_k, $k = 1, \ldots, m$, θ_{Jr}, $r = 1, \ldots, R$, $J \subseteq N$, π_0, and ψ_{Jk}, $k = 1, \ldots, m$, $J \subseteq N$ with respect to the constraints (2.26a), (2.26b), (2.26c), (2.26d), and the variable upper bounding constraints in (2.26e), respectively, we have that $(x, y) \in X_{Pn}$ if and only if (2.26) yields a feasible solution given that this (x, y) is fixed in value, which holds true by linear programming duality if and only if

$$0 = \max\{\pi x + \lambda y + \pi_0 : \ (\pi, \lambda, \theta, \pi_0, \psi) \in PC\} \tag{2.27}$$

where *PC* is a polyhedral cone defined by the dual constraints to (2.26) as follows:

$$PC = \{(\pi, \lambda, \theta, \pi_0, \psi):$$

$$\sum_{j \in J} \pi_j - \sum_{r=1}^{R} \delta_{Jr} \theta_{Jr} + \pi_0 + \sum_{k=1}^{m} \psi_{Jk} \le 0 \text{ for } J \subseteq N, \tag{2.28a}$$

$$\lambda_k + \sum_{r=1}^{R} \gamma_{rk} \theta_{Jr} - \psi_{Jk} \le 0 \text{ for } k = 1, \ldots, m, \ J \subseteq N, \tag{2.28b}$$

$$\theta_{Jr} \ge 0, \ r = 1, \ldots, R, \ J \subseteq N, \ \psi_{Jk} \ge 0, \ k = 1, \ldots, m, \ J \subseteq N\}. \tag{2.28c}$$

Now, consider the following result.

Theorem 2.4. *The set PC defined in (2.28) is an unbounded polyhedral cone with vertex at the origin and has some L distinct extreme directions or generators* $(\pi^\ell, \lambda^\ell, \theta^\ell, \pi_0^\ell, \psi^\ell)$, $\ell = 1, \ldots, L$, $L \ge 1$, *with* $\pi_0^\ell \equiv 0, \ +1,$ *or* -1 *by way of normalization. Moreover,*

$$X_{Pn} = \{(x, y): \ \pi^\ell x + \lambda^\ell y \le -\pi_0^\ell, \ \ell = 1, \ldots, L\}. \tag{2.29}$$

Proof. Noting that π, λ, and π_0 are unrestricted in sign in (2.28), *PC* is clearly unbounded. Furthermore, enforcing all the defining inequalities to be binding yields

$\theta \equiv 0$ and $\psi \equiv 0$ from (2.28c), which implies that $\lambda \equiv 0$ from (2.28b), and from (2.28a) for $J = \emptyset, \{1\}, \{2\}, \ldots, \{n\}$, we, respectively obtain $\pi_0 = 0$, $\pi_1 = 0, \ldots, \pi_n = 0$. Hence, this produces the origin as the unique feasible solution, and so there exist some q linearly independent defining hyperplanes in (2.28) which are binding at the origin, where q is the dimension of $(\pi, \lambda, \theta, \pi_0, \psi)$. Consequently, PC is a pointed polyhedral cone with the vertex at the origin, and has some L distinct extreme directions as stated in the theorem, each produced by some $(q-1)$ linearly independent hyperplanes binding from (2.28). Moreover, (2.27) holds true if and only if $\pi^\ell x + \lambda^\ell y + \pi_0^\ell \le 0$ for $\ell = 1, \ldots, L$, and hence, X_{Pn} is given by (2.29). This completes the proof. □

Corollary 2.1. *Alternatively, X_{Pn} is given by (2.29) where $(\pi^\ell, \lambda^\ell, \theta^\ell, \pi_0^\ell, \psi^\ell)$, $\ell = 1, \ldots, L$, are the extreme points of the set*

$$\overline{PC} = PC \cap \{(\pi, \lambda, \theta, \pi_0, \psi) \colon \sum_{J \subseteq N} [\sum_{j \in J} \pi_j + \sum_{k=1}^{m} \lambda_k + \pi_0 - \sum_{k=1}^{m} \psi_{Jk}$$

$$+ \sum_{r=1}^{R} \theta_{Jr} (\sum_{k=1}^{m} \gamma_{rk} - \delta_{Jr} - 1)] = -1\}. \tag{2.30}$$

Proof. This follows from the fact that the constraint imposed on PC in (2.30) is simply a regularization or normalization constraint on the generators of PC, noting that PC is a pointed cone. □

Note that if $X_{Pn} \neq \emptyset$, then the facet defining inequalities for X_{Pn} are among the constraints in (2.29) if X_{Pn} is full dimensional, and are given through appropriate intersections of these defining hyperplanes otherwise. Of course, a total enumeration of such constraints is prohibitive. However, the structure of X_{Pn} and PC given in (2.26) and (2.28), respectively, can be possibly exploited to generate various classes of facets via Theorem 2.4. This might be achievable by characterizing certain classes (not necessarily all) generators of PC for special types of problems. This feature is the principal potential utility of this characterization, and is illustrated later in Chapter 5. Furthermore, we can also generate specific valid inequalities for the problem by noting that any inequality $\pi x + \lambda y + \pi_0 \le 0$ is valid for X_{Pn} if and only if it can be obtained by surrogating the constraints (2.26) using a solution feasible to the (dual) system (2.28) defining PC. (Here, the surrogate multipliers to be used for the family of nonnegativity restrictions on U_j^k in (2.26) are the slacks in (2.28b).

2.5. Extension to Multilinear Polynomial Programs and Specializations for Equality Constraints

We conclude this chapter by presenting two important extensions. The first of these concerns multilinear mixed-integer zero-one *polynomial* programming problems in which the continuous variables $0 \le y \le e_m$ appear linearly in the constraints and the objective function. This is discussed below.

Extension 1. *Multilinear mixed-integer zero-one polynomial programming problems.*

Consider the set

$$X = \{(x, y): \sum_{t \in T_{r0}} \alpha_{rt} p(J_{1t}, J_{2t}) + \sum_{k=1}^{m} y_k \sum_{t \in T_{rk}} \gamma_{rkt} p(J_{1t}, J_{2t}) \geq \beta_r,$$

$$r = 1, \ldots, R, \; 0 \leq x \leq e_n \text{ and integer, } 0 \leq y \leq e_m\}, \tag{2.31}$$

where for all t, $p(J_{1t}, J_{2t}) \equiv [\prod_{j \in J_{1t}} x_j][\prod_{j \in J_{2t}} (1 - x_j)]$ are polynomial terms for the various sets (J_{1t}, J_{2t}), indexed by the set T_{r0} and T_{rk}, in (2.31). For $d = 0, 1, \ldots, n$, we can construct a polyhedral relaxation X_d for X by using the factors $F_d(J_1, J_2)$ to multiply the first set of constraints as before, where (J_1, J_2) are of order d. However, denoting δ_1 as the maximum degree of the polynomial terms in x not involving the y-variables, and δ_2 as the maximum degree of the polynomial terms in x that are associated with products involving y-variables, in lieu of (2.3b), we now use $F_{D_1}(J_1, J_2) \geq 0$ for (J_1, J_2) of order $D_1 = \min\{d + \delta_1, n\}$, and in lieu of (2.3c), we employ the constraints $F_{D_2}(J_1, J_2) \geq y_k F_{D_2}(J_1, J_2) \geq 0$, $k = 1, \ldots, m$, for all (J_1, J_2) of order $D_2 = \min\{d + \delta_2, n\}$. Of course, if $D_2 \geq D_1$, then the former restrictions are unnecessary, as they are implied by the latter. Note that in computing δ_1 and δ_2 in an optimization context, we consider the terms in the objective function as well, and that for the linear case, we have $\delta_1 = 1$ and $\delta_2 = 0$. Now, linearizing the resulting constraints under the substitution (2.4) produces the desired set X_d. Because of Lemma 2.3, when the integrality on the x-variables is enforced, each such set X_d is equivalent to the set X.

Moreover, by Lemma 2.1 and Equation (2.11), the proof of Theorem 2.1 continues to hold true. In particular, because of (2.11), each constraint from the first set of inequalities in X_d for any $d < n$ is obtainable by surrogating two appropriate constraints from X_{d+1} as in the proof of Theorem 2.1. Hence, we again obtain the hierarchy of relaxations $conv(X) \subseteq X_{Pn} \subseteq X_{P(n-1)} \subseteq \ldots \subseteq X_{P1} \subseteq X_0$. Furthermore, by Lemma 2.3, Lemma 2.4 also holds true for this situation.

Now, consider the set X_{Pn}. The constraints (2.3b) and (2.3c) for this set are

$$f_n(J,\bar{J}) \geq f_n^k(J,\bar{J}) \geq 0, \text{ for all } k = 1,\ldots,m,\ J \subseteq N. \tag{2.32a}$$

Furthermore, multiplying the first set of constraints defining X in (2.31) by the factors $F_n(J,\bar{J})$ for all $J \subseteq N$ produces, upon linearization through the substitution (2.4), the following inequalities:

$$\sum_{k=1}^{m} \bar{\gamma}_{rk} f_n^k(J,\bar{J}) \geq \delta_{Jr} f_n(J,\bar{J}) \text{ for } r = 1,\ldots,R,\ J \subseteq N, \tag{2.32b}$$

where $\delta_{Jr} = \beta_r - \{\sum_t \alpha_{rt} : t \in T_{r0}, J_{1t} \subseteq J, \text{ and } J_{2t} \subseteq \bar{J}\}$, for $r = 1,\ldots,R$ $J \subseteq N$, and where $\bar{\gamma}_{rk} = \{\sum_t \gamma_{rkt} :\ t \in T_{rk}, J_{1t} \subseteq J, \text{and } J_{2t} \subseteq \bar{J}\}$ for $r = 1,\ldots,R$ and $k = 1,\ldots,m$. Noting that (2.32) is precisely of the same form as the inequalities of X_{Pn} given by (2.17b) for the linear case, the proof of Theorem 2.2 establishes that $X_{Pn} \equiv conv(X)$ for this case as well. Moreover, Theorem 2.3 and the characterization of facets for $conv(X)$ continue to hold as for the linear case.

Extension 2. *Construction of Relaxations Using Equality Constraint Representations*

Note that given any set X of the form (2.1), by adding slack variables to the first R constraints, determining upper bounds on these slacks as the sum of the positive constraint coefficients minus the right-hand side constant, and accordingly scaling these slack variables onto the unit interval, we may equivalently write the set X as

$$X = \{(x, y): \sum_{j=1}^{n} \alpha_{rt} x_j + \sum_{k=1}^{m} \gamma_{rk} y_k + \gamma_{r(m+r)} y_{m+r} = \beta_r \text{ for } r = 1, \ldots, R,$$

$$0 \leq x \leq e_n, x \text{ integer}, 0 \leq y \leq e_{m+R}\}. \tag{2.33}$$

Now, for any $d \in \{1, \ldots, n\}$, observe that the factor $F_d(J_1, J_2)$ for any (J_1, J_2) of order d is a linear combination of the factors $F_p(J, \varnothing)$ for $J \subseteq N$, $p \equiv |J| = 0, 1, \ldots, d$. Hence, the constraint derived by multiplying an *equality* from (2.33) by $F_d(J_1, J_2)$ and then linearizing it via (2.4) is obtainable via an appropriate surrogate (with mixed-sign multipliers) of the constraints derived similarly, but using the factors $F_p(J, \varnothing)$ for $J \subseteq N$, $p \equiv |J| = 0, 1, ..., d$. Hence, these latter factors produce constraints that can generate the other constraints, and so X_d defined by (2.5) corresponding to X as in (2.33) is *equivalent* to the following, where $f_p(J, \varnothing) \equiv w_J$, and $f_k^p(J, \varnothing) \equiv v_{Jk}$ for all $p = 0, 1, ..., d + 1$ as in (2.4):

$X_d = \{(x, y, w, v)$: constraints of type (2.5b) and (2.5c) hold, and for $r = 1, ..., R$,

$$[\sum_{j \in J} \alpha_{rj} - \beta_r] w_J + \sum_{j \in \bar{J}} \alpha_{rj} w_{J+j} + \sum_{k=1}^{m} \gamma_{rk} v_{Jk} + \gamma_{r(m+r)} v_{J(m+r)} = 0$$

for all $J \subseteq N$ with $|J| = 0, 1, ..., d\}$. (2.34)

Note that the savings in the number of constraints in (2.34) over that in (2.5) corresponding to the set X as in (2.33) is given by

$$R\left[2^d\binom{n}{d} - \sum_{i=0}^{d}\binom{n}{i}\right].$$

Also, observe that for $J = \varnothing$, the equalities in (2.34) are precisely the original equalities defining X in (2.33). Hence, using Lemma 2.1, the assertion of Theorem 2.1 is directly seen to be true for (2.34). Of course, because (2.34) is equivalent to the set of the type (2.5) which would have been derived using the factors $F_d(J_1, J_2)$ of degree d, all the foregoing results continue to hold true for (2.34). However, establishing that $X_{Pn} = conv(X)$ and characterizing the facets of X_{Pn} is more conveniently managed using the constructs of Sections 2.2 and 2.4.

While the approach in Sections 2.2 and 2.4 developed for the inequality constrained case is convenient for theoretical purposes as it avoids the manipulation of surrogates that would be required for the equalities in (2.34), note that from a computational viewpoint, when $d < n$, the representation in (2.34) has fewer type (2.5a) "complicating" constraints and variables (including slacks in (2.5a)) than does (2.5) as given by the above savings expression, but has $R \times 2^d\binom{n}{d}$ additional constraints of the type (2.5c), counting the nonnegativity restrictions on the slacks in (2.5a) for the inequality constrained case. Hence, depending on the structure, either form of the representation of these relaxations may be employed, as convenient.

To summarize, when constructing the level d relaxation X_d in the presence of equality constraints, each equality constraint needs to be multiplied by the factors $f_p(J, \varnothing)$ for $J \subseteq N$, $p \equiv |J| = 0, 1, \ldots, d$ in the Reformulation step. Naturally, factors $f_p(J, \varnothing)$ that are known to be zeros, i.e., any such factor for which we know that there does not exist a feasible solution that has $x_j = 1 \ \forall\, j \in J$, need not be used in constructing these product constraints. The Linearization step is then applied as before.

Example 2.4. To illustrate both the foregoing extensions, let us consider the celebrated **quadratic assignment problem** defined as follows.

$$\textbf{QAP:}\ \text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k>i} \sum_{\ell \neq j} c_{ijk\ell} x_{ij} x_{k\ell} \tag{2.35a}$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_{ij} = 1 \quad \forall\, j = 1, \ldots, m \tag{2.35b}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \forall\, i = 1, \ldots, m \tag{2.35c}$$

$$x \geq 0 \text{ and integer.} \tag{2.35d}$$

First Level Relaxation:

To construct the relaxation at the first level, in addition to the constraints (2.35b, c), and the nonnegativity restrictions on x in (2.35d), we include the product constraints obtained by multiplying each constraint in (2.35b) and (2.35c) by each $x_{k\ell} \equiv F_1(\{x_{k\ell}\}, \varnothing)$, and then apply the rest of the RLT procedure as usual, using the substitution $w_{ijk\ell} = x_{ij} x_{k\ell}$

$\forall\ i, j, k, \ell$. Note that for any $j \in \{1, \ldots, m\}$, multiplying (2.35b) by x_{kj} for any $k \in \{1, \ldots, m\}$, produces upon using $x_{kj}^2 = x_{kj}$ that

$$\sum_{i \neq k} w_{ijkj} = 0 \quad \forall\ k = 1, \ldots, m$$

Hence, since $w \geq 0$, (2.36) implies that $w_{ijkj} \equiv 0\ \ \forall\ i \neq k$. Similarly, from (2.35c), we get $w_{iji\ell} = 0\ \ \forall\ j \neq \ell$. Furthermore, noting that $w_{ijk\ell} \equiv w_{k\ell ij}$, we only need to define $w_{ijk\ell}\ \ \forall\ i < k, j \neq \ell$. (For convenience in exposition below, however, we define $w_{ijk\ell}\ \ \forall\ i \neq k, j \neq \ell$ and explicitly impose $w_{ijk\ell} \equiv w_{k\ell ij}$.)

Additionally, since we only need to impose $x_{ij} \geq 0$ in (2.35d) for the continuous relaxation, since $x_{ij} \leq 1$ is then implied by the constraints (2.35b, c), we only need to multiply the factors $x_{k\ell} \geq 0$ and $(1 - x_{k\ell}) \geq 0$ with each constraint $x_{ij} \geq 0$ in the reformulation phase. This produces the restrictions $w_{ijk\ell} \leq x_{ij}$ and $w_{ijk\ell} \leq x_{k\ell}$ $\forall\ i, j, k, \ell$, along with $w \geq 0$. However, the former variable upper bounding constraints are implied by the constraints (2.36d, e, f) below. Hence, the first level relaxation (that would yield a lower bound on QAP) is given as follows.

Minimize $$\sum_i \sum_j \sum_{k>i} \sum_{\ell \neq j} c_{ijk\ell} w_{ijk\ell} \tag{2.36a}$$

subject to $$\sum_i x_{ij} = 1 \quad \forall\ j \tag{2.36b}$$

$$\sum_j x_{ij} = 1 \quad \forall\ i \tag{2.36c}$$

$$\sum_{i \neq k} w_{ijk\ell} = x_{k\ell} \quad \forall\ j, k, \ell \neq j \tag{2.36d}$$

$$\sum_{j \neq \ell} w_{ijk\ell} = x_{k\ell} \quad \forall\ i, \ell, k \neq i \tag{2.36e}$$

$$x \geq 0, w \geq 0, w_{ijk\ell} \equiv w_{k\ell ij} \quad \forall\ i \neq k, j \neq \ell. \tag{2.36f}$$

This level-1 relaxation (2.36b) was specifically introduced in Johnson (1992) and Adams and Johnson (1994), and shown to dominate all alternate linear formulations of QAP in terms of relaxation strength, as well as dominate a majority of bounding strategies. Computational results on solving this linear program via an interior point algorithm is presented by Resende *et al.* (1995).

Second Level Relaxation:

Following the same procedure as above, and eliminating null variables as well as null and redundant constraints, we would obtain the same relaxation as in (2.36) with additional constraints (and variables) corresponding to multiplying each equality constraint in (2.35b) by each factor $x_{k\ell}x_{pq}$ for $k < p, \ell \neq q \neq j$, and by multiplying each constraint in (2.35c) by each factor $x_{k\ell}x_{pq}$ for $k < p,\ k \neq i \neq p,\ \ell \neq q$. This would produce the additional constraints

$$\sum_{i \neq k,p} w_{ijk\ell pq} = w_{k\ell pq} \quad \forall\ j,k,\ell,p,q, \text{ where } k < p, \text{ and } \ell \neq q \neq j \tag{2.37a}$$

$$\sum_{j \neq \ell,q} w_{ijk\ell pq} = w_{k\ell pq} \quad \forall\ i,k,\ell,p,q, \text{ where } k < p, k \neq i \neq p, \text{ and } \ell \neq q \tag{2.37b}$$

along with

$w_{ij,k\ell,pq} \geq 0$ and is the same variable for each permutation of

$ij, k\ell, pq, \quad \forall$ distinct i, k, p and j, ℓ, q. (2.37c)

Such relaxations have been computationally tested by Ramakrishnan *et al.* (1996), and Ramachandran and Pekny (1996) with promising results. Extensions of such specializations for general set partitioning problems are given by Sherali and Lee (1996), and are further discussed in Chapter 5.

3

GENERALIZED HIERARCHY FOR EXPLOITING SPECIAL STRUCTURES IN MIXED-INTEGER ZERO-ONE PROBLEMS

In the previous chapter, we discussed a technique for generating a hierarchy of relaxations that span the spectrum from the continuous LP relaxation to the convex hull of feasible solutions for linear mixed-integer 0-1 programming problems. The key construct was to compose a set of multiplication factors based on the bounding constraints $0 \le x \le e_n$ on the binary variables x, and to use these factors to generate implied nonlinear product constraints, then tighten these constraints using the fact that $x_j^2 \equiv x_j \;\; \forall\, j = 1, \ldots, n$, and subsequently linearize the resulting polynomial problem through a variable substitution process. This process yielded tighter representations of the problem in higher dimensional spaces.

It should seem intuitive that if one were to identify a set S of constraints involving the x-variables that imply the bounding restrictions $0 \le x \le e_n$, then it might be possible to generate a similar hierarchy by applying a set of multiplicative factors that are composed using the constraints defining S. Indeed, as we exemplify in this chapter, such generalized so-called *S-factors* can not only be used to construct a hierarchy of relaxations leading to the convex hull representation, but this approach also provides an opportunity to exploit inherent special structures. Through an artful application of this strategy, one can design relaxations that are more compact than the ones available through the RLT process of Chapter 2, while at the same time, being tighter as well as affording the opportunity to construct the convex hull representation at lower levels in the hierarchy in certain cases. We illustrate this feature using various commonly occurring special structures such as generalized upper bounding (GUB) constraints, variable upper bounding (VUB) constraints, and to a lesser degree of structure, problem sparsity. These illustrations are by no means exhaustive; our motivation is to present the basic framework for this approach, and encourage the reader to design similar constructs for other applications on a case-by-case basis.

There is one other powerful trick that we introduce in this chapter, that is equally applicable to the RLT approach of the present as well as the previous chapter. This trick deals with the use of conditional logic in the generation of RLT based constraints that can enable the tightening of a relaxation at any level, which would otherwise have been possible only at some higher level in the hierarchy. In effect, this captures within the RLT process the concepts of branching and logical preprocessing, features that are critical to the efficient solution of discrete optimization problems.

We begin our discussion in Section 3.1 by presenting the enhanced RLT scheme designed for exploiting any inherent special structures, and we establish its theoretical properties in

Section 3.2. Various illustrations and examples using specific special structures are described in Section 3.3. Finally, Section 3.4 discusses the aforementioned conditional logic based strategy for tightening relaxations, illustrates how the standard lifting process can be viewed in this framework, and demonstrates an application of this approach to derive a tighter representation for the traveling salesman problem using the Miller-Tucker-Zemlin subtour elimination constraints. Note that while we focus here on *linear* mixed-integer 0-1 problems for clarity, as in Section 2.5 of Chapter 2, this approach readily extends in a similar fashion to *polynomial* mixed-integer 0-1 problems.

3.1. Generalized RLT for Exploiting Special Structures (SSRLT)

Consider the feasible region of a mixed-integer 0-1 programming problem stated in the form

$$X = \{(x, y) \in R^n \times R^m : \; Ax + Dy \geq b, \; x \in S, x \text{ binary}, \; y \geq 0\} \tag{3.1a}$$

where

$$S \equiv \{x: \; g_i x - g_{0i} \geq 0 \; \text{ for } i = 1, \ldots, p\}. \tag{3.1b}$$

Here, the constraints defining the set S have been specially composed to generate useful, tight relaxations as revealed in the sequel. For now, in theory, all that we require of the set S is that it implies the bounds $0 \leq x \leq e_n$ on the x-variables, where, as in Chapter 2, e_n is a column vector of n-ones. Specifically, we assume that for each $t = 1, \ldots, n$,

$$\min\{x_t: \; x \in S\} = 0 \quad \text{and} \quad \max\{x_t: \; x \in S\} = 1. \tag{3.2}$$

Note that if $\min(x_t) > 0$, we can fix $x_t = 1$, and if $\max(x_t) < 1$, we can fix $x_t = 0$, and if both these conditions hold for any t, then the problem is infeasible. Therefore, without loss of generality we will assume that the equalities of (3.2) hold for all t. We further note that (3.2) ensures that $p \geq n + 1$.

Now, define the sets P and $\overline{P}$ as follows, where $\overline{P}$ duplicates each index in P n times:

$$P = \{1, \ldots, p\}, \text{ and } \overline{P} = \{n \text{ copies of } P\}. \tag{3.3}$$

The construction of the new hierarchy proceeds in a manner similar to that of Chapter 2. At any chosen level of relaxation $d \in \{1, \ldots, n\}$, we construct a higher dimensional relaxation X_d by considering the *S-factors of order d* defined as follows:

$$g(J) = \prod_{i \in J} (g_i x - g_{0i}) \quad \text{for each distinct } J \subseteq \overline{P}, |J| = d. \tag{3.4}$$

Note that to compose the *S-factors* of order d, we examine the collection of *constraint-factors* $g_i x - g_{0i} \geq 0$, $i = 1, \ldots, p$, and construct the product of some d of these constraint-factors, *including possible repetitions.* To permit such repetitions as d varies from 1 to n, we have defined $\overline{P}$ as in (3.3) for use in (3.4). As we shall see, when $d = n$, the relaxation described below will recover the convex hull representation, and so, we need not consider $d > n$. Using $(d - 1)$ suitable dummy indices to represent duplications, it can be easily verified that there are a total of $\binom{p + d - 1}{d} = (p + d - 1)!/ d!(p - 1)!$ distinct factors of this type at level d.

These factors are then used in a **Special-Structure Reformulation-Linearization Technique**, abbreviated **SSRLT**, as stated below, in order to generate the relaxation $\overline{X}_d$.

(a) **Reformulation Phase**. Multiply each inequality defining the feasible region (3.1a) (including the constraints defining S) by each S-factor $g(J)$ of order d, and apply the identity $x_j^2 = x_j$ for all $j \in \{1, \ldots, n\}$.

(b) **Linearization Phase.** Linearize the resulting polynomial program by using the substitution defined in (3.5) below. This produces the d^{th} level relaxation $\overline{X}_d$.

$$w_J = \prod_{i \in J} x_J \quad \forall\ J \subseteq N,\ v_{Jk} = y_k \prod_{j \in J} x_j \quad \forall\ J \subseteq N,\ \forall k, \tag{3.5}$$

where, as in Chapter 2, the continuous variables are given by y_k, $k = 1, ..., m$.

For conceptual purposes, as in Chapter 2, define the projection of $\overline{X}_d$ onto the space of the original variables as

$$\overline{X}_{Pd} = \{(x, y)\colon\ (x, y, w, v) \in \overline{X}_d\} \quad \forall\ d = 1, \ldots, n \tag{3.6}$$

Additionally, as before, we will denote $\overline{X}_{P0} = \overline{X}_0$ (for $d = 0$) as the ordinary linear programming relaxation.

The main result of this chapter is that similar to (2.7), we have,

$$\overline{X}_{P0} \equiv \overline{X}_0 \supseteq \overline{X}_{P1} \supseteq \overline{X}_{P2} \supseteq \cdots \supseteq \overline{X}_{Pn} = conv(X) \tag{3.7}$$

where $conv(\cdot)$ denotes the convex hull of feasible solutions.

For convenience in notation, let us henceforth denote by $\{\cdot\}_L$ the process of linearizing a polynomial expression $\{\cdot\}$ in x and y via the substitution defined in (3.5), following the use of the identity $x_j^2 = x_j \ \forall\ j = 1,\ldots,n$. Accordingly, let us make the following observations that can be readily verified algebraically. Consider any pair of polynomial expressions Ψ and Φ. Then, the following operation is valid:

$$\{\Psi\}_L + \{\Phi\}_L = \{\Psi + \Phi\}_L. \tag{3.8}$$

Moreover, whenever we multiply a *linearized* expression with a *polynomial factor*, we will recognize the nonlinear form of the corresponding polynomial terms in the linearized expression, and hence treat this product as being equivalent to multiplying the corresponding *polynomial* expression with this latter factor. More succinctly, we have that

$$[\{\Psi\}_L \cdot \{\Phi\}]_L \equiv [\{\Psi\} \cdot \{\Phi\}]_L, \tag{3.9}$$

where $(\cdot)$ denotes the usual (algebraic) product.

Before proceeding to analyze the fundamental properties of the relaxations produced by SSRLT, let us highlight some important comments that pertain to the application.

First, in an actual implementation, note that under the substitution $x_j^2 = x_j$ for all j, several terms defining the factors in (3.4) might be zeros, either by definition, or due to the restrictions of the set X in (3.1a). For example, if $S \equiv \{x\colon\ 0 \le x \le e_n\}$, when $d = 2$, one such factor of type (3.4) is $x_j(1 - x_j)$ for $j \in \{1,\ldots,n\}$, which is clearly null when x_j^2 is equated with x_j. Second, some of these factors might be implied in the sense that they can be reproduced as a nonnegative surrogate of other such factors that are generated in (3.4). For example, when $d = 3$ for $S \equiv \{x\colon \ 0 \le x \le e_n\}$, the factor $x_t^2 x_r \ge 0$ of order 3 is equivalent to $x_t x_r \ge 0$ of order 2, which is implied by other nonnegative factors of order 3 generated by the RLT constraints, as proven in Lemma 2.1. All such null and implied factors and terms should be eliminated in an actual application of SSRLT. Third, if any constraint in $\overline{X}_0$ of (3.1a) is implied by the remaining constraints, then by (3.8) and (3.9) we can simply eliminate this constraint from X without changing any resulting set $\overline{X}_d$. (This same logic holds relative to the RLT of Chapter 2 and the sets X_d.) To illustrate, any single constraint in (2.35b) or (2.35c) can be removed from the QAP formulation of Example 2.4 while preserving the strengths of the prescribed relaxations at levels 1 and 2, saving $n(n-1)$ and $n(n-1)^2(n-2)$ constraints, respectively.

As evident from the foregoing comments, the RLT process described in Chapter 2 is a special case of SSRLT when $S = \{x\colon \ 0 \le x \le e_n\}$. It is precisely for the first two reasons given above that we do not define the set $\overline{P}$ as in (3.3) for this case, and include possible repetitions of the inequalities defining S in the construction of the sets X_d by the RLT in Chapter 2. Since RLT is a special case of SSRLT, we note here that the forthcoming arguments can be directly applied to the former. Note that one obvious scheme for generating tighter relaxations via SSRLT is to include in the latter set S certain suitable additional constraints depending on the problem structure, and hence generate S-factors at level d that include $F_d(J_1, J_2)$ of order d as defined in Chapter 2, along with any collection of additional S-factors as obtained via (3.4). Eliminating any null terms or implied factors thus generated, a hierarchy can be generated using SSRLT that would dominate the ordinary RLT hierarchy of relaxations at each level. Our focus here will largely reside on less obvious, and richer, instances where the set S possesses a special structure that implies the restrictions $0 \leq x \leq e_n$, without explicitly containing these bounding constraints.

Observe that we could conceptually think of SSRLT as being an inductive process, with the relaxation at level $(d + 1)$ being produced by multiplying each of the constraints in the relaxation at level d with each constraint defining S. Constraints produced by this process that effectively use null (zero) factor expressions $g(J)$ of order d are null constraints. Constraints produced by this process that effectively use factors $g(J)$ that are implied by other factors in (3.4), are by virtue of (3.8) and (3.9), themselves implied by the constraints generated using the latter factors. Hence, the process of reducing the set of factors based on eliminating null or implied factors from use at the reformulation step, or that of eliminating the corresponding redundant constraints generated by such factors, are equivalent steps. For convenience in analysis, we assume that a full relaxation using all possible factors of order d is generated at level d. By the foregoing comments, it follows that an equivalent relaxation at level d would be produced by using only the non-null, non-implied factors, recognizing any zero variable terms in the resulting relaxation as identified by the S-factors. Furthermore, such non-redundant/non-null factors can be generated inductively through the levels, recognizing zero terms revealed at previous levels. This latter relaxation is what should actually be generated in practice. Sections 3.3 and 3.4 provide several examples.

Note that we can also construct a hierarchy leading to the convex hull representation in a piecewise fashion as follows. Consider a partition of $N = \{1, 2, \ldots, n\}$ into disjoint sets $N_1, \ldots, N_r$ such that $\bigcup_{i=1}^{r} N_i = N$. Suppose that the foregoing scheme SSRLT is applied using a suitable set S by treating x_j, $j \in N_1$, as being binary valued, and the remaining x and y variables as being continuous, to produce the relaxation $\overline{X}_{|N_1|}$. Evidently, this constructs the set

$$Z_1 = conv\{\overline{X}_0 \cap \{(x, y): \quad x_j \text{ is binary for } j \in N_1\}\},$$

where $\overline{X}_0$ is the usual linear programming relaxation of the feasible region X. Note that Z_1 is the projection of the highest level relaxation $\overline{X}_{|N_1|}$ onto the space of the original variables. Therefore, every vertex of Z_1 has binary values for the variables x_j, $j \in N_1$. This process can now be repeated by treating Z_1 as the set $\overline{X}_0$, the variables in N_2 as being binary valued, and the remaining variables as being continuous. Upon the projection of this corresponding highest level relaxation, we would obtain

$$Z_2 = conv\{Z_1 \cap \{(x, y): \quad x_j \text{ is binary for } j \in N_2\}\}.$$

Note that Z_2 is the convex hull of vertices of Z_1 at which x_j, $j \in N_2$, are also binary valued. Hence, Z_2 is the convex hull of vertices of $\overline{X}_0$ at which x_j, $j \in N_1 \cup N_2$, are binary valued. Continuing in this manner, we can produce $conv(X)$. Observe the equivalence between the set Z, above and the discussion in Remark 2.2.

It is important to point out that the intermediate projection steps performed at each stage of the foregoing iterative process are not necessary to achieve a convex hull representation after r steps, or for that matter, the partial intermediate convex hull representations. After constructing factors based on the set N_i, and deriving the corresponding highest level relaxation, we can simply move on to the next set $N_{(i+1)}$ and apply the RLT process to this higher dimensional representation itself, using (3.8) and (3.9), and eliminating null/redundant constraints as mentioned above. In this manner, because of (3.2), as will be evident from the proof of Theorem 3.2, at the end of the first stage, the relaxation $\overline{X}_{|N_1|}$ will have effectively multiplied all the original constraints using factors $F_{|N_1|}(J_1, J_2)$ of order $|N_1|$ composed of the terms x_j and $(1 - x_j)$ for $j \in N_1$. At the end of the second stage, under (3.8) and (3.9), these resulting constraints will have been effectively multiplied by all factors $F_{|N_2|}(J_1, J_2)$ of order $|N_2|$ composed of the terms x_j and $(1 - x_j)$ for $j \in N_2$. This is equivalent to having multiplied the original constraints using all factors $F_{|N_1 \cup N_2|}(J_1, J_2)$ of order $|N_1 \cup N_2|$ composed of the terms x_j and $(1 - x_j)$ for $j \in N_1 \cup N_2$. Continuing this process, from Chapter 2, the projection of the final relaxation at step r onto the space of the original variables (x, y) would then represent $conv(X)$. Another insightful viewpoint of this process is that the intermediate projection operation of a higher dimensional relaxation simply constructs all possible implied surrogates of the constraints in the higher dimension that eliminate the new cross-product variables. Hence, any constraint obtained by multiplying a projected (surrogate) constraint using some factor, can be reproduced by constructing the same surrogate after multiplying the corresponding parent constraints using this same factor. Moreover, in order to computationally use the relaxations, final or intermediate, it is not advisable, or for that matter necessary, to perform the complete projection operation. The higher dimensional representations can themselves be used as linear programming relaxations.

Finally, let us comment on the treatment of equality constraints and equality factors. Note that whenever an equality constraint-factor defines an S-factor, any resulting product constraint is an equality restriction. Consequently, in the presence of equality constrained factors, in general, it is only necessary to multiply the corresponding equality constraint factors simply with each x and y variable alone, as well as by the constant 1, since the product with any other expression in x and y can be composed using these resulting products. Moreover, since the products with the x-variables are already being generated via other SSRLT constraints by virtue of the corresponding defining equality constraints of S already being included with X and since $x \geq 0$ is implied by the inequality restrictions of $x \in S$, only products using y variables are necessary. This product generation should become evident in Section 3.2.

Furthermore, in this connection, note that if X contains equality structural constraints, in general, then these can be treated as in Section 2.5. That is, at level d, these equality constraints would simply need to be multiplied by the factors $F_p(J, \emptyset)$ for $J \subseteq N$,

$p = |J| = 0, 1, \ldots, d$. Naturally, factors $F_p(J, \varnothing)$ that are known to be zeros, *i.e.*, any such factor for which we know that no feasible solution exists that has $x_j = 1 \ \forall \ j \in J$, need not be used in constructing these product constraints, and can be set equal to zero in the relaxation.

For example, suppose that we have a set $S = \{x\colon \ e_n \cdot x = 1, x \geq 0\}$. Then the S-factors of order 1 are the expressions that define the restrictions

$$\{(1 - e_n \cdot x) = 0, \ x_1 \geq 0, \ldots, x_{.} > 0\}, \tag{3.10}$$

which include the equality constraint-factor along with the bound-factors $x_1 \geq 0, \ldots, x_n \geq 0$. To compose the S-factors of order 2, note that

$$x_t(1 - e_n \cdot x) = 0 \text{ yields } \sum_{j \neq t} w_{(jt)} = 0 \quad \forall \ t, \tag{3.11}$$

upon using $x_t^2 \equiv x_t$ and substituting $w_{(jt)}$ for $x_j x_t \quad \forall \ j \neq t$ according to (3.5). (Note that we only need to define w_{jt} for $j < t$, and accordingly, we will denote $w_{(jt)} \equiv w_{jt}$ if $j < t$ and $w_{(jt)} \equiv w_{tj}$ if $t < j$.) Equation (3.11) along with

$$w_{(jt)} \equiv x_j x_t \geq 0 \quad \forall \ j \neq t \tag{3.12}$$

produced by the other S-factors of order 2 imply that $w_{(jt)} \equiv 0 \quad \forall \ j \neq t$, hence yielding null factors via (3.11) and (3.12). The only non-null S-factors of order 2 are therefore produced by pairwise self-products of the constraints defining S. But $(1 - e_n \cdot x)^2 = 0$ and $x_j^2 \geq 0$, $j = 1, \ldots, n$ respectively yield $(1 - e_n \cdot x) = 0$, and $x_j \geq 0 \ \forall \ j = 1, \ldots, n$, upon using $x_j^2 \equiv x_j$ and $x_j x_t = 0 \ \forall \ j \neq t$ as above.

Hence, the reduced set of factors of order 2 are precisely the same as those of order 1, and this continues for all levels $2, \ldots, n$. Consequently, by (3.7), the convex hull representation would necessarily be produced at level 1 itself for this example.

To produce this level 1 representation, we would multiply all the constraints defining X (including the ones in S) by each factor $x_j \geq 0$, $j = 1, \ldots, n$ from (3.10). However, for the equality factor $(1 - e_n \cdot x) = 0$, by the foregoing discussion, we would only need to construct the RLT constraints $\{y_k(1 - e_n \cdot x)\}_L = 0$ and retain $e_n \cdot x = 1$. The resulting relaxation would produce the convex hull representation as asserted above.

To further reinforce some of the preceding ideas before presenting additional specific details, we use another example that includes an equality constraint in S, but also explicitly includes the bound restrictions $0 \leq x \leq e_n$. As mentioned above, since the S-factors would now include the regular RLT bound-factors, any S-factors other than these bound-factor products are optional. To present the example, using $n = 4$ for illustration, suppose that $S = \{x \in R^4 : x_1 + x_2 + x_3 + x_4 = 2,\ 0 \leq x \leq e_4\}$. The following factors are derived that can be applied in SSRLT, noting the equality constraint defining S.

(a) *Level 1 factors:* $x_j \geq 0$ and $(1 - x_j) \geq 0$, $j = 1, \ldots, 4$, and optionally, $(e_4 \cdot x - 2) = 0$ (to be multiplied by 1 and by each y variable alone as noted above).

(b) *Level 2 factors:* Bound factors of order 2 given by $\{x_i x_j, (1 - x_i)x_j, x_i(1 - x_j)$, and $(1 - x_i)(1 - x_j)\ \forall\ 1 \leq i < j \leq 4\}$, and optionally, any factors (to be applied to

$y \geq 0$ alone) from the set $\{x_i - \sum_{j \neq i} x_i x_j = 0 \; \forall \; i = 1, \ldots, 4$ obtained by multiplying $e_4 \cdot x = 2$ by each x_i, $i = 1, \ldots, 4$, and $(e_4 \cdot x - 2) = 0$ itself, obtained from $(e_4 \cdot x - 2)^2 = 0$ upon using $\sum_{j \neq i} x_i x_j = x_i \; \forall \; i\}$.

(c) *Level d factors,* $d = 3, 4$: Bound factors $F_d(J_1, J_2) \geq 0$ of order d, with the additional restriction that all 3^{rd} and 4^{th} order terms are zeros, plus optionally, factors from the optional set at level 2. Note that the valid implication of polynomial terms of order 3 being zero, for example, is obtained through the RLT process by multiplying $x_i - \sum_{j \neq i} x_i x_j = 0$ with x_k, for each $i, k, i \neq k$. This gives $\sum_{j \neq i, k} x_i x_j x_k = 0$ which, by the nonnegativity of each triple product term, implies that $x_i x_j x_k = 0$ $\forall \; i \neq j \neq k$.

In a likewise fashion, for set partitioning problems, for example, any quadratic or higher order products of variables that involve a pair of variables that appear together in any constraint are zeros. More generally, any product term that contains variables or their complements that cannot simultaneously take on a value of 1 in any feasible solution can be restricted to zero. Sherali and Lee (1992) use this structure to present a specialization of RLT to derive explicit reduced level d representations in their analysis of set partitioning problems. This is discussed in Chapter 5. But now we are ready to establish (3.7) and then consider other specific special structures that are amenable to SSRLT.

3.2. Validation of the Hierarchy for SSRLT

In this section, we shall establish (3.7). The essence of the proof relies on the development in Chapter 2 along with the fact that the constraints of S imply the bound-factor restrictions $0 \leq x \leq e_n$. In particular, consider (3.2). For each $t = 1, \ldots, n$, denoting λ_i^t, $i = 1, \ldots, p$, as the set of optimal dual multipliers for the first (min) problem in (3.2), we have by dual feasibility that $(\sum_{i=1}^{p} \lambda_i^t g_i)x = x_t$, while the optimal objective function value $\sum_{i=1}^{p} \lambda_i^t g_{0i} = 0$. This gives,

$$\sum_{i=1}^{p} \lambda_i^t (g_i x - g_{0i}) = x_t, \text{ where } (\lambda_i^t, i = 1, \ldots, p) \geq 0, \ \forall\, t = 1, \ldots, n. \tag{3.13}$$

Similarly, for each $t = 1, \ldots, n$, denoting π_i^t, $i = 1, \ldots, p$, as the negative of the set of optimal dual multipliers for the second (max) problem in (3.2), we get,

$$\sum_{i=1}^{p} \pi_i^t (g_i x - g_{0i}) = (1 - x_t), \text{ where } (\pi_i^t, i = 1, \ldots, p) \geq 0, \ \forall\, t = 1, \ldots, n. \tag{3.14}$$

Now, consider the following results.

Theorem 3.1. (Hierarchy of Relaxations). $\overline{X}_{Pd} \subseteq \overline{X}_{P(d-1)} \ \forall\, d = 1, \ldots, n.$

Proof. It is sufficient to show that for any $d \in \{1, \ldots, n\}$, we can obtain any constraint $\alpha z - \beta \geq 0$ of $\overline{X}_{(d-1)}$, where $z \equiv (x, y, w, v)$, by surrogating appropriate constraints of $\overline{X}_d$ using nonnegative multipliers. Given such an inequality in $\overline{X}_{(d-1)}$, by the inductive process of applying SSRLT, we have inherent in $\overline{X}_d$, or implied by the constraints of $\overline{X}_d$, the set of inequalities

$$\{(\alpha z - \beta) \cdot (g_i x - g_{0i})\}_L \geq 0 \quad \forall\ i = 1, \ldots, p, \tag{3.15}$$

where the operation in (3.15) is defined in (3.9). Using (3.8), (3.9), (3.13), and (3.14), we have for any $t \in \{1, \ldots, n\}$,

$$\sum_{i=1}^{p} (\lambda_i^t + \pi_i^t)[(\alpha z - \beta) \cdot (g_i x - g_{0i})]_L = [(\alpha z - \beta) \cdot \sum_{i=1}^{p} \lambda_i^t (g_i x - g_{0i})]_L$$

$$+[(\alpha z - \beta) \cdot \sum_{i=1}^{p} \pi_i^t (g_i x - g_{0i})]_L = [(\alpha z - \beta) x_t]_L + [(\alpha z - \beta)(1 - x_t)]_L$$

$$= (\alpha z - \beta).$$

Hence, (3.15) implies that $(\alpha z - \beta) \geq 0$, and this completes the proof. □

Theorem 3.2. (Dominance of SSRLT over RLT). *$\overline{X}_d \subseteq X_d$ $\forall$ $d = 0, 1, \ldots, n$, and so, $\overline{X}_{Pd} \subseteq X_{Pd}$ $\forall$ $d = 0, 1, \ldots, n$, where the relaxation X_d and its projection X_{Pd} are given by (2.5) and (2.6), respectively.*

Proof. It is sufficient to show that for each level $d = 0, 1, \ldots, n$, any defining constraint of X_d can be obtained as a nonnegative surrogate of constraints of $\overline{X}_d$. This is clearly true for $d = 0$, since $\overline{X}_0 = X_0$ defines the usual linear programming relaxation. Now, by induction, assume that this assertion is true for $(d-1)$, and consider the corresponding d^{th} level relaxations X_d and $\overline{X}_d$, where $d \in \{1, \ldots, n\}$. By the inductive viewpoint of applying RLT, each constraint of X_d is obtained by computing the product of a constraint of $X_{(d-1)}$ with either x_t or $(1 - x_t)$, for some t. Consider any such constraint of X_d which, by (3.9), can be written as

$$[(\alpha z - \beta) \cdot x_t]_L \geq 0, \tag{3.16}$$

where $z \equiv (x, y, w, v)$, and where $\alpha z \geq \beta$ is a constraint defining $X_{(d-1)}$. (The case of $(1 - x_t)$ being applied in (3.16) is similar.) By the induction hypothesis, there exist defining constraints of $\overline{X}_{(d-1)}$ of the type $\gamma_k z - \mu_k \geq 0$, for $k = 1, \ldots, K$, and multipliers $\phi_k \geq 0$, for $k = 1, \ldots, K$, such that

$$\sum_{k=1}^{K} \phi_k [\gamma_k z - \mu_k] = (\alpha z - \beta). \tag{3.17}$$

Moreover, by the inductive viewpoint of applying SSRLT, there exist constraints of $\overline{X}_d$ of the form

$$[(\gamma_k z - \mu_k) \cdot (g_i x - g_{0i})]_L \geq 0 \quad \forall\, i, k. \tag{3.18}$$

Surrogating (3.18) via the following multipliers, where $\lambda_i^t\ \forall\, i$ are defined in (3.13), and applying (3.8), (3.9) and (3.17), yields

$$0 \leq \sum_{k=1}^{K} \phi_k \left\{ \sum_{i=1}^{p} \lambda_i^t [(\gamma_k z - \mu_k) \cdot (g_i x - g_{0i})]_L \right\}$$

$$= \sum_{k=1}^{K} \phi_k [x_t (\gamma_k z - \mu_k)]_L$$

$$= [(\alpha z \geq \beta) x_t]_L.$$

Hence, a surrogate of the constraints (3.18) of $\overline{X}_d$ using nonnegative multipliers, produces the constraint (3.16) of X_d. This completes the proof. □

Corollary 3.1. $\overline{X}_{P0} \equiv \overline{X}_0 \supseteq \overline{X}_{P1} \supseteq \overline{X}_{P2} \supseteq \cdots \supseteq \overline{X}_{Pn} = conv(X)$.

Proof. The inclusions $\bar{X}_{P0} \equiv \bar{X}_0 \supseteq \bar{X}_{P1} \supseteq \bar{X}_{P2} \supseteq \cdots \supseteq \bar{X}_{Pn}$ are proven in Theorem 3.1 above. Furthermore, since $\bar{X}_{Pn}$ is a valid relaxation, we have $conv(X) \subseteq \bar{X}_{Pn}$. Moreover, we have by Theorems 3.2 and 2.2 that $\bar{X}_{Pn} \subseteq X_{Pn} = conv(X)$. Hence, $\bar{X}_{Pn} = conv(X)$, and this completes the proof. □

3.3. Composing S and S-Factors for Some Special Structures

We now demonstrate how some specific special structures can be exploited in designing an application of the general framework of SSRLT. This discussion will also illuminate the relationship between RLT and SSRLT, beyond the simple dominance result stated in Theorem 3.2 above.

3.3.1. Generalized Upper Bounding (GUB) or Multiple Choice Constraints

Suppose that the set S of (3.1b) is given as follows,

$$S = \{x: \sum_{j \in N_i} x_j \le 1 \quad \forall\ i \in Q = \{1, \ldots, q\},\ x \ge 0\}, \tag{3.19}$$

where $\bigcup_{i \in Q} N_i \equiv N \equiv \{1, \ldots, n\}$. Problems possessing this particular special structure arise in various settings including maximum cardinality node packing, set packing, capital budgeting, and menu planning problems among others (see Nemhauser and Wolsey, 1988).

First, let us suppose that $q \equiv 1$ in (3.19), so that

$$S \equiv \{x: e_n \cdot x \le 1,\ x \ge 0\}.$$

The S-factors of various orders for this particular set can be derived as follows:

(a) *S-factors at level 1.* These factors are directly obtained from the set S via the constraint factors $(1 - e_n \cdot x) \geq 0$, and $x_j \geq 0 \ \forall \ j = 1,\ldots,n$.

(b) *S-factors at level 2*. The linearization operation $[x_t(1 - e_n \cdot x)]_L \geq 0$ produces an expression

$$\sum_{j \neq t} w_{(jt)} \leq 0 \qquad \forall \ t = 1,\ldots,n$$

where $w_{(jt)} \equiv w_{tj}$ if $t < j$, and $w_{(jt)} \equiv w_{jt}$ if $j < t$. Moreover, via the pairwise products of x_j and x_t, $j \neq t$, we obtain factors of the type

$(x_t x_j) \geq 0 \ \forall \ j \neq t$, or $w_{(jt)} \geq 0 \ \forall \ j \neq t$.

Similar to (3.11) and (3.12), the foregoing two sets of inequalities imply that

$$w_{(jt)} = 0 \ \forall \ j \neq t. \tag{3.20}$$

Consequently, under (3.20), the only S-factors of order 2 that survive such a cancellation are self-product factors of the type $(x_t x_t) \geq 0 \ \forall \ t$, and $(1 - e_n \cdot x) \cdot (1 - e_n \cdot x) \geq 0$. These yield the same factors as at level 1, upon using $x_t^2 = x_t \ \forall \ t$ along with (3.20) as seen in Section 3.1. Hence, we only need to use the factors

$$(1 - e_n \cdot x) \geq 0 \ \text{ and } \ x_j \geq 0, \ \ j = 1,\ldots,n$$

to construct the equivalent set $\overline{X}_2$. Notice that $\overline{X}_2 \equiv \overline{X}_1$, and this equivalence relation continues through all levels of relaxations up to $\overline{X}_n$. Hence, the first level relaxation itself produces the convex hull representation in this case.

There are two insightful points worthy of note in the context of this example. First, although RLT recognizes that (3.20) holds true at each relaxation level, it may not produce the convex hull representation at the first level as does SSRLT. For example, let

$$X = \{(x_1, x_2): \; 6x_1 + 3x_2 \geq 2, \; x_1 + x_2 \leq 1, \; x \text{ binary}\},$$

and consider the generation of the first level RLT relaxation. Note that the factors used in this context are x_j and $(1 - x_j)$ for $j = 1, 2$. Examining the product of $x_1 + x_2 \leq 1$ with x_1 yields $w_{12} \leq 0$, which together with $w_{12} \equiv [x_1 x_2]_L \geq 0$ yields $w_{12} \equiv 0$. Other products of the factors x_j and $(1 - x_j)$, $j = 1, 2$, with $x_1 + x_2 \leq 1$ and $0 \leq x \leq e_2$ simply reproduce these same latter constraints.

Examining the products of $6x_1 + 3x_2 \geq 2$ with these first level factors yields nonredundant inequalities when the factors $(1 - x_1)$ and $(1 - x_2)$ are used, generating the constraints $2x_1 + 3x_2 \geq 2$ and $3x_1 + x_2 \geq 1$, respectively. Hence, we obtain the first level relaxation (directly in projected form in this case) as

$$X_{P1} = \{(x_1, x_2): \; 2x_1 + 3x_2 \geq 2, \; 3x_1 + x_2 \geq 1, \; x_1 + x_2 \leq 1, \; x \geq 0\}.$$

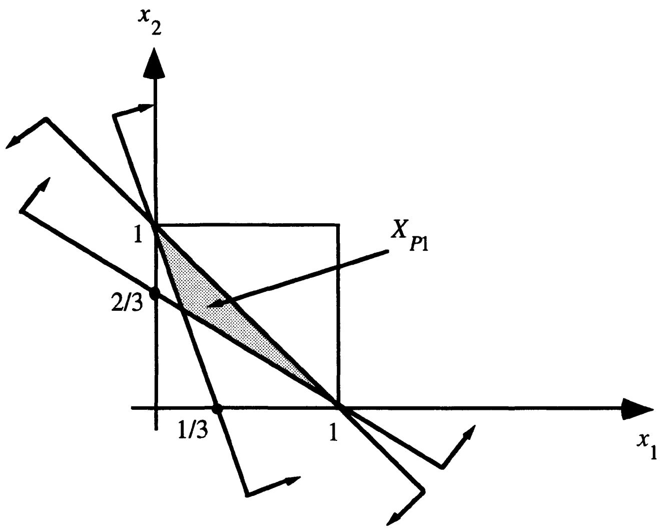


Figure 3.1. The first level relaxation using RLT.

Figure 3.1 depicts the region X_{P1}. However, using SSRLT, by the above argument, we would obtain $\overline{X}_{P1} \equiv conv(X) \equiv \{x : x_1 + x_2 = 1, x \geq 0\}$ which is a strict subset of X_{P1}.

A second point to note is that we could have written the generalized upper bounding inequality $e_n \cdot x \leq 1$ as an equality $e_n \cdot x + x_{n+1} = 1$ by introducing a slack variable x_{n+1}, and then *recognizing the binariness of this slack variable*, we could have used this additional variable in composing the bound-factor products while applying RLT. Although this would have produced the same relaxation as with RLT, the process would have generated several more redundant constraints while applying the factors x_j and $(1 - x_j)$ for $j = 1, \ldots, n + 1$, to the constraints, as opposed to using the fewer factors $(1 - e_n \cdot x)$ and x_j, $j = 1, \ldots, n$, as needed by SSRLT. However, in more general cases of the set S, such a transformation that yields the same representation using RLT as obtained via SSRLT may not be accessible. (See the numerical example of Section 3.4 below, for instance.)

Next, let us consider the case of $q = 2$ in (3.19) for the sake of illustration. Suppose that

$$S = \{x \in R^5: \ x_1 + x_2 + x_3 \le 1, \ x_3 + x_4 + x_5 \le 1, \text{ and } x \ge 0\}. \tag{3.21}$$

(a) *S-factors at level 1:* These are simply the constraints defining S.

(b) *S-factors at level 2:* As before, the pairwise products within each GUB set of variables will reproduce the same factors as at the first level, since (3.20) holds true within each GUB set. However, across the two GUB sets, we would produce the nonnegative quadratic bound factor products x_1x_4, x_1x_5, x_2x_4, and x_2x_5, along with the following factor products, recognizing that any quadratic product involving x_3 is zero, as this variable appears in both GUB sets.

$x_4 \cdot (1 - x_1 - x_2 - x_3) \ge 0$ yielding $x_4 - x_1x_4 - x_2x_4 \ge 0,$

$x_5 \cdot (1 - x_1 - x_2 - x_3) \ge 0$ yielding $x_5 - x_1x_5 - x_2x_5 \ge 0,$

$x_1 \cdot (1 - x_3 - x_4 - x_5) \ge 0$ yielding $x_1 - x_1x_4 - x_1x_5 \ge 0,$

$x_2 \cdot (1 - x_3 - x_4 - x_5) \ge 0$ yielding $x_2 - x_2x_4 - x_2x_5 \ge 0,$

and

$(1 - x_1 - x_2 - x_3) \cdot (1 - x_3 - x_4 - x_5) \ge 0$ yielding

$$1 - x_1 - x_2 - x_3 - x_4 - x_5 + x_1x_4 + x_1x_5 + x_2x_4 + x_2x_5 \geq 0.$$

These can now be applied to the constraints defining X, recognizing the terms that have been identified to be zeros.

(c) *S-factors at levels* ≥ 3: Since there are only 2 GUB sets in this example, and since any triple product of distinct factors must involve a pair of factors coming from the defining constraints corresponding to the same GUB set, and the latter product is zero, all such products must vanish. Hence, all factors at level 3, and similarly at levels 4 and 5, coincide with those at level 2. In other words, the relaxation at level 2 (defined as $\overline{X}_2$) itself yields the convex hull representation.

In general, *the level equal to the independence number of the underlying intersection graph corresponding to the GUB constraints, which simply equals the maximum number of variables that can simultaneously be 1, is sufficient to generate the convex hull representation.* In the case of (3.19), the convex hull representation would be obtained at level q, or earlier.

An enlightening special case of (3.19) deals with the vertex packing problem. Given a graph $G = (V, E)$ with vertex set $V = \{v_1, v_2, \ldots, v_n\}$, an edge set E connecting pairs of vertices in V, and a weight c_j associated with each vertex v_j, the vertex packing problem is to select a maximum weighted subset of vertices such that no two vertices are connected by an edge. For each $j = 1, \ldots, n$ by denoting the binary variable x_j to equal 1 if vertex j is chosen and 0 otherwise, the vertex

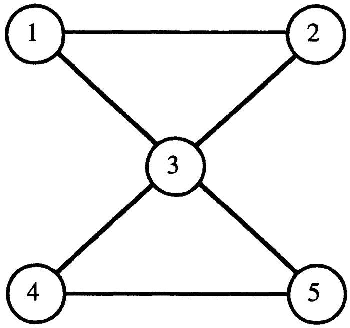


Figure 3.2. Vertex packing graph.

packing problem can be stated as maximize $\{\sum_j c_j x_j : x_i + x_j \le 1 \ \forall\ (i,j) \in E,\ x$ binary$\}$. The convex hull representation over any subset P of the variables can be obtained as above, by considering any clique cover of the subgraph induced by the corresponding vertices, with each set N_i corresponding to the variables defining some clique i. In fact, given a cover that has q cliques where each edge of E is included in some clique graph, the S-factors of level q themselves define the convex hull representation, since their products with the packing constraints, as well as with the nonnegativity restriction on x, are implied by these factors. To illustrate, the inequalities of (3.21) can be considered as a (maximum cardinality) clique cover of the vertex packing problem on the graph in Figure 3.2, and so, the stated S-factors of level 2 themselves define the convex hull representation. This general packing observation may have widespread applicability since, as noted by Garfinkel and Nemhauser (1973), any finite integer linear program can be reformulated as a packing problem.

Table 3.1. Computational results for set packing problems.

Ordinary LP relaxation				Level-one via RLT			Level-one via SSRLT		
(q, n)	DENSITY	% gap	ITER	(m', n')	% gap	ITER	(m', n')	% gap	ITER
(15,25)	66	42.0	22	(170,25)	16.8	160	(123,25)	0	35
(20,30)	56	26.7	36	(319,30)	7.2	267	(214,30)	0	32
(25,35)	52	57.3	43	(481,35)	20.2	152	(328,35)	0	113
(55,45)	27	37.0	103	(2256,63)	2.0	344	(1896,63)	0	171
(35,35)	56	36.5	48	(607,35)	10.7	292	(631,35)	0	60

To illustrate the computational benefits of SSRLT over RLT in this particular context, we conducted the following experiment using pseudo-randomly generated set packing problems of the type

$$\text{maximize } \{\textstyle\sum_{j=1}^{n} c_j x_j : \; \sum_{j\in N_i} x_j \le 1 \quad \forall\, i = 1,\dots,q, x \text{ binary}\}.$$

For several instances of such problems, we computed the optimal value of the 0-1 packing problem, that of its ordinary LP relaxation, as well as the optimal values of the first level relaxations produced by applying RLT and SSRLT, where the latter was generated by using all of the defining clique constraints, together with $x \ge 0$, to represent the set S. Table 3.1 gives the percentage gaps obtained for the latter three upper bounds with respect to the optimal 0-1 value, along with the sizes of the respective relaxations (m' = number of constraints, n' = number of variables), and the number of simplex iterations (ITER) needed by the OSL solver version 2.001, to achieve optimality. Note that for all instances, the first level application of SSRLT *was sufficient to solve the underlying integer program.* On the other hand, although RLT appreciably improved the upper bound produced by the ordinary LP relaxation, it still left a significant gap that remains to be resolved in these problem instances. Moreover, the relatively simpler structure of SSRLT results in far fewer simplex iterations being required to solve this relaxation as compared with the effort required to solve RLT.

In concluding the discussion for this case, for the sake of insights and illustration, we indicate another alternative approach for constructing relaxations leading to the convex hull representation for (3.19). Arbitrarily, consider the variables of one of the GUB constraints, say x_j, $j \in N_1$, to be binary, treat the remaining variables as being continuous, and as for the case $q = 1$, identify the appropriate first order S-factors associated with this GUB constraint. Applying these factors using SSRLT at the first level, produces the convex hull of feasible solutions for which x_j, $j \in N_1$, are restricted to be binary valued. Denote this set as $conv(X(N_1))$. Next, repeat the process using N_2, considering the variables x_j, $j \in N_2$, to be binary, along with those of N_1. Applying the factors $x_j \geq 0$, $j \in N_2$, and $(1 - \sum_{j \in N_2} x_j) \geq 0$ to the foregoing *relaxation*, produces the set $conv(X(N_1 \cup N_2))$, with obvious notation. Continuing in this manner, upon a complete sequential application of all GUB constraint S-factors, we recover

$$conv\left(X\left(\bigcup_{i \in Q} N_i\right)\right) = conv(X(N)).$$

3.3.2. Variable Upper Bounding Constraints

This example points out that in the presence of variable upper bounding (VUB) types of restrictions, a further tightening of relaxations via SSRLT, beyond that of RLT, can be similarly produced. For example, a set S might be composed as follows in a particular problem instance:

$$S = \{x \in R^6 : \ 0 \le x_1 \le x_2 \le x_3 \le 1, \ 0 \le x_4 \le x_5 \le 1, \ 0 \le x_6 \le 1\}.$$

The first level factors for this instance are given by $x_1 \ge 0$, $x_2 - x_1 \ge 0$, $x_3 - x_2 \ge 0$, $1 - x_3 \ge 0$, $x_4 \ge 0$, $x_5 - x_4 \ge 0$, $1 - x_5 \ge 0$, $x_6 \ge 0$, and $1 - x_6 \ge 0$. Compared with the RLT factors, these yield tighter constraints as they imply the RLT factors. For $d \in \{1, \ldots, 6\}$, taking these factors d at a time, including self-products, and simplifying these factors by eliminating null or implied factors, would produce the relaxation $\overline{X}_d$.

It is interesting to note in this connection that the VUB constraints of the type $0 \le x_1 \le x_2 \le \cdots \le x_k \le 1$ used in this example, can be equivalently transformed into a GUB constraint via the substitution $z_j = x_j - x_{j-1}$ for $j = 1, \ldots, k$, where $x_0 \equiv 0$. The inverse transformation yields $x_j = \sum_{t=1}^{j} z_t$ for $j = 1, \ldots, k$, thereby producing the equivalent representation $z_1 + z_2 + \cdots + z_k \le 1$, $z \ge 0$. Under this transformation, and imposing binary restrictions on all the z-variables, the reformulation strategies described in Section 3.3.1 above can be employed. However, note that the process of applying

RLT as in Chapter 2 to the original or to the transformed problem can produce different representations. To illustrate this insight, suppose that

$$X = \{(x_1, x_2): \ -6x_1 + 3x_2 \le 1, \ 0 \le x_1 \le x_2 \le 1, x \text{ binary}\}.$$

The convex hull of feasible solutions is given by $0 \le x_1 = x_2 \le 1$ (see Figure 3.3(a)). This representation is produced by the level-1 SSRLT relaxation using the VUB constraints to define S, where the relevant constraint $x_1 \ge x_2$ which yields $x_1 = x_2$ is obtained by noting that the factor products $[x_1(x_2 - x_1)]_L \ge 0$ and $[x_1(1 - x_2)]_L \ge 0$ respectively give $w_{12} \ge x_1$ and $w_{12} \le x_1$, or that $w_{12} = x_1$. This together with the constraint $[(x_2 - x_1)(1 + 6x_1 - 3x_2)]_L \ge 0$ yields $-2(x_2 - x_1) \ge 0$, or that $x_1 \ge x_2$.

On the other hand, constructing RLT as in Chapter 2 at level 1 by applying the factors x_j and $(1 - x_j)$, $j = 1, 2$, to the inequality restrictions of X, produces the relaxation (directly in projected form)

$$X_{P1} = \{(x_1, x_2): \ 2x_1 - 3x_2 \ge -1, \ 3x_1 \ge x_2, \ 0 \le x_1 \le x_2 \le 1\},$$

where $w_{12} = x_1$ is produced as with SSRLT, and where the first two constraints defining X_{P1} result from the product constraints $[(1 + 6x_1 - 3x_2)(1 - x_1)]_L \ge 0$ and $[(1 + 6x_1 - 3x_2)x_2]_L \ge 0$, respectively. Figure 3.3(a) depicts the region defined by this relaxation.

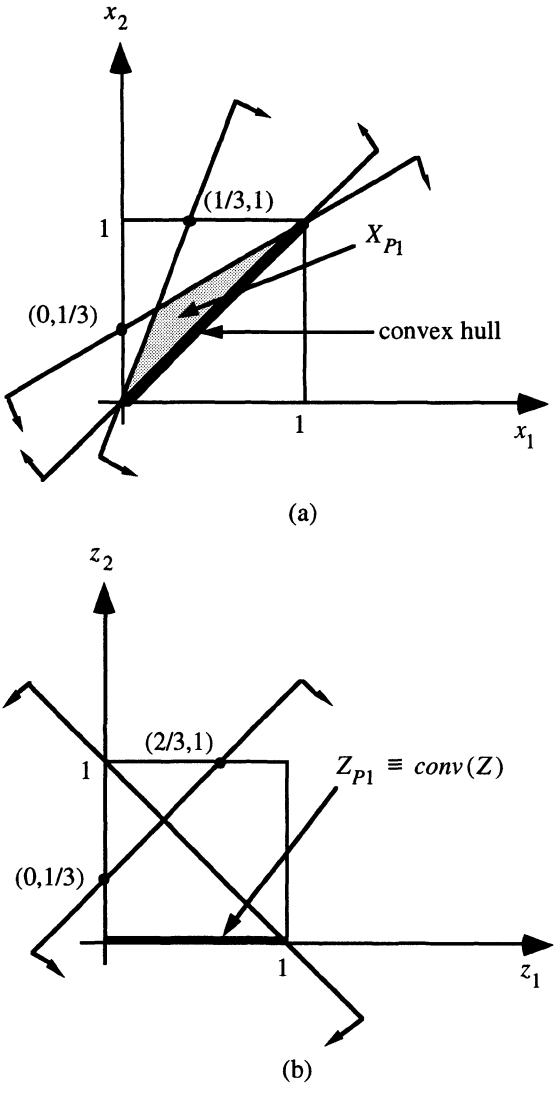


Figure 3.3. Depiction of the first level relaxations using RLT.

However, if we were to apply the transformation $z_1 = x_1$, $z_2 = x_2 - x_1$ to X, where the inverse transformation is given by $x_1 = z_1$ and $x_2 = z_1 + z_2$, the problem representation in z-space becomes

$$Z = \{(z_1, z_2): \ -3z_1 + 3z_2 \leq 1, \ z_1 + z_2 \leq 1, z \text{ binary}\},$$

where the *binariness on* z_2 *has been additionally recognized.* Figure 3.3(b) illustrates that the set $conv(Z)$ is given by the constraints $0 \leq z_1 \leq 1$, $z_2 = 0$. Now, applying RLT to this transformed space, the relevant constraint $z_2 = 0$ is produced via $z_2 \geq 0$ and the first level product constraint $[(1 + 3z_1 - 3z_2)z_2]_L \geq 0$ which yields $-2z_2 \geq 0$, where $[z_1z_2]_L \equiv 0$ from $[z_1z_2]_L \geq 0$ and $[z_1(1 - z_1 - z_2]_L \geq 0$. Hence, for this transformed problem, RLT produces the first level relaxation $Z_{P1} = conv(Z)$, while we had $X_{P1} \supset conv(X)$ when applying RLT to the original problem. However, as with the illustration in Section 3.3.1 for the case $q = 1$ (treating that as the transformed z-variable problem), we could have possibly obtained $Z_{P1} \supset conv(Z)$ as well, whereas SSRLT would necessarily produce the convex hull representation at level one in either case.

3.3.3. Sparse Constraints

In this example, we illustrate how one can exploit problem sparsity. Suppose that in some 0-1 mixed-integer problem, we have the knapsack constraint (either inherent in the problem or implied by it) given by $2x_1 + x_2 + 2x_3 \geq 3$. The facets of the convex hull of $\{(x_1, x_2, x_3): \ 2x_1 + x_2 + 2x_3 \geq 3, \ x_i \text{ binary}, \ i = 1, 2, 3\}$ can be readily obtained as $\{x_1 + x_2 + x_3 \geq 2,\ \ x_1 \leq 1,\ \ x_2 \leq 1,\ \ x_3 \leq 1\}$. Similarly, another knapsack constraint might be of the type $x_4 + 2x_5 + 2x_6 \leq 2$, and the corresponding facets of the convex hull of feasible 0-1 solutions can be obtained as $\{x_4 + x_5 + x_6 \leq 1,$ $x_4 \geq 0,\ x_5 \geq 0,\ x_6 \geq 0\}$. The set S can now be composed of these two sets of facets, along with other similar constraints involving the remaining variables on which binariness is being enforced, including perhaps, simple bound constraint factors. Note that in order to generate valid tighter relaxations, we can simply enforce binariness on variables that fractionate in the original linear programming relaxation in the present framework. Furthermore, entire convex hull representations of underlying knapsack polytopes are not necessary — simply, the condition (3.2) needs to be satisfied, perhaps by explicitly including simple bounding constraints. This extends the strategy of Crowder *et al.* (1983) in using facets obtained as liftings of minimal covers from knapsack constraints within this framework, in order to generate tighter relaxations.

3.4. Using Conditional Logic to Strengthen RLT/SSRLT Constraints

In all of the foregoing discussion, depending on the structure of the problem, there is another idea that we can exploit to even further tighten the RLT constraints that are generated. To introduce the basic concept involved, for simplicity, consider the following first level RLT constraint that is generated by multiplying a factor $(\alpha x - \beta) \geq 0$ with a constraint $(\gamma x - \delta) \geq 0$, where x is supposed to be binary valued, and where the data is all-integer. (Similar extensions can be developed for mixed-integer constraints, as well as for higher-order SSRLT constraints in which some factor is being applied to some other valid constraint-or-bound-factor product of order greater than one.)

$$\{(\alpha x - \beta)(\gamma x - \delta)\}_L \geq 0. \tag{3.22}$$

Observe that if $\alpha x = \beta$, then $(\alpha x - \beta)(\gamma x - \delta) \geq 0$ is valid regardless of the validity of $\gamma x \geq \delta$. Otherwise, we must have $\alpha x \geq \beta + 1$ (or possibly greater than $\beta + 1$, if the structure of $\alpha x \geq \beta$ so permits), and we can then perform standard logical preprocessing tests (zero-one fixing, coefficient reduction, etc. — see Nemhauser and Wolsey, 1988, for example) on the set of constraints $\alpha x \geq \beta + 1$, $\gamma x \geq \delta$, x binary, along with possibly other constraints, to tighten the form of $\gamma x \geq \delta$ to the form $\gamma' x \geq \delta'$. For example, if $\alpha x \geq \beta$ is of the type $(1 - x_j) \geq 0$, for some $j \in \{1, \ldots, n\}$, then the restriction $\alpha x \geq \beta + 1$, x binary, asserts that $x_j = 0$, and so, $\gamma x \geq \delta$ can be tightened under the condition that $x_j = 0$. (Similarly, in a higher-order constraint, if a factor $F_d(J_1, J_2)$ multiplies $\gamma x \geq \delta$, then the latter constraint can be tightened under conditional logical tests based on setting $x_j = 1 \ \forall \ j \in J_1$ and $x_j = 0 \ \forall \ j \in J_2$.)

Additionally, the resulting constraint $\gamma' x \geq \delta'$ can be potentially further tightened by finding the maximum $\theta \geq 0$ for which $\gamma' x \geq \delta' + \theta$ is valid when $\alpha x \geq \beta + 1$ is imposed by considering the problem

$$\theta = -\delta' + \min\{\gamma' x\colon \ \alpha x \geq \beta + 1, \text{any other valid inequalities}, x \text{ binary}\}. \tag{3.23}$$

and by increasing δ' by this quantity θ. Note that, of course, we can simply solve the continuous relaxation of (3.23) and use the resulting value θ after rounding it upwards (using $\theta = 0$ if this value is negative), in order to impose the following SSRLT constraint, in lieu of the weaker restriction (3.22), within the underlying problem:

$$\{(\alpha x - \beta)(\gamma' x - \delta' - \theta)\}_L \geq 0. \tag{3.24}$$

Observe that this also affords the opportunity to now tighten the factor $(\alpha x - \beta)$ in a similar fashion in (3.24), based on the valid constraint $\gamma' x \geq \delta' + \theta$.

3.4.1. Examining Sequential Lifting as an RLT Process

Interestingly, the sequential lifting process of Balas and Zemel (1978), for lifting a minimal cover inequality into a facet for a full-dimensional knapsack polytope defined by the constraint $\sum_{j \in N} a_j x_j \geq b$, x binary, where $0 \leq a_j \leq b \;\forall\; j \in N$, can be viewed as a consequence of the above RLT approach as follows. (Note that full dimensionality requires that $\sum_{j \in N} a_j - a_k \geq b \;\forall\; k \in N$.) Recall that at any stage of the sequential lifting process, given a partially lifted valid inequality $\sum_{j \in T} \gamma_j x_j \geq \delta$, where $T \subset N$, this inequality is lifted to a constraint of the form

$$\sum_{j \in T} \gamma_j x_j \geq \delta + (1 - x_t)\gamma_t, \tag{3.25}$$

where $t \in N - T$. This is done by solving the problem

$$\gamma_t = -\delta + \min\left\{ \sum_{j \in T} \gamma_j x_j : \sum_{j \in N} a_j x_j \geq b, x_t = 0, x \text{ binary} \right\}. \tag{3.26}$$

The lifted constraint given by (3.25) and (3.26) can alternatively be derived by using the foregoing concept. Consider the RLT constraints

$$\left\{x_t\left[\sum_{j\in T}\gamma_j x_j - \delta\right]\right\}_L \geq 0 \text{ and } \left\{(1-x_t)\left[\sum_{j\in T}\gamma_j x_j - \delta\right]\right\}_L \geq 0. \tag{3.27}$$

Treating $(1 - x_t)$ as $\alpha x - \beta$ and $\sum_{j\in T}\gamma_j x_j \geq \delta$ as $\gamma' x \geq \delta'$, we see that θ given by (3.23) is precisely γ_t given by (3.26), when the underlying knapsack constraint is used in place of the "other valid inequalities" in (3.23), since then, $(1 - x_t) \geq 1$, x binary, enforces $x_t = 0$. Hence, the second RLT constraint in (3.27) can be tightened to the form (3.24) given by

$$\left\{(1-x_t)\left[\sum_{j\in T}\gamma_j x_j - \delta - \gamma_t\right]\right\}_L \geq 0. \tag{3.28}$$

Summing (3.28) and the first RLT constraint in (3.27) yields the lifted inequality (3.25).

To illustrate, consider the following knapsack constraint in binary variables x_1, x_2, and x_3: $2x_1 + 3x_2 + 3x_3 \geq 4$. Let us examine the RLT constraint

$$\{x_3[2x_1 + 3x_2 + 3x_3 - 4]\}_L \geq 0. \tag{3.29}$$

Applying the foregoing idea, we can tighten (3.29) under the restriction that $x_3 = 1$, knowing that it is always valid when $x_3 = 0$ regardless of the nonnegativity of any expression contained in $[\cdot]$. However, when $x_3 = 1$, the given knapsack constraint becomes $2x_1 + 3x_2 \geq 1$, which by coefficient reduction, can be tightened to $x_1 + x_2 \geq 1$. Hence, (3.29) can be replaced by the tighter restriction

$$\{x_3[x_1 + x_2 - 1]\}_L \geq 0. \tag{3.30}$$

Similarly, consider the RLT constraint

$$\{(1 - x_3)[2x_1 + 3x_2 + 3x_3 - 4]\}_L \geq 0. \tag{3.31}$$

This time, imposing $(1 - x_3) \geq 1$ or $x_3 = 0$, the knapsack constraint becomes $2x_1 + 3x_2 \geq 4$ which implies via standard logical tests that $x_1 = x_2 = 1$. Hence, we can impose the *equalities*

$$\{(1 - x_3)(x_1 - 1]\}_L = 0 \text{ and } \{(1 - x_3)(x_2 - 1]\}_L = 0 \tag{3.32}$$

in lieu of (3.31), which is now implied. Observe that in this example, the sum of the RLT constraints in (3.30) and (3.32) yield $x_1 + x_2 + x_3 \geq 2$ which happens to be a facet of the knapsack polytope $conv\{x\colon 2x_1 + 3x_2 + 3x_3 \geq 4,\ x \text{ binary}\}$. This facet can alternatively be obtained by lifting the minimal cover inequality $x_1 + x_2 \geq 1$ as in (3.25) and (3.26).

3.4.2. Numerical Example to Illustrate Conditional Logic Based Enhancement of SSRLT

To illustrate the use of the foregoing conditional logic based tightening procedure in the context of solving an optimization problem using general *S*-factors, consider the following problem:

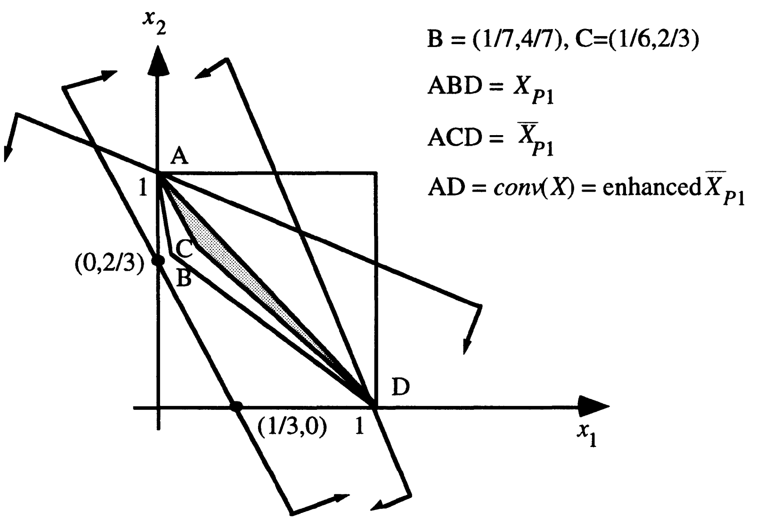


Figure 3.4. Depiction of the various first-level RLT relaxations.

$$\text{Minimize } \{x_1 + x_2 : \ 6x_1 + 3x_2 \geq 2, \ x \in S, \ x \text{ binary}\} \tag{3.33}$$

where $S = \{(x_1, x_2): \ 2x_1 + x_2 \leq 2, \ x_1 + 2x_2 \leq 2, \ x \geq 0\}$.

Figure 3.4 depicts this problem graphically. The integer problem has the optimal value $v(\text{IP}) = 1$, attained at the solution (1,0) or (0,1). The ordinary LP relaxation has the optimal value $v(\text{LP}) = 1/3$ attained at the solution (1/3,0).

Next, let us consider the first level relaxation (RLT-1) produced by RLT, using the factors x_j and $(1 - x_j)$, $j = 1, 2$, to multiply all the problem constraints. Note that $[(2 - 2x_1 - x_2)x_1]_L \geq 0$ yields $w_{12} \leq 0$, which together with $[x_1 x_2]_L \equiv w_{12} \geq 0$, gives $w_{12} = 0$. The other non-redundant constraints defining this relaxation are produced via the product constraints $[(1 - x_1)(1 - x_2)]_L \geq 0$,

$[(6x_1 + 3x_2 - 2)(1 - x_1)]_L \geq 0$ and $[(6x_1 + 3x_2 - 2)(1 - x_2)]_L \geq 0$, which respectively yield (using $w_{12} = 0$), $x_1 + x_2 \leq 1$, $2x_1 + 3x_2 \geq 2$, and $3x_1 + x_2 \geq 1$. This gives (directly in projected form)

$$X_{P1} = \{(x_1, x_2): \; x_1 + x_2 \leq 1 \; , \; 2x_1 + 3x_2 \geq 2, \; 3x_1 + x_2 \geq 1\}.$$

Figure 3.4 depicts this region. The optimal value using this relaxation is given by ν(RLT-1) = 5/7, attained at the solution (1/7, 4/7).

On the other hand, to construct the first level relaxation (SSRLT-1) produced by SSRLT, we would employ the S-factors given by the constraints in (3.33). As in RLT, we obtain $w_{12} = 0$, and using this, the other non-redundant constraints defining this relaxation are produced via the product constraints

$[(2 - 2x_1 - x_2)(2 - x_1 - 2x_2)]_L \geq 0$,

$[(6x_1 + 3x_2 - 2)(2 - 2x_1 - x_2)]_L \geq 0$, and

$[(6x_1 + 3x_2 - 2)(2 - x_1 - 2x_2)]_L \geq 0$, which respectively yield, $x_1 + x_2 \leq 1$, $4x_1 + 5x_2 \geq 4$, and $2x_1 + x_2 \geq 1$. This gives (directly in projected form)

$$\overline{X}_{P1} = \{(x_1 + x_2): \; x_1 + x_2 \leq 1, \; 4x_1 + 5x_2 \geq 4, \; 2x_1 + x_2 \geq 1\}.$$

Figure 3.4 depicts this region. Note that $\overline{X}_{P1} \subset X_{P1}$ and that the optimal value of this relaxation is given by ν(SSRLT-1) = 5/6 > ν(RLT-1), and is attained at the solution (1/6, 2/3).

Now, consider an enhancement of SSRLT-1 using conditional logic (a similar enhancement can be exhibited for RLT-1). Specifically, consider the product constraint $[(2 - 2x_1 - x_2) \quad (6x_1 + 3x_2 - 2)]_L \geq 0$ of the form (3.22). Imposing $(2 - 2x_1 - x_2) \geq 1$ as for (3.22), *i.e.*, $2x_1 + x_2 \leq 1$, yields $x_1 = 0$ by a standard logical test. This together with $6x_1 + 3x_2 \geq 2$ implies that $x_2 = 1$. Hence, the tightened form of this constraint is $(2 - 2x_1 - x_2)(x_1) = 0$ and $(2 - 2x_1 - x_2)(1 - x_2) = 0$. The first constraint yields $w_{12} = 0$ (as before), while the second constraint states that $x_1 + x_2 = 1$. This produces the convex hull representation, and so, this enhanced relaxation now recovers an optimal integer solution.

3.4.3. Application to the Traveling Salesman Problem

In this example, we demonstrate the potential utility of the various concepts developed in this chapter by applying them to the celebrated traveling salesman problem (TSP). We assume the case of a general (asymmetric) TSP defined on a totally dense graph (see Lawler *et al.*, 1985, for example). For this problem, Desrochers and Laporte (1991) have derived a strengthened version of the Miller-Tucker-Zemlin (MTZ) formulation obtained by lifting the MTZ-subtour elimination constraints into facets of the underlying TSP polytope. While the traditional MTZ formulation of TSP is well known to yield weak relaxations, Desrochers and Laporte exhibit computationally that their lifted-MTZ formulation significantly tightens this representation. We show below that an application of SSRLT concepts to the MTZ formulation of TSP, used in concert with the conditional logic based strengthening procedure, *automatically recovers* the formulation of Desrochers and Laporte.

Toward this end, consider the following statement of the asymmetric traveling salesman problem, where $x_{ij} = 1$ if the tour proceeds from city i to city j, and is 0 otherwise, for all $i, j, = 1, \ldots, n,\ i \neq j$.

ATSP: Minimize $\sum_i \sum_{j \neq i} c_{ij} x_{ij}$

subject to:

$$\sum_{j \neq i} x_{ij} = 1 \qquad \forall\ i = 1, \ldots, n \tag{3.34a}$$

$$\sum_{i \neq j} x_{ij} = 1 \qquad \forall\ j = 1, \ldots, n \tag{3.34b}$$

$$u_j \geq (u_i + 1) - (n-1)(1 - x_{ij}) \qquad \forall\ i, j \geq 2, i \neq j \tag{3.34c}$$

$$1 \leq u_j \leq (n-1) \qquad \forall\ j = 2, \ldots, n \tag{3.34d}$$

$$x_{ij} \text{ binary} \qquad \forall\ i, j = 1, \ldots, n, i \neq j. \tag{3.34e}$$

Note that (3.34a, b, and e) represent the assignment constraints and (3.34c) and (3.34d) are the MTZ subtour elimination constraints. These latter constraints are derived based on letting u_j represent the rank order in which city j is visited, using $u_1 \equiv 0$, and enforcing that $u_j = u_i + 1$ whenever $x_{ij} = 1$ in any binary feasible solution. Now, in order to construct a suitable reformulation using SSRLT, let us compose the set S as follows, and include this set of implied inequalities within the problem ATSP stated above.

$$S \equiv \{x: \; x_{ij} + x_{ji} \leq 1 \; \forall i, j \geq 2, \; i \neq j, \; x_{1j} + x_{j1} \leq 1 \; \forall j \geq 2,$$
$$x_{ij} \geq 0 \; \forall i, j, i \neq j\}. \tag{3.35}$$

Note that S is comprised of simple two-city subtour elimination constraints of the form proposed by Dantzig, Fulkerson, and Johnson (1954). Next, let us construct the following selected S-factor constraint products. First, consider the product constraint generated by multiplying (3.34c) with the S-factor x_{ij}. Using conditional logic as with (3.22), and noting that we can impose $u_j = (u_i + 1)$ when $x_{ij} = 1$, this yields the constraint

$$x_{ij}u_j = x_{ij}(u_i + 1) \qquad \forall \; i, j \geq 2, i \neq j. \tag{3.36}$$

Similarly, considering $[x_{1j}(u_j - 1)]_L \geq 0$ from (3.34d), and enhancing this by the conditional logic that $u_j = 1$ when $x_{1j} = 1$, we get

$$x_{1j}u_j = x_{1j} \qquad \forall \; j = 2, \ldots, n. \tag{3.37}$$

Repeating this with the upper bounding constraint in (3.34d), we can enhance $[x_{j1}(n - 1 - u_j)]_L \geq 0$ to the following constraint, noting that $u_j = (n - 1)$ if $x_{j1} = 1$:

$$x_{j1}u_j = (n - 1)x_{j1} \qquad \forall \; j = 2, \ldots, n. \tag{3.38}$$

Next, let us consider the product of (3.34c) with the S-factor $(1 - x_{ij} - x_{ji}) \geq 0$. This gives the constraint

$$[(1 - x_{ij} - x_{ji})(u_j - u_i - 1 + (n-1)(1 - x_{ij}))]_L \geq 0.$$

Using $x_{ij}^2 = x_{ij}$ and $x_{ij}x_{ji} = 0$ (since $x_{ij} + x_{ji} \leq 1$), the foregoing constraint becomes

$$(u_j - u_i) \geq (u_j - u_i)(x_{ij} + x_{ji}) - (n-2)(1 - x_{ij}) + (n-2)x_{ji}.$$

From (3.36), we get $(u_j - u_i)x_{ij} = x_{ij}$, and upon interchanging i and j, this yields $(u_j - u_i)x_{ji} = -x_{ji}$. Substituting this into the foregoing SSRLT constraint, we obtain the valid inequality

$$u_j \geq (u_i + 1) - (n-1)(1 - x_{ij}) + (n-3)x_{ji} \qquad \forall\ i, j \geq 2, i \neq j. \tag{3.39}$$

Similarly, multiplying (3.34d) with the S-factor $(1 - x_{1j} - x_{j1}) \geq 0$ yields $[(1 - x_{1j} - x_{j1})(u_j - 1)]_L \geq 0$ and $[(1 - x_{1j} - x_{j1})(n - 1 - u_j)]_L \geq 0$. Using the conditional logic procedure, under $x_{1j} = x_{j1} = 0$, these constraints can be respectively tightened to $[(1 - x_{1j} - x_{j1})(u_j - 2)]_L \geq 0$ and $[(1 - x_{1j} - x_{j1})(n - 2 - u_j)]_L \geq 0$. Simplifying these products and using (3.37) and (3.38) yields the constraints

$$1 + (1 - x_{1j}) + (n-3)x_{j1} \leq u_j \leq (n-1) - (1 - x_{j1}) - (n-3)x_{1j}. \tag{3.40}$$

Observe that (3.39) and (3.40) are tightened versions of (3.34c) and (3.34d), respectively, and are precisely the facet-defining, lifted-MTZ constraints derived by Desrochers and Laporte (1991). Hence, we have shown that a selected application of SSRLT used in concert with our conditional logic based strengthening procedure *automatically* generates this improved MTZ formulation. Sherali and Driscoll (1996) have developed further enhancements that tighten and subsume this formulation for both the ordinary as well as for the precedence constrained version of the asymmetric traveling salesman problem, using these SSRLT constructs along with conditional logical implications.

4

RLT HIERARCHY FOR GENERAL DISCRETE MIXED-INTEGER PROBLEMS

Thus far, we have been focusing on 0-1 mixed-integer programming problems and have developed a general theory for generating a hierarchy of tight relaxations leading to the convex hull representation. However, in many applications, we encounter problems in which the decision variables are required to take on more general discrete or integer values as opposed to simply zero or one values. Of particular interest in this context are classes of problems involving variables that are restricted to taken on only a few discrete values, that are possibly not even consecutive integers, for which one might expect to have weak ordinary linear programming relaxations. In this chapter, we demonstrate how one can generate a hierarchy of relaxations leading to the convex hull representation for such general bounded variable discrete optimization problems.

As the reader might anticipate, one can always represent such a problem as an equivalent zero-one problem to which the development in the foregoing chapters could be applied. The issue of interest, however, is to devise a mechanism for generating such a hierarchy directly in the original variable space without resorting to such a transformation. As we shall see in the sequel, it is possible to translate the developed theory as applied to the transformed zero-one problem into one that is directly applicable in the original variable space in order to reveal a new RLT methodology. This approach also provides insights into how various classes of valid inequalities for 0-1 problems could be translated to classes of valid inequalities for general integer programming problems. This is an issue that is of considerable contemporary interest.

The remainder of this chapter is organized as follows. In Section 4.1 we introduce the discrete problem of concern and formulate its equivalent 0-1 representation. The concepts of Chapter 3 are then applied to this representation in Section 4.2 in order to develop a hierarchy of relaxations in the transformed 0-1 space. A parallel translation of this methodology that is directly applicable in the original variable space itself is developed in Section 4.3 through the concept of an inverse transformation. The equivalence of these hierarchies in the transformed 0-1 and original variable spaces is established in Section 4.4. An illustrative example is presented in Section 4.5, and the chapter closes with some discussion in Section 4.6 that provides insights related to how classes of valid inequalities for (generalized upper bounded) 0-1 problems can be translated to generate counterpart valid inequalities for general discrete programs.

4.1. The Discrete Problem and its Equivalent Zero-One Representation

Consider the feasible region of a discrete "mixed-integer" programming problem specified in the form

$$X = \{(x, y) \in R^n \times R^m : Ax + Dy \geq b, x_j \in S_j \ \forall \ j = 1, ..., n, y \geq 0\} \tag{4.1a}$$

where

$$S_j = \{\theta_{jk}, k = 1, ..., k_j\} \quad \forall \ j = 1, ..., n \tag{4.1b}$$

and where for each $j = 1, ..., n$, θ_{jk}, for $k = 1, ..., k_j$, are some real distinct numbers, not necessarily integral or of any particular sign. A bounded variable integer program would be a special case of this type of a situation in which the sets S_j would typically be comprised of a set of consecutive nonnegative integers.

We can convert (4.1) into an equivalent zero-one mixed-integer representation by using the transformation

$$\{x_j = \sum_{k=1}^{k_j} \theta_{jk}\lambda_{jk}, \text{ where } \sum_{k=1}^{k_j} \lambda_{jk} = 1, \ \lambda_{jk} \in \{0, 1\} \quad \forall \ k = 1, ..., k_j\}$$

for each $j = 1, ..., n$. (4.2)

Under the transformation (4.2), we can equivalently represent X in the expanded variable space (x, y, λ) as

$$X^{\lambda} = \{(x, y, \lambda) : \quad Ax + Dy \geq b \tag{4.3a}$$

$$x_j = \sum_{k=1}^{k_j} \theta_{jk}\lambda_{jk} \quad \forall \ j = 1, ..., n \tag{4.3b}$$

$$\sum_{k=1}^{k_j} \lambda_{jk} = 1 \quad \forall \ j = 1, ..., n \tag{4.3c}$$

$$y \geq 0 \tag{4.3d}$$

$$\lambda \geq 0 \text{ and binary}\}. \tag{4.3e}$$

Note that we could have eliminated the x-variables from the representation (4.3) and written this transformed set in only the variables y and λ. However, we shall find it convenient to retain the x-variables in this representation in order to make the development more transparent. More importantly, observe that (4.3) possesses a 0-1 generalized upper bounding (GUB) constraint structure that renders it amenable to applying the SSRLT procedure developed in the previous chapter. This is done next in the following section.

4.2. Hierarchy of Relaxations in the Transformed Zero-One Space

Consider the set X^{λ} defined by (4.3) and let us identify this with (3.1) by treating the x and y variables as continuous, and the GUB constraints (4.3c, e) as comprising the specially structured set "S" of (3.1b). Observe that as discussed in Section 3.3, since the maximum number of λ-variables that can be simultaneously equal to one is n, we only need to generate a hierarchy for levels $d = 1, \ldots, n$ using the constructs of Chapter 3, and we would obtain a convex hull representation at level $d = n$. Moreover, recalling the comments in Sections 3.1 and 3.3 regarding equality and GUB constraints, the SSRLT process at level $d \in \{1, \ldots, n\}$, denoted SSRLT(d), would proceed as follows. Note that in the Reformulation Phase below, the factors of order d that are comprised of products of λ-variables need only be composed as having at most one variable from each of the sets $\{\lambda_{jk}, k = 1, \ldots, k_j\}$, because $\lambda_{jk_1} \cdot \lambda_{jk_2} \equiv 0 \ \forall\ k_1 \neq k_2, \forall\ j$, due to the GUB constraints (4.3c,e). Furthermore, whenever a factor is composed of products of some λ-

variables and the GUB constraint factors $(\sum_k \lambda_{jk} - 1)$, the indices j involved in each of these variables and GUB components of the factor must necessarily be distinct, or else, this factor would be either null or of order less than d. By Lemma 2.1, such factors would be implied by other generated factors. In addition, whenever this factor contains at least one GUB component, the resultant RLT constraint that is generated by it is an equality restriction. Consequently, as explained in Section 3.1, factors containing such GUB components need only be used to multiply the constraints (4.3d); their application to the other constraints would result in restrictions that are implied by the other RLT constraints that are being generated below.

To elucidate further, consider for example the *equality* constraint generated by multiplying an inequality in (4.3a) with a factor that contains a GUB component $(\sum_k \lambda_{jk} - 1)$. In the resulting constraint, the right-hand side would be zero by virtue of $R(c)$ below, each y_i variable in the constraint times this factor would be zero by virtue of $R(d)$ below, and each x_j variable in this constraint times this factor would, by $R(b)$ below, simplify to an expression that equals a weighted sum of products of some λ-variables times the GUB constraint factor, which would again be zero by virtue of $R(c)$. This fact, along with Theorems 3.1 and 3.2, also implies that the original constraints (4.3a) are implied by the level d relaxation SSRLT(d). The relaxation, in simplified form, is generated as follows.

Reformulation Phase

Construct the following RLT product constraints.

R(a) Multiply (4.3a) by all possible product factors composed by selecting some d distinct indices j, and for each selected index, choosing some variable λ_{jk} for $k \in \{1, ..., k_j\}$.

R(b) Multiply (4.3b) for each j by all possible product factors composed by selecting some r distinct indices $q \neq j$, and for each selected index q, choosing some variable λ_{qk} for $k \in \{1, ..., k_q\}$, where $r = 0, 1, ..., d$. (Note that for $r \equiv 0$, this factor is equal to one, and simply reproduces the original constraints.)

R(c) Multiply (4.3c) for each j by all possible factors of the type in R(b) above.

R(d) Multiply (4.3d) by all possible product factors composed by selecting some d distinct indices j, and for each selected index j, choosing either some variable λ_{jk} for $k \in \{1, ..., k_j\}$ or the expression $(\sum_k \lambda_{jk} - 1)$. Note that whenever a factor includes any GUB term of the latter type, the resulting RLT constraint is written as an equality.

R(e) Include **nonnegativity restrictions** on all terms produced above that are composed as products of λ-variables.

R(f) Use the **identities** given by (4.4) below in the resulting nonlinear program produced by the foregoing operations.

$$\lambda_{jk_1}\lambda_{jk_2} = 0 \quad \forall\, k_1 \neq k_2,\ \forall\, j, \qquad \lambda_{jk}^2 = \lambda_{jk} \quad \forall\, j, k \tag{4.4a}$$

$$x_j\lambda_{jk} = \theta_{jk}\lambda_{jk} \qquad \forall\, k = 1, ..., k_j, \quad \forall\, j. \tag{4.4b}$$

Remark 4.1. Note that the restrictions (4.4a) that are a consequence of standard RLT products as illustrated in Section 3.1 have already been *implicitly* applied in the foregoing description of the equivalent reduced RLT process. Moreover, (4.4b) is a consequence of multiplying (4.3b) for each j with some λ_{jk}, $k \in \{1, ..., k_j\}$, and applying (4.4a). This has also been partially used above to avoid unnecessary products, as for example, in R(b) where we have employed factors having component indices q that are different from the corresponding constraint index j.

Linearization Phase

Linearize the polynomial program (in variables x, y, and λ) produced by the Reformulation Phase by substituting

$w_{\cdot}$ -variables for each distinct product of λ-variables (4.5a)

$u_{j\cdot}$-variables for each distinct product of x_j with λ-variables, $\forall\ j$ (4.5b)

$v_{t\cdot}$-variables for each distinct product of y_t with λ-variables, $\forall\ t$ (4.5c)

where each dot $(\cdot)$ in (4.5) denotes the appropriate collection of indices to represent the corresponding distinct λ-variable factors in the product term, and where as before, note that in (4.5b), x_j appears in a product term with λ_{qk} types of variables for indices q that are distinct from j.

This RLT process produces a polyhedron $X(d)$ at level d in the lifted higher dimensional space of the variables (x, y, λ, w, u, v). From Chapter 3, we know that as d varies from

$0,1,\ldots,n$, we obtain a progression of tighter relaxations as (implicitly) viewed in the projected (x, y, λ) space, with the representation at level n leading to the convex hull of X^{λ}. The projection of X^{λ} onto the original (x, y) space, would equivalently produce $conv(X)$. We now proceed to construct a parallel hierarchy of relaxations by designing an RLT process that is applied directly to X itself, rather than to the transformed equivalent set X^{λ}.

4.3. Structure of a Parallel Hierarchy in the Original Variable Space

Note that in the previous section, the design of the RLT process as applied to X^{λ} was naturally driven by the λ-variables. The key insight into deriving an equivalent process that can be applied directly to X itself without employing the transformation (4.2) lies in the fact that the inverse relationship to (4.2) is given by

$$\lambda_{jk} = \frac{\prod_{p \neq k}(x_j - \theta_{jp})}{\prod_{p \neq k}(\theta_{jk} - \theta_{jp})} \equiv L_{jk}, \text{ say, } \forall\ k = 1,\ldots,k_j,\ \ \forall\ j. \tag{4.6}$$

Observe in (4.6) that whenever $x_j = \theta_{jk}$, we have that $\lambda_{jk} = 1$, and whenever x_j takes on one of its other discrete values θ_{jp}, $p \neq k$, we obtain $\lambda_{jk} = 0$. The polynomial expression in x_j, denoted L_{jk} in (4.6), is known as the Lagrange Interpolation Polynomial **(LIP)** (see Volkov (1990), for example) and denotes the unique polynomial of order $k_j - 1$ that achieves a value of one at the grid point $x_j = \theta_{jk}$ and a value of 0 at the other grid points $x_j = \theta_{jp}$, $p = 1,\ldots,k_j$, $p \neq k$. In particular, if $S_j = \{0, 1\}$, then L_{jk} reduces to $(1 - x_j)$ and x_j for $k = 1$ and 2, respectively. Hence, in this case, the LIP-factors L_{j1} and L_{j2} are simply the regular bound-factors for binary variables.

Now, consider the following *integer-programming RLT* process **IP-RLT(d)** at level $d \in \{0, 1, \ldots, n\}$, using the Lagrange interpolating polynomials (4.6), and noting that (4.3a) and (4.3d) represent the inequalities defining X.

Reformulation Phase

R′(a) Multiply (4.3a) by all possible product factors composed by selecting some d distinct indices j, and for each selected index, choosing some LIP L_{jk} for $k \in \{1, \ldots, k_j\}$.

R′(b) Multiply (4.3d) by all possible factors of the type in R′(a) above.

R′(c) Include **nonnegativity restrictions** on all possible product factors composed by selecting some $D = \min\{d + 1, n\}$ distinct indices , and for each selected index j, choosing some LIP L_{jk} for $k \in \{1, \ldots, k_j\}$.

R′(d) Use the **identities** given by (4.7) below in the resulting nonlinear program produced by the foregoing operations. (Observe that this directly extends (4.4b) to the present case, noting (4.6), and is the critical step that leads to a tightening of the relaxation. In particular, of $S_j = \{0, 1\}$, then (4.7) reduces to the identities $x_j(1 - x_j) = 0$ and $x_j^2 = x_j$, which happen to coincide in this case.)

$$x_j L_{jk} = \theta_{jk} L_{jk} \qquad \forall\ k \in \{1, \ldots, k_j\}, \quad \forall\ j \tag{4.7}$$

Linearization Phase

Linearize the polynomial program (in variables x and y) produced by the Reformulation Phase by substituting

$z_{\cdot}$ -variables for each distinct product of x-variables (4.8a)

$\xi_{t\cdot}$-variables for each distinct product of y_t with x-variables (4.8b)

where again, each dot $(\cdot)$ in (4.8) denotes an appropriate collection of indices to represent the corresponding (not necessarily distinct) x-variable factors in the product term.

This IP-RLT(d) process produces a polyhedron $X'(d)$ at level d in the lifted higher dimensional space of the variables (x, y, z, ξ). We would like to show that as d varies from $0,1,\ldots,n$, the (implicit) projection of these polyhedra onto the original space of the (x, y) variables yields a hierarchy of tighter relaxations that leads to the convex hull of X at level $d = n$. This is done in the next section.

4.4. Equivalence of the Hierarchies in the Transformed and Original Spaces

In order to establish the claim that IP-RLT(d), $d = 0, 1, \ldots, n$, produces a hierarchy of relaxations leading to the convex hull representation, let us begin by adding a set of redundant constraints to $X'(d)$, and using a set of additional variables to represent certain expressions, for the sake of convenience. This produces an augmented polyhedron $X'_+(d)$ as described below.

Generation of Polyhedron $\mathbf{X}'_{+}(d)$:

Reformulation Phase

Augment the steps R′(a), R′(b), R′(c) and R′(d) of the Reformulation Phase of IP-RLT(d) by the following additional steps. The redundancy of these steps is exhibited below in Remark 4.2.

R′(e) For each j, **multiply the identity**

$$\left[x_j - \sum_{k=1}^{k_j} \theta_{jk} L_{jk}\right] = 0 \tag{4.9}$$

by all possible product factors composed by selecting some r distinct indices $q \neq j$, and for each selected index q, choosing some LIP L_{qk} for $k \in \{1, \ldots, k_q\}$, where $r = 0, 1, \ldots, d$.

R′(f) For each j, **multiply the identity**

$$\left[\sum_{k=1}^{k_j} L_{jk} - 1\right] = 0 \tag{4.10}$$

by all possible factors of the type in R′(e) above.

R′(g) Multiply (4.3d) by all possible product factors composed by selecting some d distinct indices j, and for each selected index j, choosing either some LIP L_{jk} or the expression $\left[\sum_k L_{jk} - 1\right]$ as a component term, where the latter expression is selected for at least one index j in each factor. Set the resultant product constraint expression equal to zero.

R′(h) Include **nonnegativity restrictions** on all possible product factors composed by selecting some r distinct indices j, and for each selected index j, choosing some LIP L_{jk}, $k \in \{1, ..., k_j\}$, for $r = 0, 1, ..., d$.

Remark 4.2. Note that with L_{jk} defined as the polynomial expression in (4.6), it is readily verified that the expression $\sum_k \theta_{jk} L_{jk}$ simplifies to x_j $\forall$ j. Hence, the identity (4.9) is a "null constraint" of the type "0=0", and so, the RLT constraints produced by R′(e) are null equations. Similarly, R′(f) and R′(g) produce null equations upon applying the stated RLT process because the expression $\sum_k L_{jk}$ simplifies to equal 1 $\forall$ j, when L_{jk}, $\forall$ j, k, are given by (4.6). Finally, the constraints generated via R′(h) are also redundant since they are implied via surrogates of constraints in R′(c) above. This can be seen by using an inductive process similar to that in the proof of Lemma 2.1, along with the identity (4.10).

Remark 4.3. It is insightful at this stage to note the relationships between the foregoing RLT process and that of Section 4.2, as delineated below.

R′(a) relates to R(a)

R′(e) relates to R(b)

R′(f) relates to R(c)

R′(b) and R′(g) jointly relate to R(d)

R'(c) and R'(h) jointly relate to R(e)

and R'(d) relates to R(b), noting Remark 4.1.

Motivated by Remark 4.3, we substitute the linearization process for IP-RLT(d) by the following *equivalent* steps.

Linearization Phase

Step 1. Substitute the variable λ_{jk} in place of each nonlinear expression L_{jk} appearing in the reformulated problem $\forall\ j,\ k$.

Step 2. Linearize the resulting problem using the substitution (4.5).

Step 3. To account for the actual substitutions (4.8) that ought to have been used in lieu of Steps 1 and 2 above, include the following explicit variable defining constraints.

$$\lambda_{jk} = [L_{jk}]_{\ell \text{ under } (4.8)} \quad \forall\ j, k \tag{4.11a}$$

$$w_{\cdot} = \begin{bmatrix} \text{products of LIPs } L_{jk} \text{ for distinct } j, \\ k \in \{1, \ldots, k_j\}, \text{ as contained in } (\cdot) \end{bmatrix}_{\ell \text{ under } (4.8)} \quad \forall\ w_{\cdot} \tag{4.11b}$$

$$u_{j\cdot} = \begin{bmatrix} (x_j) * \text{ products of LIPs } L_{qk} \text{ for distinct } q \neq j, \\ k \in \{1, \ldots, k_q\}, \text{ as contained in } (\cdot) \end{bmatrix}_{\ell \text{ under } (4.8)} \quad \forall\ u_{j\cdot} \tag{4.11c}$$

$$v_{t\cdot} = \begin{bmatrix} (y_t) * \text{ products of LIPs } L_{jk} \text{ for distinct } j, \\ k \in \{1, \ldots, k_j\}, \text{ as contained in } (\cdot) \end{bmatrix}_{\ell \text{ under } (4.8)} \quad \forall\ v_{t\cdot} \tag{4.11d}$$

where $[\cdot]_{\ell \text{ under } (4.8)}$ denotes the linearization of the expression $[\cdot]$ under the substitution (4.8).

This produces the polyhedron $X'_+(d)$ in the space of the variables $(x, y, \lambda, w, u, v, z, \xi)$ which reduces to $X'(d)$ in the variables (x, y, z, ξ) by substituting out the variables (λ, w, u, v) using (4.11), and deleting null and redundant constraints. Moreover, we have that

$$X'_+(d) \equiv \{(x, y, \lambda, w, u, v, z, \xi) : \text{ constraints defining } X(d) \text{ hold true,}$$
$$\text{and the variable identities in (4.11) hold true}\}. \tag{4.12}$$

We now establish the desired equivalence relationship between $X'_+(d)$ and $X(d)$, and thereby validate the claim regarding the hierarchy generated by IP-RLT(d). Before presenting this main result, consider the following technical lemma.

Lemma 4.1. *For any j, k, consider the LIP expression for* L_{jk} *as a function of* x_j *given by (4.6), and suppose that we substitute*

$$x_j = \sum_q \theta_{jq}\lambda_{jq}, \text{ where } \lambda_{jq_1}\lambda_{jq_2} = 0 \ \forall \ q_1 \neq q_2, \text{ and } \lambda_{jq}^2 = \lambda_{jq} \ \forall \ q. \tag{4.13}$$

Then this expression simplifies to

$$L_{jk} = \lambda_{jk} + \left[\frac{(-1)^\sigma \prod_{p \neq k} \theta_{jp}}{\prod_{p \neq k} (\theta_{jk} - \theta_{jp})}\right](1 - \sum_q \lambda_{jq}), \text{ where } \sigma = k_j - 1. \tag{4.14}$$

Proof. Under (4.13), the numerator (NUM, say) of L_{jk} in (4.6) is given by

$$\text{NUM} = \prod_{p \neq k}\left[\sum_q \theta_{jq}\lambda_{jq} - \theta_{jp}\right] = \prod_{p \neq k}\left\{(\theta_{jk} - \theta_{jp})\lambda_{jk} + \sum_{q \neq k}\theta_{jq}\lambda_{jq} - \theta_{jp}(1 - \lambda_{jk})\right\}$$

$$= \prod_{p \neq k}(\theta_{jk} - \theta_{jp})\lambda_{jk} + \prod_{p \neq k}\left\{\sum_{q \neq k}\theta_{jq}\lambda_{jq} - \theta_{jp}(1 - \lambda_{jk})\right\}$$

since by (4.13), $\lambda_{jk}\lambda_{jq} = 0 \ \forall \ q \neq k$, and $\lambda_{jk}(1 - \lambda_{jk}) = 0$ as well. Again, since the product of any pair of terms in $\{\lambda_{jq}$ for $q \neq k$, $(1 - \sum_q \lambda_{jq})\}$ is zero, we have,

$$\text{NUM} = \prod_{p \neq k}(\theta_{jk} - \theta_{jp})\lambda_{jk} + \prod_{p \neq k}\left\{\sum_{q \neq k}(\theta_{jq} - \theta_{jp})\lambda_{jq} - \theta_{jp}\left[1 - \sum_q \lambda_{jq}\right]\right\}$$

$$= \prod_{p \neq k}(\theta_{jk} - \theta_{jp})\lambda_{jk} + \sum_{q \neq k}\lambda_{jq}\left[\prod_{p \neq k}(\theta_{jq} - \theta_{jp})\right] + \left[(-1)^{\sigma}\prod_{p \neq k}\theta_{jp}\right]\left[1 - \sum_q \lambda_{jq}\right]$$

$$= \prod_{p \neq k}(\theta_{jk} - \theta_{jp})\lambda_{jk} + \left[(-1)^{\sigma}\prod_{p \neq k}\theta_{jp}\right]\left[1 - \sum_q \lambda_{jq}\right]$$

since $\prod_{p \neq k}(\theta_{jq} - \theta_{jp}) = 0 \ \forall \ q \neq k$. Noting the denominator of L_{jk} in (4.6), this completes the proof. □

Theorem 4.1. *The polyhedra $X(d)$ and $X'_+(d)$ are equivalent in the sense that for any feasible solution $(x, y, \lambda, w, u, v, z, \xi)$ to $X'_+(d)$ we have that (x, y, λ, w, u, v) is feasible to $X(d)$, and conversely, for any feasible solution (x, y, λ, w, u, v) to $X(d)$, there exists a (z, ξ) such that $(x, y, \lambda, w, u, v, z, \xi)$ is feasible to $X'_+(d)$. Consequently, IP-RLT(d) produces a hierarchy of relaxations for $d = 0, 1, \ldots, n$, leading to the convex hull representation of X in the projected space of the (x, y) variables at level $d = n$.*

Proof. Given $(x, y, \lambda, w, u, v, z, \xi) \in X'_{+}(d)$, we have by (4.12) that $(x, y, \lambda, w, u, v) \in X(d)$. Hence, suppose that we are given a feasible solution $(x, y, \lambda, w, u, v) \in X(d)$. Let us define z and ξ as follows.

$$\text{Each } z\text{-variable} = \left[\begin{array}{l}\text{corresponding product of right-hand sides of (4.3b)} \\ \text{for the } x\text{-variables whose products define this } z\text{-variable} \\ \text{in (4.8a), using (4.4a) to simplify}\end{array}\right]_{\ell \text{ under (4.5)}} \tag{4.15a}$$

$$\text{Each } \xi_{t}\text{-variable} = \left[\begin{array}{l}(y_t) * \text{(corresponding product of right-hand sides} \\ \text{of (4.3b) for the } x\text{-variables whose products define} \\ \text{this } \xi\text{-variable in (4.8b), using (4.4a) to simplify}\end{array}\right]_{\ell \text{ under (4.5)}} \tag{4.15b}$$

where as before, $[\cdot]_{\ell \text{ under (4.5)}}$ denotes the linearization operation under the substitution (4.5). We will show that using the values of z and ξ as specified by (4.15), along with the given values of $x, y, \lambda, w, u,$ and v, we obtain that these values satisfy (4.11), and so by (4.12), we will have $(x, y, \lambda, w, u, v, z, \xi) \in X'_{+}(d)$.

Toward this end, first of all, note that the z and ξ variables are being defined in (4.15) by using the substitution and simplification embodied in (4.13) within the corresponding product expressions in (4.8), and linearizing via (4.5). Hence, we evidently obtain for each constraint in (4.11), denoting "right-hand side" by RHS in general, that

$$\left[\begin{array}{l}\text{Expression on RHS of the} \\ \text{constraint in (4.11) when} \\ z \text{ and } \xi \text{ are substituted for as} \\ \text{in (4.15)}\end{array}\right] = \left[\begin{array}{l}\text{Expression obtained for this RHS} \\ \text{by substituting for } x_j \text{ and simplifying} \\ \text{as in (4.13), and then applying the} \\ \text{linearization (4.5).}\end{array}\right] \tag{4.16}$$

Hence, consider (4.11a) under (4.15). By (4.16) and (4.14) of Lemma 4.1, we get

[RHS for (4.11a)] $= \lambda_{jk}$ since $(1 - \sum_q \lambda_{jq}) = 0$ by (4.3c) for the given solution. Hence, (4.11a) is satisfied.

Similarly, by (4.16) and (4.14), of RHS of (4.11b) is given by the corresponding products of the right-hand sides of (4.14), linearized under (4.5). But by R(c) under (4.5), this simplifies to

[products of the corresponding λ_{jk}-variables $]_{\ell \text{ under } (4.5)} \equiv w_{(\cdot)}$.

Hence, (4.11b) holds true.

Next, consider (4.11c). Here, we get via (4.16) and (4.14) that

[RHS of (4.11c)]

$$= \left[\left(\sum_k \theta_{jk} \lambda_{jk} \right) \cdot \left(\text{products of corresponding expressions on RHS of (4.14)} \right) \right]_{\ell \text{ under } (4.5)}$$

which, via R(c) and (4.5), is

$$= \left[\left(\sum_k \theta_{jk} \lambda_{jk} \right) \cdot \left(\text{products of corresponding } \lambda_{q\ell}\text{-variables} \right) \right]_{\ell \text{ under } (4.5)}$$

which, via R(b) and (4.5), is

$$= \left[\left(x_j \right) \cdot \left(\text{products of corresponding } \lambda_{q\ell}\text{-variables} \right) \right]_{\ell \text{ under } (4.5)}$$

which equals $u_{j\cdot}$. Hence, (4.11c) holds true.

Finally, by (4.16) and (4.14), the RHS of (4.11d) is given by the product of y_t times the products of the right-hand sides of the corresponding relations (4.14). By R(d) under (4.5), this simplifies to

$$\left[(y_t) * \left(\text{products of the corresponding } \lambda_{jk}\text{-variables}\right)\right]_{\ell \text{ under (4.5)}} \equiv v_t,$$

and so, (4.11d) holds true as well. This establishes the stated equivalence between $X(d)$ and $X'_+(d)$.

Consequently, the projection of $X'_+(d)$ onto the space of the variables (x, y, λ, w, u, v) is precisely $X(d)$. But by Chapter 3, $X(d)$ produces the desired hierarchy of relaxations as d varies from $0, 1, \ldots, n$, leading to the convex hull representation at level $d = n$, when viewed in the projected (x, y, λ) space or the projected (x, y) space. Hence, so does $X'_+(d)$. Noting the equivalence between $X'(d)$ produced by IP-RLT(d) and $X'_+(d)$, this completes the proof. □

Remark 4.4. In the proof of Theorem 4.1, we showed, in particular, that given a solution $(x, y, \lambda, w, u, v) \in X(d)$, there exists a (z, ξ) given by (4.15) such that the constraints (4.11) are satisfied, so that $(x, y, \lambda, w, u, v, z, \xi)$ then belongs to $X'_+(d)$. A question of interest is whether this prescription of (z, ξ) is unique so that the relationship among these solutions is actually a *one-to-one transformation.* The answer is affirmative. This can be shown by demonstrating that if (4.11) are surrogated by composing on the left-hand side (LHS) of (4.11) the same expression as on the RHS of

(4.15a), then we will obtain the corresponding $z_.$ variable on the RHS of (4.11). Hence, $z_.$ would be uniquely given by this surrogate expression, which precisely coincides with (4.15a). The case of $\xi_.$ as given by (4.15b) is similar. This is demonstrated by the following Corollary 4.1 that uses the technical result of general interest stated as Lemma 4.2 below.

Lemma 4.2. *Consider the LIPs L_{jk} $\forall k$, defined as functions of x_j for $k = 1, \ldots, k_j$. Then, for any $0 \le n_j \le k_j - 1$, we have,*

$$\sum_{k=1}^{k_j} \theta_{jk}^{n_j} L_{jk} = x_j^{n_j}. \tag{4.17}$$

Proof. Observe that the left-hand side of (4.17) is a polynomial of degree at most $(k_j - 1)$, which by (4.6), has the property that it takes on the values of $x_j^{n_j}$ for $x_j = \theta_{jk}$, $k = 1, \ldots, k_j$, for any $0 \le n_j \le k_j - 1$. The relation (4.17) is hence a Lagrange interpolation polynomial representation of the function $x_j^{n_j}$, and this completes the proof. □

Corollary 4.1. *The relations (4.11) define a one-to-one transformation between the sets of variables (λ, w, u, v) and (x, z, ξ). In particular, linear combinations of the equations (4.11) produce the relations (4.15).*

Proof. Given (x, z, ξ), the equations (4.11) evidently yield a unique value for (λ, w, u, v). Conversely, given (λ, w, u, v), we have seen by the proof of Theorem 4.1 that there exists a solution (x, z, ξ) satisfying (4.11) as given by (4.15), where

$z_j \equiv x_j \ \forall\ j$. To show this is unique, let us demonstrate that particular linear combinations of the equations (4.11) produce the relations (4.15).

Consider any z_J variable, where J records the indices (including repetitions) of the x-variables whose products compose z in (4.8). (When $J \equiv \{j\}$, we have $z_J \equiv x_j$.) First of all, note that by (4.15), we have

$$z_J = \left[\prod_{j \in J}\left(\sum_k \theta_{jk}\lambda_{jk}\right)\right]_{\ell \text{ under (4.4, 4.5)}} = \left[\prod_{j \in J'}\left(\sum_k \theta_{jk}^{n_j}\lambda_{jk}\right)\right]_{\ell \text{ under (4.5)}} \tag{4.18}$$

where J' is the set of *distinct* indices in J, with the index $j \in J'$ appearing n_j times in J, where $1 \le n_j \le k_j - 1$ by virtue of having used (4.7) in the IP-RLT(d) process. Now, the last expression within $[\cdot]$ in (4.18) describes, in expanded form, a weighted sum of products of λ-variables, where each term that is comprised of a product of more than one λ-variable involves distinct indices j, and therefore has a corresponding representative variable w . Composing the weighted linear combination of the equations (4.11a, b) corresponding to this expression, and using the fact that $\alpha[\psi]_\ell + \beta[\varphi]_\ell \equiv [\alpha\psi + \beta\varphi]_\ell$ for any scalars α, β and nonlinear expressions ψ and φ, the right-hand side of this linear combination of (4.11) (denoted $(\text{RHS})_{\text{LC}}$) will be given by (upon recomposing (4.18))

$$(\text{RHS})_{\text{LC}} = \left[\prod_{j \in J'}\left(\sum_k \theta_{jk}^{n_j} L_{jk}\right)\right]_{\ell \text{ under (4.8)}} \tag{4.19}$$

Applying (4.17) of Lemma 4.2, this yields

$$(\text{RHS})_{\text{LC}} = \left[\prod_{j \in J'} x_j^{n_j}\right]_{\ell \text{ under (4.8)}} \equiv z_J \text{ as required.}$$

In an identical manner, we can show that a particular linear combination of (4.11) as prompted by (4.15b) would produce each ξ-variable on the right-hand side of the resulting constraint, and this completes the proof. □

Remark 4.5. Note that in an actual implementation in which the transformation (4.2) is applied to X, we would derive the set (4.3) but with the x-variables substituted out using (4.3b). The SSRLT(d) process would then apply R(a), R(c), R(d), R(e), and R(f) along with the linearization (4.5), except that there would be no u-variables present. Note that in applying R(d), we could drop the factors that contain the GUB expressions $\left(\sum_k \lambda_{jk} - 1\right)$, but instead, multiply the individual GUB constraints in (4.3c) by factors composed of y_t times some r variables λ_{jk} corresponding to distinct indices j, for $r = 0, 1, \ldots, d, \forall\ t$. The dropped constraints can then be composed as surrogates of the resulting relaxation constraints. Comparing this SSRLT(d) process with the IP-RLT(d) process, note that in essence, SSRLT(d) involves carrying the GUB constraints (4.3c) and generating RLT constraints by multiplying these with factors of the type in R(c) and also by y_t times these factors $\forall t$. In contrast, since the LIP sum $\sum_k L_{jk}$ *algebraically* equals one explicitly, IP-RLT(d) does not need to carry these constraints or generate RLT product-constraints off them. This results in a savings in problem size. On the other hand, the nonnegativity restrictions in R(e) of SSRLT(d) are simple variable nonnegativity constraints, while those in R′(c) of IP-RLT(d) involve several linearization variables in a structural relationship. Likewise, in the variable definitions (4.5) versus (4.8), whereas SSRLT(d) defines specific variables for each distinct product combination of λ_{jk} variables, when IP-RLT(d) considers a corresponding product of LIPs L_{jk}, it recognizes each L_{jk} as a function of the same variable x_j for each $k = 1, \ldots, k_j$, and hence defines RLT variables in (4.8) as products of the original x_j variables (including self-products). This can result in a reduction in the number of variables as well.

Remark 4.6. Consider the set $X'(d)$ and suppose that $x_j \in S_j \ \forall\ j$ in some feasible solution. Examining the equivalent representation $X'_+(d)$, the variables λ_{jk} that yield these values of x_j via $x_j = \sum_k \theta_{jk}\lambda_{jk}, \ \forall\ j$, need not necessarily be binary (although there exist corresponding binary feasible solution values for these λ-variables). Accordingly, the variables z and ξ given by (4.15) (see Corollary 4.1) need not necessarily equal the values of the actual products of the variables that they represent as in (4.8). However, there does exist an alternative feasible solution to $X'(d)$ for which the variables z and ξ take on the values as dictated by the product expressions in (4.8). Moreover, in either case, since the original inequalities defining X are implied by $X'(d)$, these are always satisfied. In this sense, any mixed-integer programming problem of the type to

minimize $\{cx + dy : (x, y) \in X\}$ (4.20a)

is equivalent for each $d = 0, 1, \ldots, n$ to

minimize $\{cx + dy : (x, y, z, \xi) \in X'(d), x_j \in S_j \ \forall\ j\}$. (4.20b)

In particular, by Theorem 4.1, this equivalence holds for $d = n$ without the explicit restrictions $x_j \in S_j \; \forall \; j$ in (4.20b), provided we derive an extreme point optimal solution to this resulting problem.

4.5. Illustrative Example

Consider the discrete set

$$X = \{(x, y) : 2x + 2y \leq 5, \; 0 \leq y \leq 1, \; x = 0, 1, \text{ or } 2\}. \tag{4.21}$$

Let us apply the procedure IP-RLT(1) at level $d = 1$. Since $n = 1$, we ought to produce the convex hull representation as illustrated in Figure 4.1. Note that the LIPs given by (4.6) are as follows, where we have dropped the subscript j since $n = 1$ in our example.

$$L_1 = \frac{(x-1)(x-2)}{2}, \; L_2 = \frac{x(x-2)}{-1}, \text{ and } L_3 = \frac{x(x-1)}{2}. \tag{4.22}$$

By R′(a), we need to multiply $2x + 2y \leq 5$ by L_1, L_2, and L_3, using (4.7), which says that $xL_1 \equiv 0$, $xL_2 \equiv L_2$, and $xL_3 \equiv 2L_3$. This gives constraints (4.23a, b, c) stated below, upon linearizing via (4.8). Note that the sum of these constraints produces the original inequality $2x + 2y \leq 5$, thereby implying it. Likewise, the linearized product of $y \leq 1$ with L_1, L_2, and L_3 produces (4.23d, e, f), where the sum of the latter implies the former. By R′(b), the linearized product of $y \geq 0$ with L_1, L_2, and L_3 produces (4.23g, h, i), and again, the sum of these RLT constraints implies the original inequality $y \geq 0$. Finally, R′(c) produces (4.23j, k, ℓ) upon linearization.

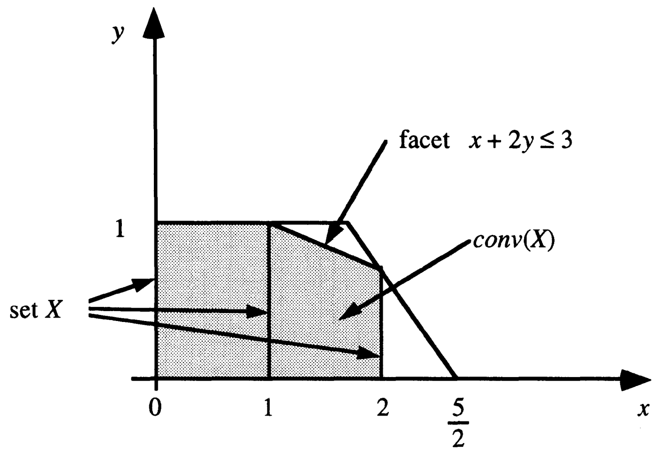


Figure 4.1. Discrete set and its convex hull representation.

$$
\boxed{(2x+2y)\le 5}\;\left\{\begin{array}{llllllll}
*L_1 \Rightarrow & \frac{-15}{2}x & -2y & +\frac{5}{2}z & +3\xi_1 & -\xi_2 & +5\ge 0 & (4.23a)\\
*L_2 \Rightarrow & 6x & & -3z & -4\xi_1 & +2\xi_2 & \ge 0 & (4.23b)\\
*L_3 \Rightarrow & \frac{-1}{2}x & & +\frac{1}{2}z & +\xi_1 & -\xi_2 & \ge 0 & (4.23c)
\end{array}\right.
$$

$$
\boxed{y\le 1}\;\left\{\begin{array}{llllllll}
*L_1 \Rightarrow & \frac{-3}{2}x & -y & +\frac{1}{2}z & +\frac{3}{2}\xi_1 & -\frac{1}{2}\xi_2 & +1\ge 0 & (4.23d)\\
*L_2 \Rightarrow & 2x & & -z & -2\xi_1 & +\xi_2 & \ge 0 & (4.23e)\\
*L_3 \Rightarrow & \frac{-1}{2}x & & +\frac{1}{2}z & +\frac{1}{2}\xi_1 & -\frac{1}{2}\xi_2 & \ge 0 & (4.23f)
\end{array}\right.
$$

$$
\boxed{y\ge 0}\;\left\{\begin{array}{llllllll}
*L_1 \Rightarrow & & y & & \frac{-3}{2}\xi_1 & +\frac{1}{2}\xi_2 & \ge 0 & (4.23g)\\
*L_2 \Rightarrow & & & & 2\xi_1 & -\xi_2 & \ge 0 & (4.23h)\\
*L_3 \Rightarrow & & & & -\frac{1}{2}\xi_1 & +\frac{1}{2}\xi_2 & \ge 0 & (4.23i)
\end{array}\right.
$$

$$
\boxed{\text{Nonnegativity of}}\left\{\begin{array}{llll}
L_1 \Rightarrow & \dfrac{-3}{2}x & +\dfrac{1}{2}z & +1 \geq 0 \quad (4.23j) \\
L_2 \Rightarrow & 2x & -z & \geq 0 \quad (4.23k) \\
L_3 \Rightarrow & \dfrac{-x}{2} & +\dfrac{1}{2}z & \geq 0. (4.23\ell)
\end{array}\right.
$$

Aside from the facets $0 \leq y \leq 1$ of *conv*(X) (see Figure 4.1) which are included in the original constraints, the facet $x \geq 0$ is implied by the surrogate (1/2)(4.23k) + (4.23 ℓ) which yields $x/2 \geq 0$, the facet $x \leq 2$ is obtained as the surrogate 2(4.23j) + (4.23k), and the key facet $x + 2y \leq 3$ is given by the surrogate

(4.23c) + 2(4.23d) + 2(4.23e) + (4.23j).

Hence, *conv*(X) is implied by (4.23), as expected.

4.6. Translating Valid Inequalities from Zero-One to General Discrete Spaces

For the sake of illustration, consider a knapsack constraint in the discrete variables $j \in N \equiv \{1, ..., n\}$ given by

$$\sum_{j \in N} a_j x_j \leq a_0. \tag{4.24}$$

Using the transformation (4.2), we can write this constraint as

$$\sum_{j \in N} \sum_{k=1}^{k_j} (a_j \theta_{jk}) \lambda_{jk} \leq a_0 \tag{4.25a}$$

where,

$$\sum_{k=1}^{k_j} \lambda_{jk} = 1 \qquad \forall\ j \in N,\ \lambda_{jk} \in \{0, 1\}\ \forall\ j, k. \tag{4.25b}$$

This representation permits us to derive a variety of strong valid inequalities that are available for GUB constrained 0-1 knapsack polytopes (see Wolsey (1990), Gu *et al.*, and Sherali and Lee (1995), for example). Suppose that any such valid inequality is obtained as

$$\sum_j \sum_k \alpha_{jk} \lambda_{jk} \leq \alpha_0. \tag{4.26}$$

Noting the proportionality structure of the coefficients in (4.25a), in the special case that the coefficients in (4.26) satisfy

$$\alpha_{jk} \equiv \alpha_j \theta_{jk} \text{ for } k = 1, \ldots, k_j, \ \forall\ j \in N \tag{4.27a}$$

for some α_j, $j \in N$, the constraint (4.26) would translate to the valid inequality

$$\sum_{j \in N} \alpha_j x_j \leq \alpha_0 \tag{4.27b}$$

in the space of the x-variables. However, in general, we would need to apply the inverse transformation (4.6) to (4.26) in order to derive a valid inequality in the x-variable space which, in this case, would be a *nonlinear* inequality of the form

$$\sum_{j \in N} \gamma_j x_j + \sum_{j \in N} \left[\sum_{r=2}^{k_j - 1} \delta_{jr} x_j^r \right] \leq \gamma_0. \tag{4.28}$$

There are two fundamental ways in which (4.28) can be treated. In the case when other constraints of IP-RLT(d) are being generated, (4.28) can be likewise linearized by substituting the appropriate $z_.$ variables as in (4.8a) in lieu of $x_j^r\ \forall\ r, j$. An alternative would be to project out the nonlinear terms or bound them. For example, for each $j \in N$, we can examine the values taken on by $\sum_r \delta_{jr} x_j^r$ when $x_j = \theta_{jk}$ for $k = 1, \ldots, k_j$, and accordingly, derive affine supports $\psi_j x_j + \psi_{0j}$ such that

$$\psi_j x_j + \psi_{0j} \leq \sum_{r=2}^{k_j - 1} \delta_{jr} x_j^r \quad \forall\ x_j = \theta_{jk},\ k = 1, \ldots, k_j, \text{ for each } j \in N. \tag{4.29}$$

Then, exploiting the fact that x_j must lie in the discrete set S_j $\forall\ j \in N$, we can use (4.29) in (4.28) to derive a valid inequality

$$\sum_{j \in N} (\gamma_j + \psi_j) x_j \leq (\gamma_0 - \sum_{j \in N} \psi_{0j}). \tag{4.30}$$

Note that we can also derive more than one affine support of the type (4.29) for use in (4.28), and in this case, we could replace (4.28) by the inequality

$$\sum_{j \in N} \gamma_j x_j + \sum_{j \in N} \phi_j \leq \gamma_0 \tag{4.31}$$

where the variable ϕ_j is further constrained to be greater than or equal to each such affine support generated for each $j \in N$.

In any case, this concept opens the possibility of deriving a myriad of valid inequalities for general discrete structured problems, by exploiting the rich library of techniques for generating such constraints that are available for (GUB) zero-one programming problems. We encourage the reader to explore such constructs.

5

GENERATING VALID INEQUALITIES AND FACETS USING RLT

Thus far, we have presented a hierarchy of relaxations leading up to the convex hull representation for zero-one mixed-integer programming problems, and have developed extensions of this hierarchy to accommodate inherent special structures as well as to handle general discrete (as opposed to simply 0-1) variables. A key advantage of this development that we wish to discuss in the present chapter is that the RLT produces an algebraically explicit convex hull representation at the highest level. While it might not be computationally feasible to actually generate and solve the linear program based on this convex hull representation because of its potentially exponential size, there are other ways to exploit this information or facility to advantage as we shall presently see.

One obvious tactic might be to identify a subset of variables along with either original or implied constraints involving these variables for which such a convex hull representation (or some higher order relaxation) would be manageable. Another approach might be to exploit any special structures in order to derive a simplified algebraic form of the convex hull representation for which the polyhedral cone that identifies all defining facets itself possesses a special structure. Exploiting this structure, it might then be possible to generate specific extreme directions of this cone, and hence identify certain (classes of) facets that could be used to strengthen the original problem formulation. Perhaps, this algebraic simplification could yield more compact, specially structured, representations of each level in the hierarchy, hence enabling an actual implementation of a full or partial application of RLT at some intermediate level in order to compose tight relaxations. Each of these ideas are explored in the present chapter.

We begin in Section 5.1 with the Boolean Quadric Polytope and we derive a characterization for all its defining facets via the extreme directions of a specially structured polyhedral cone. We next demonstrate how various classes of facets are revealed via the enumeration of different types of extreme directions, some derived by examining lower-dimensional projected restriction of this polyhedral cone. This type of an analysis is also illustrated in Section 5.2 for the very useful class of generalized upper bounding (GUB) constrained knapsack polytopes. Finally, in Section 5.3, we consider another important class of combinatorial optimization problems namely the set partitioning problem, and we describe simplified forms of the RLT relaxation obtained by exploiting its special structures. We then exhibit how various types of strong valid inequalities are all automatically captured within its first and second RLT relaxation levels. The polyhedral structure of many other specially structured combinatorial optimization problems can be studied using such constructs, and tight relaxations and strong valid inequalities can be generated for such problems. As recently shown by Sherali (1996), even convex envelopes of some special multilinear nonconvex functions can be generated using this approach. We hope that the discussion in this chapter motivates the readers to pursue investigations of this type.

5.1. Convex Hull Characterization and Facets for the Quadric Boolean Polytope

Consider the unconstrained pseudo-Boolean quadratic zero-one problem in n variables:

$$\textbf{QP:}\ \text{Maximize}\left\{\sum_{j\in N} q_j x_j + \sum_{1\le i<j\le n} q_{ij} x_i x_j :\ x_j \in (0,1),\ \ \forall\ j \in N\right\},$$

where all the data is rational, and where $N \equiv \{1, \ldots, n\}$. Problem QP can be used for the analysis of many combinatorial optimization problems. As shown by Hansen (1979), problems such as vertex packing, maximum cut, quadratic assignment, and set partitioning, to name a few, can be reformulated as problem QP. It is also well known that the maximization of any pseudo-Boolean polynomial function can be reduced to maximizing a quadratic function.

A typical approach for solving Problem QP is to linearize the quadratic terms $x_i x_j$ by introducing new variables w_{ij} that are zero-one valued and satisfy the following standard set of constraints:

$$w_{ij} - x_i \le 0 \quad \forall\ 1 \le i < j \le n, \tag{5.1a}$$

$$w_{ij} - x_j \le 0 \quad \forall\ 1 \le i < j \le n, \tag{5.1b}$$

$$-w_{ij} + x_i + x_j \le 1 \quad \forall\ 1 \le i < j \le n, \tag{5.1c}$$

$$x_i \in \{0,1\} \quad \forall\ i \in N, \quad w_{ij} \geq 0 \quad \forall\ 1 \leq i < j \leq n. \tag{5.1d}$$

Then, problem QP becomes the following linear zero-one (mixed) integer program.

$$\text{Maximize } \left\{ \sum_{j \in N} q_j x_j + \sum_{1 \leq i < j \leq n} q_{ij} w_{ij} : \ (x, w) \text{ satisfies } (5.1\text{a})\text{-}(5.1\text{d}) \right\}.$$

Accordingly, denoting the convex hull by "*conv*," define

$$\text{QP}^n = conv\{(x, w) \in R^{n_0} : \ (x, w) \text{ satisfies } (5.1\text{a})\text{-}(5.1\text{d})\} \tag{5.2}$$

as the *Boolean quadric polytope*, where $n_0 \equiv \frac{1}{2} n(n+1)$. It is easy to verify that QP^n is full dimensional.

Note that the structure defined by (5.1) would arise in the linearization of quadratic terms $x_i x_j$ involving binary variables that might appear in any quadratically constrained quadratic program as well. Moreover, as observed by Simone (1989), QP^n can be equivalently transformed into the cut polytope defined by Barahona and Mahjoub (1986), and vice versa. Hence, the study of the polyhedral structure of QP^n is very important, and has been widely explored in the literature (see Sherali *et al.*, 1995, on which this section is based, and the references cited therein).

Now, let us derive the higher dimensional convex hull relaxation X_n for QP^n, along with its projection X_{Pn} as in Chapter 2. Note that in QP^n, the x-variables are binary valued, and the w-variables are continuous (these were referred to as "y"-variables in Chapter 2; however, since these variables represent the product terms $x_i x_j$, we will continue to use the present notation). To generate X_n, we need to multiply each inequality in (5.1) by each factor $F_n(J, \bar{J})$ $\forall J \subseteq N$. As before, defining $f_n(J, \bar{J})$ $\forall J \subseteq N$ and $f_n^{ij}(J, \bar{J})$ as the linearizations under RLT of $F_n(J, \bar{J})$ and $w_{ij} F_n(J, \bar{J})$, respectively, this yields the following. The products of $F_n(J, \bar{J})$ with (5.1a, b) give

$f_n(J, \bar{J}) \geq f_n^{ij}(J, \bar{J})$ if $\{i, j\} \subseteq J$ and $f_n^{ij}(J, \bar{J}) \leq 0$ otherwise

$\forall J \subseteq N, \ \forall i < j.$

Similarly, the products of $F_n(J, \bar{J})$ with (5.1c) yield $F_n(J, \bar{J}) \leq f_n^{ij}(J, \bar{J})$ if $\{i, j\} \in J$, and essentially, $f_n^{ij}(J, \bar{J}) \geq 0$ otherwise. This together with the nonnegativity of $f_n(J, \bar{J})$ and $f_n^{ij}(J, \bar{J})$ yields

$$f_n^{ij}(J, \bar{J}) \equiv \begin{cases} f_n(J, \bar{J}) & \forall i < j \text{ for which } \{i, j\} \subseteq J \\ 0 \text{ otherwise} \end{cases} \quad \forall J \subseteq N. \tag{5.3}$$

Using (5.3) in Equation (2.26), (and noting that (5.3) is a simplification of (2.26c) and (2.26e)), we obtain

$$X_{Pn} = (x, w): \sum_{J \subseteq N: j \in J} U_J^0 = x_j \quad \forall j = 1, \ldots, n \tag{5.4a}$$

$$\sum_{J \subseteq N: i, j \in J} U_J^0 = w_{ij} \quad \forall i < j \tag{5.4b}$$

$$-\sum_{J \subseteq N} U_J^0 = -1 \tag{5.4c}$$

$$U_J^0 \geq 0 \quad \forall J \subseteq N\}. \tag{5.4d}$$

Hence, associating dual multipliers λ_j, λ_{ij} for $i < j$, and λ_0 with the constraints (5.4a), (5.4b), and (5.4c), respectively, by linear programming duality or Farkas' Lemma (as in Equation (2.29)), we obtain

$$X_{Pn} = \{(x, w): \sum_{j \in N} \lambda_j x_j + \sum_{i<j \in N} \lambda_{ij} w_{ij} \le \lambda_0\} \equiv QP^n \tag{5.5}$$

where $\lambda \equiv (\lambda_0, \lambda_j \text{ for } j \in N, \lambda_{ij} \text{ for } i < j \in N)$ belongs to Λ, and where

$$\Lambda \equiv \left\{ \lambda: \sum_{j \in J} \lambda_j + \sum_{i<j \in J} \lambda_{ij} \le \lambda_0 \quad \forall\, J \subseteq N \right\}. \tag{5.6}$$

It is easy to verify that Λ is a pointed polyhedral cone having its vertex at the origin. Moreover, by Chapter 2, the inequality in (5.5) is facet-defining for $\mathrm{QP}^n \equiv X_{Pn}$ if and only if λ is an extreme direction of Λ. Hence, we will now focus on the set Λ in order to identify classes of facets for QP^n.

By Theorem 3 (Lifting Theorem) of Padberg (1989), any facet of $\mathrm{QP}^{n'}$ $(n' < n)$ is a facet for QP^n, and conversely, any facet of QP^n that has nonzero coefficients for variables corresponding to some $n' < n$ indices, is also a facet for the corresponding polytope $\mathrm{QP}^{n'}$. Hence, inductively, it is sufficient to study facets having all the indices appearing among its nonzero coefficients. Also, note that all the facets of QP^n for $n \le 3$ are trivially identified by Padberg (1989). Therefore, we henceforth assume that $n \ge 4$. Furthermore, by Theorem 6 (Symmetry Theorem) of Padberg (1989), the *vertex figure* of any vertex $(\bar{x}, \bar{w})$ of QP^n is the same as that of any other vertex, say, the origin; that is, the facets binding at any vertex $(\bar{x}, \bar{w})$ are obtainable from those binding at the origin through a simple transformation. Hence, it is sufficient to characterize only the facets given by (5.5) that are binding at the origin, and as per the foregoing remark, that additionally have all n indices represented therein.

Accordingly, we are henceforth interested in identifying facets of the type

$$\sum_{j \in N} \lambda_j x_j + \sum_{1 \le i < j \le n} \lambda_{ij} w_{ij} \le 0 \tag{5.7}$$

that additionally have all the indices in N accounted for among the nonzero coefficients, where λ is an extreme direction of the set Λ_0 given below:

$$\Lambda_0 \equiv \left\{ \lambda: \sum_{j \in J} \lambda_j + \sum_{i < j \in J} \lambda_{ij} \le 0 \quad \forall\ J \subseteq N \right\}. \tag{5.8}$$

Now, consider a facet of the type (5.7) as specified below for the case of $(n-1)$ variables x_j, $j \in N' \equiv \{1, \ldots, n-1\}$:

$$\sum_{j \in N'} \bar{\lambda}_j x_j + \sum_{i < j \in N'} \bar{\lambda}_{ij} w_{ij} \le 0. \tag{5.9}$$

Similar to (5.8), note that $\bar{\lambda}(N') \equiv (\bar{\lambda}_j \text{ for } j \in N', \bar{\lambda}_{ij} \text{ for } i < j \in N')$ is an extreme direction of the following polyhedral cone:

$$\Lambda_0' \equiv \left\{ \lambda: \sum_{j \in J} \lambda_j + \sum_{i < j \in J} \lambda_{ij} \le 0 \quad \forall\ J \subseteq N' \right\}. \tag{5.10}$$

Now, let us consider a lifting of the inequality (5.9) into a facet for QP^n.

Proposition 5.1. *Suppose that the inequality given by (5.9) is a facet for* QP^{n-1}. Then, the inequality

$$\bar{\lambda}_n x_n + \sum_{j \in N'} \bar{\lambda}_{jn} w_{jn} + \sum_{j \in N'} \bar{\lambda}_j x_j + \sum_{i<j \in N'} \bar{\lambda}_{ij} w_{ij} \le 0 \tag{5.11}$$

is a facet for QP^n *if and only if* $\bar{\lambda}(n) \equiv (\bar{\lambda}_n, \bar{\lambda}_{jn}$ *for* $j \in N')$ *is a vertex of the following polyhedron:*

$$\left\{ \lambda(n):\ \lambda_n + \sum_{j \in J} \lambda_{jn} \le - \sum_{j \in J} \bar{\lambda}_j - \sum_{i<j \in J} \bar{\lambda}_{ij} \quad \forall\ J \subseteq N' \right\}. \tag{5.12}$$

Proof. Let us partition the constraints of Λ_0 in (5.8) as follows:

$$N'\text{-Block:}\quad \sum_{j \in J} \lambda_j + \sum_{i<j \in J} \lambda_{ij} \le 0 \quad \forall\ J \subseteq N' \tag{5.13a}$$

$$n\text{-Block:}\quad \lambda_n + \sum_{j \in J} \lambda_{jn} \le - \sum_{j \in J} \lambda_j - \sum_{i<j \in J} \lambda_{ij} \quad \forall\ J \subseteq N'. \tag{5.13b}$$

From (5.9) and (5.10), at the solution $\bar{\lambda}(N')$, we have that there are some $\frac{1}{2}n(n-1) - 1$ linearly independent hyperplanes binding from (5.13a). In fact, these equations yield values of $\bar{\lambda}(N')$ uniquely when some variable in N' is fixed at the corresponding value in $\bar{\lambda}(N')$ by way of normalization. Hence, if $\bar{\lambda}(n) \equiv (\bar{\lambda}_n, \bar{\lambda}_{jn}$ for $j \in N')$ is an extreme point of (5.12), then it is uniquely determined by some n linearly independent binding inequalities defining this polyhedron. Examining (5.12) and (5.13b), this means that the corresponding vertex $\bar{\lambda}(N) \equiv (\bar{\lambda}_j$ for $j \in N$, $\bar{\lambda}_{ij}$ for $i < j \in N)$ is feasible to Λ_0 of (5.8), and has $[\frac{1}{2}n(n-1) - 1] + n = \frac{1}{2}n(n+1) - 1$ linearly independent defining hyperplanes from (5.13) binding at this solution. Hence, $\bar{\lambda}(N)$ is an extreme direction of Λ_0, and is therefore a facet of QP^n. The converse argument is similar, and this completes the proof. □

In the light of Proposition 5.1, we are interested in characterizing vertices of (5.12), where the right-hand sides in (5.12) have values given by any known facet for the $(n-1)$ variable case. Note that this is like a *simultaneous lifting* process (see Nemhauser and Wolsey, 1988, for example), since the coefficients $\lambda(n) \equiv (\lambda_n, \lambda_{jn}$ for $j \in N')$ are being *simultaneously* determined, as opposed to being *sequentially* determined.

Observe that we have set the stage here to generate any class of facets for QP^n that can be obtained via a simultaneous lifting of some other class of facet written for QP^{n-1}, by enumerating the vertices of (5.12). In effect, we have reduced the task of examining, in general, all the extreme rays of Λ defined in (5.6) to that of exploring the extreme rays of a projected lower dimensional subset of Λ.

As an example, consider the following *homogeneous product-form facets* for QP^n (see Boros and Hammer, 1990, and Deza and Laurent, 1988):

$$-\sum_{j\in J} x_j - \sum_{i<j\in J} w_{ij} - \sum_{i<j\in \bar{J}} w_{ij} + \sum_{(i,j)\in(J:\bar{J})} w_{ij} \leq 0$$

$$\forall\ J \subseteq N,\ |J| = 1,\ldots,n-2 \qquad (5.14)$$

where $\bar{J} \equiv N - J$ and $(J:\bar{J}) \equiv \{(i,j)\colon i<j$, and either $i \in J,\ j \in \bar{J}$, or $i \in \bar{J}$ and $j \in J\}$.

In N'-space, for some $J_1 \subseteq N'$, $|J_1| \in \{1, \ldots, n-3\}$, $\bar{J}_1 \equiv N' - J_1$, this type of facet is as specified below:

$$-\sum_{j \in J_1} x_j - \sum_{i<j \in J_1} w_{ij} - \sum_{i<j \in \bar{J}_1} w_{ij} + \sum_{(i,j) \in (J_1 : \bar{J}_1)} w_{ij} \leq 0 \ . \tag{5.15}$$

Let $\bar{\lambda}(N')$ denote the coefficients of (5.15). Accordingly, denote the right-hand side of (5.12) by b_J, for any $J \subseteq N'$, when $\bar{\lambda}(N')$ is given by (5.15). Then, for any $J \subseteq N'$, $|J \cap J_1| = p$, $|J \cap \bar{J}_1| = q$, we have that

$$b_J \equiv p + \frac{1}{2} p(p-1) + \frac{1}{2} q(q-1) - pq = \frac{1}{2}(p-q)(p-q+1).$$

This generates the constraints of (5.12) as follows:

$$\lambda_n + \sum_{j \in J} \lambda_{jn} \leq \frac{1}{2}(p-q)(p-q+1) \qquad \forall \ \bar{J} \subseteq N',$$

$$\text{where } p \equiv |J \cap J_1| \text{ and } q \equiv |J \cap \bar{J}_1| \ . \tag{5.16}$$

Proposition 5.2. *The following are three vertices of (5.16):*

(i) $\lambda_n = 0$, $\lambda_{jn} = 0$, $\forall \ j \in N'$,

(ii) $\lambda_n = -1$, $\lambda_{jn} = -1$, *for* $j \in J_1$, $\lambda_{jn} = 1$ *for* $j \in \bar{J}_1$ *and*

(iii) $\lambda_n = 0$, $\lambda_{jn} = 1$ *for* $j \in J_1$, $\lambda_{jn} = -1$ *for* $j \in \bar{J}_1$.

In particular, (i) *corresponds to the facet of* QP^{n-1} *itself, which is a facet for* QP^{n}, (ii) *corresponds to* $J = J_1 \cup \{n\}$ *and* $\bar{J} = \bar{J}_1$ *in the product-form facet (5.14), and* (iii) *corresponds to* $J = J_1$ *and* $\bar{J} = \bar{J}_1 \cup \{n\}$ *in (5.14).*

Proof. First of all, note that the given values of $(\lambda_n, \lambda_{jn}$ for $j \in N')$ in (i)-(iii), in combination with (5.15), yield inequalities of type (5.14), and therefore correspond to facets of QP^n. Consequently, by Proposition 5.1, the given three sets of values of $(\lambda_n, \lambda_{jn}$ for $j \in N')$ must be vertices of (5.16). The other assertions of the proposition are readily evident, and this completes the proof. $\square$

The question that can be raised here is whether Proposition 5.2 defines all the vertices of (5.16), and so, by lifting (5.15) from the N'-space to the N-space, we produce only facets of the same type (5.14). By examining the resulting facets obtained for $n \leq 4$ with the known ones for this case as given by Deza and Laurent (1988), this is true for $n \leq 4$. However, as the following example shows, this is false for $n \geq 5$. Indeed, for $n \geq 5$, not all facets are of the type (5.14), and in fact, as we show below, vertices of (5.16) themselves can produce facets different from this class.

Example 5.1. Let $n = 5$, $J_1 = \{1, 2\}$, $\bar{J}_1 = \{3, 4\}$, $\bar{\lambda}_5 = -1$, $\bar{\lambda}_{j5} = 2$ $\forall\ j \in \bar{J}_1$, $\bar{\lambda}_{j5} = -2$ $\forall\ j \in J_1$. Let us verify that this vector $\bar{\lambda}(n)$ is a vertex of (5.16). To show feasibility, we must have $-1 + 2p - 2q \leq \frac{1}{2}(p-q)(p-q+1)$, i.e., $(p-q)^2 - 3(p-q) + 2 \geq 0$ for $p = 0, 1, 2$, and $q = 0, 1, 2$. This expression is convex in $(p-q)$ and is minimized when $(p-q) = \frac{3}{2}$. Moreover, it is zero when

$(p - q) = 1$ or 2, and so by convexity, it is nonnegative for all integer values of $(p - q)$. Furthermore, the following five constraints from (5.16) are linearly independent and are binding at $\bar{\lambda}(n)$, yielding $\bar{\lambda}(n)$ as the unique solution:

$$\lambda_5 + \lambda_{15} = 1,\ \lambda_5 + \lambda_{25} = 1,\ \lambda_5 + \lambda_{15} + \lambda_{25} = 3,$$

$$\lambda_5 + \lambda_{15} + \lambda_{25} + \lambda_{35} = 1,\ \lambda_5 + \lambda_{15} + \lambda_{25} + \lambda_{45} = 1.$$

Consequently, $\bar{\lambda}(n)$ is a vertex of (5.16). The corresponding facet is

$$[-x_1 - x_2 - w_{12} - w_{34} + w_{13} + w_{14} + w_{23} + w_{24}]$$
$$+ [-x_5 + 2w_{15} + 2w_{25} - 2w_{35} - 2w_{45}] \leq 0.$$

This facet is not of the type (5.14), but it has been obtained by lifting a facet of the type (5.14) from the N'-space. In fact, as we shall show, the above inequality is a member of an entire class of facets obtainable in this fashion. □

Toward this end, consider (5.16), and suppose that for some θ, we restrict

$$\lambda_{jn} = \theta \quad \forall\ j \in J_1 \quad \text{and} \quad \lambda_{jn} = -\theta \quad \forall\ j \in \bar{J}_1. \tag{5.17}$$

Then, system (5.16) becomes

$$\lambda_n + \theta(p - q) \leq \frac{1}{2}(p - q)(p - q + 1) \quad \forall\ p = 0, 1, \ldots, P,$$
$$q = 0, 1, \ldots, n - 1 - P,$$

where $P = |J_1|$. This is equivalent to the system

$$\lambda_n + \theta r \le \frac{1}{2} r(r+1) \qquad \forall\ r = P+1-n, \ldots, P. \tag{5.18}$$

Proposition 5.3. *The vertices* (λ_n, θ) *of the two-dimensional polyhedron (5.18) are all of the type*

$$\lambda_n = -\frac{1}{2} R(R-1) \textit{ and } \theta = R, \textit{ for } R = P+2-n, \ldots, P. \tag{5.19}$$

Proof. Consider any defining inequality in (5.18) for r equal to some R, say. Setting this as an equality gives $\lambda_n = \frac{1}{2} R(R+1) - \theta R$. By examining the end points of the segment of this plane (line) that are feasible to (5.18), we can obtain vertices of (5.18) that lie on this plane. Repeating this for all planes, will give all the vertices of (5.18). Accordingly, note that points in the above plane are feasible provided $\frac{1}{2} R(R+1) - \theta R + \theta r \le \frac{1}{2} r(r+1)$, i.e., $\theta(R-r) \ge \frac{1}{2}(R-r) \quad (R+r+1)$ $\forall\ r = P+1-n, \ldots, P.$

In turn, this is equivalent to the following:

$$\theta \ge \frac{1}{2}(R+r+1)\ \forall\ r = P+1-n, \ldots, R-1 \text{ and}$$

$$\theta \le \frac{1}{2}(R+r+1)\ \forall\ r = R+1, \ldots, P.$$

This implies that

$\theta \ge R$ if $R \ge P+2-n$ and $\theta \le (R+1)$ if $R \le P-1$.

Hence, tracing end points as R varies from $P + 1 - n$ to P, we get vertices of (5.18) as $\theta = R$ and $\lambda_n = \frac{1}{2}R(R+1) - R^2 = -\frac{1}{2}R(R-1)$, for $R = P + 2 - n, \ldots, P$. This completes the proof. $\square$

The following proposition verifies that the vertices (5.19) of (5.18) also correspond to vertices of (5.16) via (5.17).

Proposition 5.4. *The vector* $\bar{\lambda}(n) \equiv \{\bar{\lambda}_n = -\frac{1}{2}R(R-1),\ \bar{\lambda}_{jn} = R \ \ \forall\ j \in J_1,$ $\bar{\lambda}_{jn} = -R \ \ \forall\ j \in \bar{J}_1\}$, *for each* $R = P + 2 - n, \ldots, P$, *where* $P = |J_1|$, *defines a vertex of (5.16).*

Proof. Given any $R \in \{P + 2 - n, \ldots, P\}$, since $\theta = R$ and $\lambda_n = -\frac{1}{2}R(R-1)$ are feasible to (5.18) by Proposition 5.3, we have from (5.17) that

$$\{\lambda_n = -\frac{1}{2}R(R-1),\ \lambda_{jn} = R \ \ \forall\ j \in J_1,\ \lambda_{jn} = -R \ \ \forall\ j \in \bar{J}_1\} \tag{5.20}$$

is feasible to (5.16) for all $R = P + 2 - n, \ldots, P$, where $|J_1| \equiv P$. We need to show that (5.20) is a vertex of (5.16). For this solution, the left-hand side of any constraint in (5.16) is given by $-\frac{1}{2}R(R-1) + R(p-q)$ and the right-hand side is $\frac{1}{2}(p-q)(p-q+1)$. Hence, the constraint is binding whenever these are equal. This happens whenever $(p - q - R)(p - q - R + 1) = 0$, that is, $(p-q) = R$ or $(p-q) = R - 1$.

To complete the proof, it is sufficient to show that the set of binding constraints yield (5.20) as the unique solution. Toward this end, let $s \in J_1$ and $t \in \bar{J}_1$ be arbitrarily selected. (Note that $|J_1| \geq 1$ and $|\bar{J}_1| \geq 2$ when $n \geq 4$.) We show below that the system of binding constraints implies that $\lambda_{sn} = -\lambda_{tn}$. Since the choice of s and t is arbitrary, this in turn would mean that the foregoing system implies that $\lambda_{jn} = \theta \;\; \forall\, j \in J_1$, $\lambda_{jn} = -\theta \;\; \forall\, j \in \bar{J}_1$. But this reduces the system (5.16) to the system (5.18). Examining the constraints in (5.18) for $r = (p - q) = R$ and $r = (p - q) = R - 1$, which are binding at the given solution (5.20) as noted above, we get $\lambda_n + \theta R = \frac{1}{2} R(R+1)$ and $\lambda_n + \theta(R-1) = \frac{1}{2} R(R-1)$. These equations uniquely yield $\lambda_n = -\frac{1}{2} R(R-1)$ and $\theta = R$, and so, this would imply that (5.20) is a vertex of (5.16).

Hence, to show that $\lambda_{sn} = -\lambda_{tn}$, consider a set $J \subseteq N'$ such that $|J \cap J_1| = p, |J \cap \bar{J}_1| = q, \{s, t\} \cap J = \emptyset$, and $(p - q) = (R - 1)$ if $R = P$, and $(p - q) = R$ if $R < P$. (Note that if $R = P$, we must have $J = J_1 - \{s\}$, and if $R < P$, then since $R \geq P + 2 - n$ such a J exists for which $p \leq P - 1 = |J_1| - 1$, $q \leq n - 2 - P = |\bar{J}_1| - 1$, and $(p - q) = R$.) Since the constraint for this index set J is binding, as also is the one for $J \cup \{s, t\}$, with both having the same right-hand side value since $(p - q)$ remains the same, we get from the two equations that

$$\lambda_n + \sum_{j \in J} \lambda_{jn} = \lambda_n + \sum_{j \in J} \lambda_{jn} + \lambda_{sn} + \lambda_{tn}.$$

This implies that $\lambda_{sn} = -\lambda_{tn}$, and hence, the proof is complete. □

Consequently, we have the following result that reveals a class of lifted facets for QP^n. (This class overlaps some two other known classes of facets for the corresponding cut polytope.)

Proposition 5.5. *The following inequalities define a class of facets for* QP^n:

$$-\sum_{j\in J_1} x_j - \sum_{i<j\in J_1} w_{ij} - \sum_{i<j\in \bar{J}_1} w_{ij} + \sum_{(i,j)\in(J_1:\bar{J}_1)} w_{ij} - \frac{1}{2}R(R-1)x_n$$

$$+ R\sum_{j\in J_1} w_{jn} - R\sum_{j\in \bar{J}_1} w_{jn} \leq 0,$$

where $J_1 \subseteq N'$, $|J_1| = P \geq 1$, $\bar{J}_1 = N' - J_1$ *with* $|\bar{J}_1| \geq 2$, *and*

$R \in \{P+2-n, \ldots, P\}$.

Proof. Follows from the Propositions 5.3 and 5.4. □

5.2. Convex Hull Characterization and Facets for GUB Constrained Knapsack Polytopes

Consider the generalized upper bounding (GUB) constrained knapsack polytope whose feasible region X is given as follows

$$X \equiv \{x \in (0,1)^n: \sum_{i\in M}\sum_{j\in N_i} a_j x_j \geq b, \quad \sum_{j\in N_i} x_j \leq 1 \quad \forall\ i \in M\} \tag{5.21}$$

where the data is all integer, $N = \{1, \ldots, n\}$, $M = \{1, \ldots, m\}$, and where $\bigcup_{i\in M} N_i \equiv N$, with $N_i \cap N_j = \phi$ for $i, j \in M$, $i \neq j$. Johnson and Padberg (1981) show that any GUB knapsack problem with arbitrarily signed coefficients b and a_j,

$j \in N$, can be equivalently transformed into a form with $b > 0$, and with $0 < a_j \leq b$ $\forall\ j \in N$, and they relate facets of the transformed problem with those of the original problem. Hence, without loss of generality, we will also assume that $b > 0$ and that $0 < a_j \leq b \quad \forall\ j \in N$. Note that if $|N_i| = 1 \quad \forall\ i \in M$, then X represents the constraints of the ordinary knapsack problem.

There are many useful applications in which GUB knapsack constraints arise, such as in capital budgeting problems having a single resource and where the investment opportunities are divided into disjoint subsets, and in menu planning for determining what food items should be selected from various daily menu courses in order to maximize an individual's food preference, subject to a calorie constraint. More importantly, this structure frequently arises as a subset of large-scale real-world 0-1 integer programming problems. As demonstrated in the results of Crowder *et al.* (1983) and Hoffman and Padberg (1991), even a partial knowledge of the polyhedral structure of ordinary and GUB constrained knapsack polytopes can significantly enhance the overall performance of branch-and-cut algorithms. In this spirit, the structure of X has been used to generate classes of valid inequalities for certain scheduling polytopes (see Sherali and Lee, 1990, and Wolsey, 1990), in order to tighten their underlying linear programming relaxations.

Let us begin our analysis by introducing some notation. For $K \subseteq N$, let $M_K = \{i \in M\colon\ j \in N_i, \text{ for some } j \in K\}$. Also, for $k \in N$, we denote $M_{\{k\}}$ simply as M_k. For each $i \in M$, define a *key index* $j(i)$ such that $j(i) \in \arg\max_{j \in N_i} (a_j)$. Similarly, given any $B \subseteq N$, for each $i \in M_B$, define a *key* *index* $j_B(i)$ such that $j_B(i) \in \arg\max_{j \in N_i \cap B}(a_j)$. For $A \subseteq M$, denote $A_+ = \{j(i): \; i \in A\}$. Similarly, for $B \subseteq N$, denote $B_+ = \{j_B(i): \; i \in M_B\}$, and let $B_- = B - B_+$.

Now, let us suppose that for each $k \in N$

$$a_k + \sum_{i \in (M - M_k)} a_{j(i)} \geq b \;, \tag{5.22}$$

for otherwise, $x_k = 0$ in every feasible solution to X. Denoting the convex hull operation by $conv(\cdot)$, let GUBKP $\equiv conv(X)$, and let dim(GUBKP) be the dimension of GUBKP, which is the maximum number of affinely independent points in GUBKP minus one.

Proposition 5.6. *dim(GUBKP)* $= n - |M_0|$, *where* $M_0 = \{i \in M: \sum_{p \in (M-i)} a_{j(p)} < b\}$.

Proof. By the definition of M_0, we must have $\sum_{j \in N_i} x_j = 1$ for each $i \in M_0$. Hence, it follows that dim(GUBKP) $\leq n - |M_0|$. To prove that dim(GUBKP) $= n - |M_0|$, it suffices to show that there exist $n - |M_0| + 1$ affinely independent points in GUBKP.

For each $i \in (M - M_0)$, we construct a set of feasible points in GUBKP as follows. For each $k \in N_i$, construct $x^{(k,i)} \equiv \{x_k = 1, \; x_j = 1$ for $j = j(p)$, $\forall \; p \in (M - i)$, $x_j = 0$ otherwise$\}$, and let $x^{(0,i)} \equiv \{x_j = 1$ for $j = j(p)$, $\forall \; p \in (M - i)$, $x_j = 0$ otherwise$\}$. Similarly, for each $i \in M_0$, we construct a set of feasible points in GUBKP as follows. For each $k \in N_i$, construct $x^{(k,i)} \equiv \{x_k = 1, x_j = 1 \text{ for } j = j(p),\ \forall\ p \in M - i,\ x_j = 0 \text{ otherwise}\}$. Then, the total number of distinct feasible points thus constructed is $n - |M_0| + 1$. Let $\tilde{X}$ be the set of these distinct points x^j, indexed by $j = 1, \ldots, n - |M_0| + 1$. Without loss of generality, let $x^{n-|M_0|+1} \equiv \{x_j = 1 \text{ for } j \in N_+,\ x_j = 0 \text{ otherwise}\}$. Construct a matrix D whose row vectors are $x^j - x^{n-|M_0|+1}$, $j = 1, \ldots, n - |M_0|$. Then the matrix D can be readily seen to possess a block-diagonal structure, with the rows corresponding to each block being linearly independent. Hence, x^j, $j = 1, \ldots, n - |M_0| + 1$, are affinely independent. This completes the proof. □

Now, if dim(GUBKP) is less than n, then by writing each inequality constraint $i \in M_0$ as an equality constraint, and using this equation to eliminate the variable that has the smallest a_j coefficient, we obtain a full dimensional subpolytope of dimension of $n - |M_0|$. Therefore, without loss of generality, we can assume henceforth that GUBKP is a full dimensional polytope.

As in Section 5.1, let us now algebraically derive GUBKP $\equiv conv(X) \equiv X_{Pn}$ using the RLT procedure of Chapter 2, noting that X is defined purely in terms of 0-1 variables in this case. Again, we need to multiply all the problem constraints with the factors $F_n(J, \bar{J})\ \ \forall\ J \subseteq N$. Observe that the linearized product of the factor $F_n(J, \bar{J})$ with the knapsack constraint yields

$$\left[\sum_{j \in J} a_j - b\right] f_n(J, \bar{J}) \geq 0 \quad \forall\ J \subseteq N, \tag{5.23a}$$

while the linearized RLT products with the GUB constraints yield

$$\left[1-|J \cap N_i|\right] f_n(J, \bar{J}) \geq 0 \quad \forall\, i \in M, \quad \forall\, J \subseteq N. \tag{5.23b}$$

In addition to (5.23), we have $f_n(J, \bar{J}) \geq 0 \quad \forall\, J \subseteq N$. This together with (5.23) is simply equivalent to $f_n(J, \bar{J}) \geq 0 \quad \forall\, J \subseteq N$, but $f_n(J, \bar{J}) \equiv 0$ if the multiplier within $[\cdot]$ in (5.23a or b) is negative for any such constraint for this J. Consequently, if we define

$F \equiv \{J \subseteq N: \ \sum_{j \in J} a_j \geq b, \ |J \cap N_i| \leq 1 \quad \forall\, i \in M\}$ and let

$\bar{F} \equiv \{J \subseteq N: \ J \notin F\}$,

we have by (2.26) that

$$\text{GUBKP} \equiv X_{Pn} \equiv conv(X) = \{x: \ x_j = \sum_{J \subseteq N: j \in J} U_J^0 \quad \forall\, j \in N,$$

$$\sum_{J \subseteq N} U_J^0 = 1, \ U_J^0 \geq 0, \ \forall\, J \in F, \ U_J^0 \equiv 0 \quad \forall\, J \in \bar{F}\}. \tag{5.24}$$

Observe that F represents the set of feasible solutions to X, and in effect, (5.24) writes GUBKP as the convex combination of feasible solutions, where U_J^0, $J \in F$, are the convex combination weights. Using the standing projection operation, the set of all x-variables for which there exist corresponding vectors U_J^0 that yield a feasible solution to (5.24) are given by duality or Farkas' Lemma as

$$\text{GUBKP} \equiv \{x: \ \sum_{j \in N} \pi_j^k x_j \geq \pi_0^k,$$

where (π_j^k, π_0^k), $k = 1, \ldots, K$, are the extreme directions of $\Pi\}$ (5.25a)

and where $\Pi \equiv \{(\pi, \pi_0)\colon\ \sum_{j \in J} \pi_j - \pi_0 \geq 0 \quad \forall\ J \in F\}$. (5.25b)

We will now characterize a special class of facets for GUBKP that correspond to a simultaneous lifting (see Nemhauser and Wolsey, 1988) of certain valid inequalities known as minimal GUB covers (as defined in Sherali and Lee, 1995). These are direct extensions of the well known minimal cover inequalities for the ordinary knapsack polytope, and are defined as follows.

We will say that a set $K = \bigcup_{i \in Q} N_i$, for some $Q \subseteq M$, is called a GUB *cover* of X if $\sum_{i \in M_{\overline{K}}} a_{j(i)} \leq b - 1$, where $\overline{K} \equiv N - K$. A GUB cover K is called a *minimal GUB cover* of X if $\sum_{i \in M_{\overline{K}}} a_{j(i)} + \min_{i \in M_K} (a_{j(i)}) \geq b$. Accordingly, we define a *minimal GUB cover inequality* as the valid inequality

$$\sum_{j \in K} x_j \geq 1. \tag{5.26}$$

Furthermore, for a given set $H \subseteq N$, define

$$X(H) \equiv X \cap \{x \in (0,1)^n\colon\ x_{j(i)} = 1 \quad \forall\ i \in M_H\}. \tag{5.27}$$

Then, consider the following result that lays the foundation for a lifting process.

Proposition 5.7. *For a minimal GUB cover K, the minimal GUB cover inequality is a facet of conv(x($\overline{K}$)) if and only if* $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$, *where X(R) is defined by (5.27).*

Proof. The proof follows easily by examining the $|K|$ linearly independent unit vectors e_j, one for each $j \in K$, which belong to $conv(X(\overline{K}))$, and which satisfy the minimal GUB cover inequality as an equality. □

Example 5.2. Consider the following example, where $X \equiv \{x \in (0,1)^8:$ $x_1 + 5x_2 + x_3 + 5x_4 + x_5 + 3x_6 + x_7 + 3x_8 \geq 9,$ $x_1 + x_2 \leq 1,$ $x_3 + x_4 \leq 1,$ $x_5 + x_6 \leq 1,$ $x_7 + x_8 \leq 1\}.$

Since for all $i \in M$, $\sum_{p \in (M-i)} a_{j(p)} \geq 9$, the convex hull of X is a full dimensional polytope by Proposition (5.6). A GUB cover is given by $K = \{1, 2, 3, 4, 5, 6\}$, where $K_+ = \{2, 4, 6\}$ and $K_- = \{1, 3, 5\}$. A minimal GUB cover is given by $K = \{3, 4, 5, 6\}$. Moreover, since $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} = 9 = b$, by Proposition 5.7, the corresponding minimal GUB cover inequality $x_3 + x_4 + x_5 + x_6 \geq 1$ is a facet of $conv(X(\overline{K})) \equiv X \cap \{x \in (0,1)^8:$ $x_2 = 1, x_8 = 1\}.$ □

We remark that within the context of 0-1 programming problems that contain GUB constraints, given a fractional solution to the continuous relaxation, one can set up a separation problem using individual problem constraints along with (a subset of) the GUB constraints in order to possibly generate a minimal GUB cover that deletes this fractional solution. Such an approach would be similar to that used by Crowder *et al.* (1983) (also, see Hoffman and Padberg (1991), except that this separation problem would be a GUB constrained knapsack problem. Note that the transformation suggested by

Johnson and Padberg (1981) can be used to put this GUB constrained knapsack problem in the standard form considered herein. Having generated such a minimal cover, this can be possibly lifted into a facet of GUBKP. Sherali and Lee (1995) describe a polynomial-time strengthening procedure that sequentially lifts a given minimal GUB cover inequality into a facet of GUBKP, given that the condition of Proposition 5.7 holds true. This is actually a sequential-simultaneous process in which each GUB set is considered sequentially, but within any such GUB set, the lifting is performed simultaneously. However, in the present section, we shall focus on a total simultaneous lifting process to illustrate the use of GUBKP $\equiv conv(X)$ given by (5.24) or (5.25).

Consider a minimal GUB cover inequality, and suppose that the condition of Proposition 5.7 holds true. We are interested in finding a lifted inequality, which is a facet of GUBKP, and is of the form $\sum_{j\in K} x_j + \sum_{j\in \overline{K}_-} \pi_j x_j - \sum_{j\in \overline{K}_+} \pi_j \bar{x}_j \geq 1$ where $\bar{x}_j \equiv (1 - x_j)\ \forall\ j \in N$, and $\pi_j\ \ \forall\ j \in \overline{K}$ are unrestricted in sign. This is of the form

$$\sum_{j\in K} x_j + \sum_{j\in \overline{K}_-} \pi_j x_j + \sum_{j\in \overline{K}_+} \pi_j x_j \geq 1 + \sum_{j\in \overline{K}_+} \pi_j. \tag{5.28}$$

Motivated by (5.25) and the form of (5.28), consider a polyhedral set $\Pi_{\overline{K}}$ where

$$\Pi_{\overline{K}} \equiv \{\pi_j,\ j \in K\!:\ (\pi, \pi_0) \in \Pi,\ \pi_j = 1\ \ \forall\ j \in K,\ \pi_0 = 1 + \textstyle\sum_{j\in \overline{K}_+} \pi_j\}.$$

Let us now proceed through a series of simplifications in characterizing $\Pi_{\overline{K}}$ more precisely, and then state our main result regarding this set and the facetial nature of (5.28), in the same spirit as that in (5.25).

To begin, observe that $\Pi_{\overline{K}}$ can be represented as follows, where $\pi_{\overline{K}} \equiv \{\pi_j: \ j \in \overline{K}\}$

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}}: \quad \sum_{j \in J \cap \overline{K}} \pi_j \geq 1 + \sum_{j \in \overline{K}_+} \pi_j - |J \cap K|, \quad \forall\ J \in F\}. \tag{5.29}$$

Equivalently, we have,

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}}: \quad \sum_{j \in (\overline{K}_+ - J)} \pi_j - \sum_{j \in J \cap \overline{K}_-} \pi_j \leq |J \cap K| - 1, \quad \forall\ J \in F\}.$$

Note that we need to consider only those $J \in F$ above, for which $\overline{KJ} \equiv (\overline{K}_+ - J) \cup (J \cap \overline{K}_-) \neq \phi$. Hence, we have that

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}}: \quad \sum_{j \in (\overline{K}_+ - J)} \pi_j - \sum_{j \in J \cap \overline{K}_-} \pi_j \leq |J \cap K| - 1,$$

$$\forall\ J \in F \text{ having } \overline{KJ} \neq \phi\}. \tag{5.30}$$

Furthermore, note that we need to examine only the most restrictive constraints in (5.30). Toward this end, define $\text{GUB}(\overline{K}) = \{T \subseteq \overline{K}: |T \cap N_i| \leq 1 \ \ \forall\ i \in M_{\overline{K}}\}$. Now for any $T \in \text{GUB}(\overline{K})$, define the feasible extension of T as $J(T) = \{J \in F: \ \ J = J_1 \cup T$ for some $J_1 \subseteq K\}$, and let $\overline{KT} = (\overline{K}_+ - T) \cup (T \cap \overline{K}_-)$. Accordingly, define

$$F_K = \{T \in \text{GUB}(\overline{K}): \ \ J(T) \neq \phi, \text{ and } \overline{KT} \neq \phi\}.$$

Then, we can restate (5.30) as follows.

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}}: \sum_{j\in(\overline{K}_+-T)} \pi_j - \sum_{j\in T\cap\overline{K}_-} \pi_j \leq \min\{|J \cap K| : J \in J(T)\} - 1$$

$$\forall\ T \in F_K\}. \tag{5.31}$$

We now consider an explicit representation of the minimization problem in (5.31). Note that $T \in F_K$ if and only if (i) $|T \cap N_i| \leq 1$ for $i \in M_{\overline{K}}$, i.e., $T \in \text{GUB}(\overline{K})$, (ii) $\sum_{j\in K_+} a_j + \sum_{j\in T} a_j \geq b$, i.e., $J(T) \neq \phi$, and (iii) $\overline{KT} \neq \phi$. Hence, for each $T \in F_K$, we can represent the minimization problem in (5.31), denoted by *AGUBKP(T)*, as follows.

AGUBKP(T): Minimize $\{ \sum_{j\in K_+} y_j : \sum_{j\in K_+} a_j y_j \geq b - \sum_{j\in T} a_j ,$

$$y_j \in (0,1) \quad \forall\ J \in K_+\}.$$

Note that $1 \leq v(\text{AGUBKP}(T)) \leq |M_K|$, where $v(P)$ denotes the optimal objective value of the corresponding problem P. Of course, AGUBKP is an easy problem in the sense that we can readily compute $v(\text{AGUBKP}(T))$ for each $T \in F_K$, using a greedy procedure. Let $\overline{b} = b - \sum_{j\in T} a_j$. If $\overline{b}$ is less than or equal to $\max_{j\in K_+}(a_j)$, then $v(\text{AGUBKP}(T)) = 1$. Otherwise, if $\overline{b}$ is less than or equal to the sum of the first two largest a_j's for $j \in K_+$, then $v(\text{AGUBKP}(T)) = 2$, and so on. Hence, the time complexity of solving AGUBKP is $O(|M|\log|M|)$. Let $N(T) = v(\text{AGUBKP}(T)) - 1$. Then, we have

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}}: \sum_{j\in(\overline{K}_+-T)} \pi_j - \sum_{j\in T\cap\overline{K}_-} \pi_j \leq N(T) \quad \forall\ T \in F_K\}. \tag{5.32}$$

Proposition 5.8. *For a minimal GUB cover K, the inequality (5.28) is a valid inequality for GUBKP if and only if* $\pi_{\overline{K}} \in \Pi_{\overline{K}}$, where $\pi_{\overline{K}} = \{\pi_j : \; j \in \overline{K}\}$, *and* $\Pi_{\overline{K}}$ *is given by (5.32).*

Proof. $\pi x \geq \pi_0$ is valid for GUBKP if and only if $\sum_{j \in J} \pi_j \geq \pi_0 \quad \forall \; J \in F$, that is, from (5.25), if and only if $(\pi, \pi_0) \in \Pi$. Hence, noting the form of (5.28) and the derivation of (5.32), we have that (5.28) is valid for GUBKP if and only if $\pi_{\overline{K}} \in \Pi_{\overline{K}}$, where $\Pi_{\overline{K}}$ is given by (5.32). This completes the proof. $\square$

Proposition 5.9. *Let K be a minimal GUB cover such that* $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$, *and let* $\Pi_{\overline{K}}$ *be given by (5.32). Then, the inequality (5.28) having* $1 + \sum_{j \in \overline{K}_+} \pi_j > 0$ *is a facet of GUBKP if and only if* $(\pi_j, \; j \in \overline{K})$ *is a vertex of* $\Pi_{\overline{K}}$ *with* $1 + \sum_{j \in \overline{K}_+} \pi_j > 0$.

Proof. Similar to the foregoing derivation of (5.25), using Chapter 2, it is readily verified that $\pi x \geq 1$ is a facet of GUBKP if and only if π is an extreme point of Π_1, where

$$\Pi_1 = \{\pi : \sum_{j \in J} \pi_j \geq 1 \quad \forall \; J \in F\}. \tag{5.33}$$

Hence, (5.28) with $\pi_j = \overline{\pi}_j$ for $j \in \overline{K}$ is a facet of GUBKP if and only if the scaled partitioned vector $\hat{\pi}$, where

$$\hat{\pi} = \{(\hat{\pi}_j = \frac{1}{(1 + \sum_{j \in \overline{K}_+} \overline{\pi}_j)}, \; j \in K), \; (\hat{\pi}_j = \frac{\overline{\pi}_j}{(1 + \sum_{j \in \overline{K}_+} \overline{\pi}_j)}, \; j \in \overline{K})\} \tag{5.34}$$

is an extreme point of (5.33).

Now, for each $j \in K$, the set $J(j) \equiv \{j\} \cup \overline{K}_+ \in F$ by the hypothesis of the theorem, and the corresponding constraints of (5.33) are linearly independent and are binding at the solution (5.34). The latter $|K|$ linearly independent equality constraints appear as

$$\pi_j = 1 - \sum_{t \in \overline{K}_+} \pi_t \text{ for } j \in K, \tag{5.35}$$

and so determine π_j, $j \in K$, uniquely in terms of π_j, $j \in \overline{K}_+$. Now, (5.34) is a vertex of (5.33) if and only if it is feasible to (5.33) and there exist some $|\overline{K}|$ hyperplanes binding from (5.33) that are linearly independent in combination with (5.35). This happens if and only if $\{\hat{\pi}_j, j \in \overline{K}\}$ is an extreme point of the set obtained by imposing (5.35) on (5.33), i.e., the set

$$\{\pi_{\overline{K}} : \sum_{j \in J \cap \overline{K}} \pi_j \geq 1 + |J \cap K| (\sum_{t \in \overline{K}_+} \pi_t - 1) \quad \forall J \in F\}. \tag{5.36}$$

This holds if and only if $(\hat{\pi}_j, j \in \overline{K})$ is feasible to (5.36), and there exist some $|\overline{K}|$ linearly independent hyperplanes that are binding at $(\hat{\pi}_j, j \in \overline{K})$. Feasibility of $\hat{\pi}$ to (5.36) requires from (5.34) that

$$\sum_{j \in J \cap \overline{K}} \frac{\overline{\pi}_j}{(1 + \sum_{t \in \overline{K}_+} \overline{\pi}_t)} \geq 1 + |J \cap K| \{\frac{\sum_{t \in \overline{K}_+} \overline{\pi}_t}{(1 + \sum_{t \in \overline{K}_+} \overline{\pi}_t)} - 1\} \quad \forall j \in F.$$

That is, we must have

$$\sum_{j \in J \cap \overline{K}} \overline{\pi}_j \geq 1 + \sum_{t \in \overline{K}_+} \overline{\pi}_t - |J \cap K| \quad \forall J \in F. \tag{5.37}$$

Note by (5.29) that (5.37) is equivalent to requiring that $\bar{\pi}_{\bar{K}}$ belongs to $\Pi_{\bar{K}}$. Moreover, an inequality in (5.36) is binding at $\hat{\pi}$ if and only if the corresponding inequality in (5.37) is binding. Also, a collection of $|\bar{K}|$ linearly independent equations from (5.36) give $\hat{\pi}$ as the unique solution if and only if the corresponding $|\bar{K}|$ equations from (5.37) give $\bar{\pi}$ as the unique solution, since from (5.34), there is a 1-1 correspondence between $\hat{\pi}$ and $\bar{\pi}$ according to

$$\{\hat{\pi}_j = \frac{\bar{\pi}_j}{(1 + \sum_{j \in \bar{K}_+} \bar{\pi}_j)} \ \forall\ j \in \bar{K}\} \text{ and } \{\bar{\pi}_j = \frac{\hat{\pi}_j}{(1 - \sum_{j \in \bar{K}_+} \hat{\pi}_j)} \ \forall\ j \in \bar{K}\}.$$

Hence, $\hat{\pi}_{\bar{K}}$ is an extreme point of (5.36) if and only if $\bar{\pi}_{\bar{K}}$ is an extreme point of $\Pi_{\bar{K}}$ with $1 + \sum_{j \in \bar{K}_+} \bar{\pi}_j > 0$, and this completes the proof. □

As mentioned above, the foregoing proposition can be used in the context of a separation problem for generating a simultaneously lifted facet of GUBKP, based on a given minimal GUB cover, that deletes a given fractional solution to the continuous relaxation of a 0-1 problem. In this context, an additional linear program would need to be solved over the polytope (5.32) in order to generate the cut coefficients for $j \in \bar{K}$, where K is the minimal GUB cover being used. Note that if such a strategy is being used as in Crowder *et al.* (1983) and Hoffman and Padberg (1991) within an overall algorithm for solving a 0-1 integer program, then if this problem is sparse, we might expect $|\bar{K}|$ and $|M|$ to be manageably small when employing a GUB constrained polytope based on a single problem constraint, and so, the generation of the foregoing facetial cut would not be too computationally burdensome. In this type of an analysis, for further restricting a subset of the cut coefficients *a priori* before employing a reduced sized set (5.32) along with Propositions 5.8 and 5.9 to generate the remaining coefficients that define strong valid inequalities, it would be computationally useful to have knowledge of lower and upper bounds on the simultaneously lifted facet coefficients π_j for $j \in \overline{K}$ in (5.28). Such bounds are derived in Sherali and Lee (1995) based on the foregoing constructs, and we refer the reader to this paper for further discussion. We conclude this section with an illustrative example.

Example 5.3. Consider the following example to illustrate the above simultaneous lifting procedure. Let $X \equiv \{x \in (0,1)^9: \quad x_1 + x_2 + 2x_3 + x_4 + x_5 + 2x_6 + x_7 + x_8 + 3x_9 \geq 4,\ x_1 + x_2 + x_3 \leq 1,\ x_4 + x_5 + x_6 \leq 1,\ x_7 + x_8 + x_9 \leq 1\}$. A minimal GUB cover inequality is $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1$, which is a facet of $conv(X(\overline{K}))$, where $\overline{K} = \{7, 8, 9\}$. For this minimal cover, we have that $F_{\overline{K}} = \{\emptyset, \{7\}, \{8\}\}$, thereby leading to the following computations.

T	$\overline{KT}$	$\sum_{j \in T} a_j$	v(AGUBKP(T))	Inequalities of $\Pi_{\overline{K}}$ in (5.32)
$\emptyset$	9	0	2	$\pi_9 \leq 1$
7	7, 9	1	2	$\pi_9 - \pi_7 \leq 1$
8	8, 9	1	2	$\pi_9 - \pi_8 \leq 1$

The point (0, 0, 1) is the only vertex of $\Pi_{\overline{K}}$ with $1 + \sum_{j \in \overline{K}_+} \pi_j = 2 > 0$. Hence, $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_9 \geq 2$ is the only facet obtainable from the minimal GUB cover inequality by the lifting procedure.

5.3. Tight Representations and Strong Valid Inequalities for Set Partitioning Problems

The set partitioning problem can be stated as follows.

SP: Minimize $\{cx\colon\ Ax = e,\ x_j = 0 \text{ for } 1\ \forall j \in N\}$

where $A = (a_{ij})$ is an $m \times n$ matrix of zeros and ones, e is a vector of m ones, and $N = \{1, \ldots, n\}$. Also, let us denote $M = \{1, \ldots, m\}$, and let a_j represent the j^{th} column of A. We will assume that A has no zero rows or columns, that rank $(A) = m \leq n$, and that *SP* is feasible.

Problem *SP* has been extensively investigated by several researchers for the last 30 years because of its special structure and its numerous practical applications, such as in crew scheduling, truck scheduling, information retrieval, circuit design, capacity balancing, capital investment, facility location, political districting, and radio communication planning.

In this section, we will apply RLT to SP in order to generate a specialized hierarchy of relaxations by exploiting the structure of this set partitioning polytope. We will then show that several known seminal classes of valid inequalities for this polytope, as well as related tightening and composition rules as derived by Balas (1977), are automatically captured within the first and second-level relaxations of this hierarchy. Hence, these relaxations provide a unifying framework for a broad class of such inequalities. Furthermore, in an LP-based branch-and-cut procedure, it is possible to implement only partial forms of these relaxations from the viewpoint of generating tighter relaxations that delete the underlying linear programming solution to the set partitioning problem, based on variables that are fractional at an optimum to this problem.

5.3.1. Notation

To facilitate the reading and as a quick reference guide, we summarize our notation below, providing only a verbal description to enhance understanding whenever the meaning is clear. Let us begin by re-writing problem SP as follows, and then provide a list of related notation.

SP: Minimize

$$\left\{ \sum_{j \in N} c_j x_j : \sum_{j \in N_i} x_j = 1 \ \forall i \in M, \ x_j \text{ binary } \forall j \in N \right\}. \tag{5.38}$$

$\overline{SP}$: Linear programming relaxation of SP obtained by replacing x_j binary by $x_j \geq 0 \ \forall \ j = 1, \ldots, n$.

$N = \{1, \ldots, n\} \equiv$ index set of variables.

$M = \{1, \ldots, m\} \equiv$ index set of constraints.

N_i = {set of variables appearing in constraint i}, $\forall\ i \in M$.

M_k = {set of constraints that contain the variable x_k}, $\forall\ k \in N$.

$\overline{N}_i = N - N_i\ \forall\ i \in M$.

$\overline{M}_k = M - M_k\ \forall\ k \in N$.

$M_J \equiv \bigcup_{k \in J} M_k$ = set of constraints containing any variable from the set $J \subseteq N$.

$\overline{M}_J \equiv$ complement of $M_J \equiv$ set of constraints not containing any variable from the set $J \subseteq N$.

$N_{i,k} \equiv$ set of variables in constraint i that do not appear with x_k in any row of *SP* for $i \in \overline{M}_k$, $k \in N$.

$N_{i,J} \equiv \bigcap_{k \in J} N_{i,k}$, for $i \in \overline{M}_J$, $J \subseteq N \equiv$ set of variables in constraint i that do not appear with any variable from $J \subseteq N$ in any row of *SP*.

$L(k) \equiv \bigcup_{i \in \overline{M}_k} N_{i,k}$, for $k \in N \equiv$ set of variables in Problem *SP* that do not appear in any constraint along with x_k, for $k \in N$.

$\overline{L}(k) \equiv N - L(k) - \{k\}$, the complement of $L(k)$, other than k itself.

5.3.2. A Specialized Hierarchy of Relaxations for the Set Partitioning Polytope

Recall from Chapter 2 that in the presence of equality constraints, as in Problem SP, the RLT relaxation at level d, for $d = 1, \ldots, n$ would be constructed as follows.

First, we would multiply each of the equalities in (5.38) by each of the factors $F_d(J) \equiv F_d(J, \emptyset)$ of degree d. We would also include the constraints representing the nonnegativity of all possible factors $F_d(J_1, J_2)$ of degree d, $d = \min\{\delta + 1, n\}$. Then, using the identity $x_j^2 \equiv x_j$, for each binary variable x_j, $j = 1, \ldots, n$ in the resulting polynomial constraints, we would linearize these constraints by substituting the variable w_J in place of the product term $\prod_{j \in J} x_j$ for each $J \subseteq N$.

Directly applying these steps to the set partitioning Problem SP given by (5.38), we obtain the following polyhedral relaxation SPP_δ at level δ, where we have included additional nonnegativity constraints in (5.40) below for the sake of convenience.

$SPP_\delta = \{(x, w)$:

$$\left[|N_i \cap J| w_J + \sum_{j \in \{N_i - J\}} w_{J+j} = w_J \ \forall\, i \in M \right],$$

$$\forall\, J \subseteq N \ni |J| = d,\ d = 0, \ldots, \delta \tag{5.39}$$

$$f_d(J_1, J_2) \geq 0,\ \forall\ (J_1, J_2) \text{ of order } d,\ d = 1, \ldots, \min\{\delta + 1,\ n\}\}. \tag{5.40}$$

Note that for the case $\delta = 0$, using the fact that $f_0(\emptyset, \emptyset) \equiv 1$, and that $f_1(j, \emptyset) \equiv x_j$ and $f_1(\emptyset, j) \equiv (1 - x_j)$ for $j=1,\ldots,n$, it follows that SPP_0 is simply the feasible region of $\overline{SP}$. Moreover, if we denote the projection of the set SPP_δ onto the space of the original variables x by $SPP_{P\delta}$, we have from (2.7) that

$$SPP \equiv conv\{x \in R^n: \; Ax = e, \; x \text{ binary}\} = SPP_{Pn} \subseteq SPP_{P(n-1)} \subseteq \cdots$$

$$\subseteq SPP_{P1} \subseteq SPP_0. \qquad (5.41)$$

Before proceeding further, let us provide a simplification for SPP_δ in two steps. First, as the following result shows, we can equivalently replace the constraints (5.40) with the following set of simple nonnegativity constraints:

$$w_J \geq 0, \; \forall J \subseteq N \ni |J| = d, \; d = 1,\ldots, \min \; \{\delta + 1, n\}. \qquad (5.42)$$

Proposition 5.10. *For any $\delta \in N$, the constraints (5.39) and (5.42) imply the constraints (5.40) in SPP_δ.*

Proof. Consider the set SPP_δ for any $\delta \in N$. We will use induction on $|J_2|$ to prove the theorem.

(a) Consider any $d \in \{1,\ldots, \min\{\delta + 1, n\}\}$. If $|J_2| = 0$, then $f_d(J_1, J_2) \equiv w_{J_1} \geq 0$ is implied by (5.42). Next, suppose that $|J_2| = 1$, say, $J_2 = \{k\} \in N$. Then,

$$f_d(J_1, J_2) \equiv \left[(1 - x_k) \prod_{j \in J_1} x_j\right]_L \equiv w_{J_1} - w_{J_1+k}. \qquad (5.43)$$

Now, for some $i \in M_k$, we have from (5.39) for $J = J_1$ that

$$w_{J_1} - w_{J_1+k} = \sum_{j\in\{N_i - J_1 - k\}} w_{J_1+j} + \left|N_i \cap J_1\right| w_{J_1} \,. \tag{5.44}$$

From (5.42)-(5.44), it follows that $f_d(J_1, J_2) \geq 0$.

(b) Assume that $f_d(J_1, J_2) \geq 0$, $d = 1, \ldots, \min\{\delta + 1, n\}$, is implied by the constraints (5.39) and (5.42) whenever $\left|J_2\right| = 1, \ldots, (p-1)$, and consider the case of $\left|J_2\right| = p$, where $p \geq 2$. Suppose that $k \in J_2$. Hence, for any appropriate *d*, we can write

$$f_d(J_1, J_2) \equiv \left[(1 - x_k)\prod_{j\in J_1} x_j \prod_{j\in\{J_2 - k\}} (1 - x_j)\right]_L .$$

Now, for some $i \in M_k$, we have the set partitioning constraint $x_k + \sum_{j\in\{N_i - k\}} x_j = 1$. Note that (5.39) includes constraints obtained by multiplying the foregoing constraint with all factors $F_d(J)$, $|J| = 0, 1, \ldots, \delta$, and that $\prod_{j\in J_1} x_j \prod_{j\in\{J_2 - k\}} (1 - x_j)$ is a linear combination of such factors. Hence, by surrogating the constraints obtained in (5.39) by multiplying the (signed) factors in this combination with the foregoing constraint, we get,

$$\left[\left(\sum_{j\in\{N_i - k\}} x_j\right) \prod_{j\in J_1} x_j \prod_{j\in\{J_2 - k\}} (1 - x_j)\right]_L$$

$$\equiv \left[(1 - x_k)\prod_{j\in J_1} x_j \prod_{j\in\{J_2 - k\}} (1 - x_j)\right]_L \equiv f_d(J_1, J_2).$$

Letting $J_2' \equiv \{J_2 - k\}$, the left-hand side of the above equation is comprised of terms of the type $f_d(J_1 + j, J_2')$ for $j \in \{N_i - k\} \ni j \notin J_1 \cup J_2'$, and of the type $f_{d-1}(J_1, J_2')$ for $j \in \{N_i - k\} \ni j \in J_1$, and zeros in case $j \in \{N_i - k\} \cap J_2'$. Since $|J_2'| = (p-1)$, the induction hypothesis implies that all these terms are nonnegative. Hence, $f_d(J_1, J_2) \geq 0$ is also implied, and this completes the proof. □

The second simplification in SPP_δ results upon deleting certain null variables and the resulting trivial constraints, as follows. Examine constraint (5.39) for any $J \subseteq N$, $|J| = d \in \{1, \ldots, \delta\}$, and for any $i \in M_J$, where $M_J \equiv \bigcup_{k \in J} M_k$ as defined above. Since $|N_i \cap J| \geq 1$, the nonnegativity constraints (5.42) imply that $w_{J+j} = 0 \ \forall \ j \in \{N_i - J\}$. Furthermore, if $|N_i \cap J| > 1$, then we also have $w_J \equiv 0$. Hence, for any such J, we need to write (5.39) only for $i \in \overline{M}_J$. Moreover, noting that for $J = \varnothing$, (5.39) represents the original set partitioning constraints, we obtain upon eliminating the identified null variables, a revised equivalent representation of SPP_δ as specified below.

$SPP_\delta = \{(x, w)$:

$$\sum_{j \in N_i} x_j = 1 \ \forall \ i \in M \tag{5.45a}$$

$$\left[\sum_{j \in N_{i,J}} w_{J+j} = w_J \ \forall \ i \in \overline{M}_J\right] \ \forall \ J \subseteq N \ni |J| = d, \ d = 1, \ldots, \delta \tag{5.45b}$$

$[w_J \geq 0$, and

$$w_{J+j} \geq 0 \ \forall j \in N_{i,J}, \ \forall \ i \in \overline{M}_J] \ \forall \ J \subseteq N \ni |J| = d, \ d = 1, \ldots, \delta\}. \tag{5.45c}$$

In particular, we can make the following observation with respect to the convex hull representation *SPP* in the hierarchy (5.41). Let G be the *intersection graph* associated with *SP*, and let $\alpha(G)$ be its *independence number* (see Nemhauser and Wolsey, 1988, for these standard definitions). Note that $\alpha(G)$ essentially represents the maximum number of x-variables that can be simultaneously equal to one in any feasible solution. Assume also that G is connected (otherwise, *SP* is separable) and that G is not a complete graph (or else, *SP* is trivial). Since $\sum_{j \in N} x_j \leq \alpha(G)\ \forall\, x$ feasible to *SP*, we have that $\prod_{j \in J} x_j = 0\ \forall\, J \subseteq N \ni |J| > \alpha(G)$, i.e., $w_J \equiv 0$, for all $|J| = \alpha(G) + 1, \ldots, n$. Hence, we have the following result, and so, no higher level relaxation than that at level $\alpha(G)$ is necessary in (5.41).

Proposition 5.11. $SPP \equiv conv\{x\!: \ Ax = e, x \ binary\} \equiv SPP_{Pa(G)}$. □

We now examine the specialized forms of the first and second-level relaxations SPP_1 and SPP_2, and demonstrate that these relaxations *automatically* subsume (in a *continuous* sense) known classes of valid inequalities, along with various strengthened and composed versions of these inequalities, as proposed by Balas (1977). Hence, these relaxations afford a unifying framework for viewing such inequalities, and admit tight representations that subsume them.

The first-level RLT relaxation SPP_1 of *SPP*, can be written as follows. Note that in this relaxation, w_{jk} is the linearized term for the product $x_j x_k$, $j < k$. We will denote $w_{(jk)}$ to be w_{jk} if $j < k$ and w_{kj} if $k < j$.

$$SPP_1 = \{(x, w):$$

$$\sum_{j \in N_i} x_j = 1 \ \forall i \in M \tag{5.46a}$$

$$\sum_{j \in N_{i,k}} w_{(jk)} = x_k \ \forall i \in \overline{M}_k, \ \forall k \in N \tag{5.46b}$$

$$[x_k \geq 0, \ w_{(jk)} \geq 0 \ \forall \ j \in N_{i,k}, \ \forall \ i \in \overline{M}_k] \ \forall \ k \in N\}. \tag{5.46c}$$

Similarly, we can write the second-level ($\delta = 2$) RLT relaxation SPP_2 of SPP as follows. Note here that $w_{jk\ell}$ is the linearized term for the product $x_j x_k x_\ell$, for $j < k < \ell$, and whenever the indices are not necessarily so arranged, we simply write this product term as $w_{(jk\ell)}$.

$$SPP_2 = \{(x, w):$$

$$\sum_{j \in N_i} x_j = 1 \ \forall \ i \in M \tag{5.47a}$$

$$\sum_{j \in N_{i,k}} w_{(jk)} = x_k \ \forall \ i \in \overline{M}_k, \ \forall \ k \in N \tag{5.47b}$$

$$\sum_{j \in N_{i,\{k,\ell\}}} w_{(jk\ell)} = w_{(k\ell)} \ \forall \ i \in \overline{M}_{\{k,\ell\}}, \ \forall \ k < \ell \in N \tag{5.47c}$$

$x \geq 0$, and

$$[w_{(k\ell)} \geq 0, \ w_{(jk\ell)} \geq 0 \ \forall \ j \in N_{i,\{k,\ell\}}, \ \forall \ i \in \overline{M}_{\{k,\ell\}}] \ \forall \ k < \ell \in N\} \tag{5.47d}$$

Proposition 5.12 below reveals that the polyhedral representation SPP_1 implies the class of *valid elementary inequalities*, given by (5.48) below, as introduced by Balas (1977).

Proposition 5.12. *For every $k \in N$ and $i \in \overline{M}_k$, the inequalities*

$$x_k - \sum_{j \in N_{i,k}} x_j \leq 0 \tag{5.48}$$

are satisfied by all $x \in SPP_1$.

Proof. For any $k \in N$ and $i \in \overline{M}_k$, consider the constraint (5.46b). By the nonnegativity constraints (5.46c), this implies that $w_{(jk)} \leq x_k \ \forall\ j \in N_{i,k}$. Similarly, for each $j \in N_{i,k}$, we have that $k \in N_{t,j}$ for some $t \in \overline{M}_j$. Examining (5.46b) written for this combination of t and j, we get $w_{(jk)} \leq x_j$. Hence, the constraint (5.46b) implies that $x_k = \sum_{j \in N_{i,k}} w_{(jk)} \leq \sum_{j \in N_{i,k}} x_j$, and this completes the proof. □

Balas (1977) develops several strengthening procedures and composition rules to generate additional valid inequalities from such elementary inequalities. We will show that these strengthening procedures are imbedded within the structure of SPP_1 and SPP_2, and so, by directly employing the reformulations SPP_1 or SPP_2 in solving the set partitioning problem, we *automatically* incorporate many of these strong valid inequalities.

To discuss the relationship between the aforementioned strengthening and composition rules with the relaxations SPP_1 and SPP_2, let $L(k)$ and its complement $\overline{L}(k)$ be as defined in the list of notation given above. For a given $k \in N$, let $N_k^0 = \{j \in L(k)\colon\ x_j = 0 \ \forall\, x$ feasible to SP and having $x_k = 1\}$. While finding the entire set N_k^0 is impractical, we can easily construct a subset of N_k^0 for some $k \in N$ as follows. Note that $w_{(k\ell)} = 0$ for all feasible solutions to any RLT formulation implies that $x_k + x_\ell \leq 1$ for any vertex x of *SPP*, *i.e.*, $\ell \in N_k^0$. Now, consider the following constraints (5.47c) of SPP_2 for a given $k \in N$, and some $\ell \in L(k)$:

$$w_{(k\ell)} = \sum_{j \in N_{i,\{k,\ell\}}} w_{(jk\ell)} \quad \forall\ i \in \overline{M}_{\{k,\ell\}}.$$

Let us define $z_{(k\ell)}$ as

$$z_{(k\ell)} = \text{maximum } \{w_{(k\ell)} \colon\ w_{(k\ell)} = \sum_{j \in N_{i,\{k,\ell\}}} w_{(jk\ell)},\ \forall\ i \in \overline{M}_{\{k,\ell\}},$$

$$w_{(jk\ell)} \geq 0\ \forall\ (jk\ell),\ w_{(k\ell)} \leq 1\}. \tag{5.49}$$

By its structure, $z_{(k\ell)}$ equals zero or one. Hence, if $z_{(k\ell)} = 0$, then $w_{(k\ell)} = 0$, i.e., $\ell \in N_k^0$. Therefore, we can delete $w_{(k\ell)}$ from the first-level RLT formulation SPP_1.

The above procedure for detecting a set of zero $w_{(k\ell)}$ variables generalizes Balas' two procedures for strengthening elementary inequalities. We show below that these two procedures yield simple sufficient conditions for the optimal solution of (5.49) to be zero. The first of these procedures is considered in the following proposition.

Proposition 5.13. *(Special case of Proposition 3.1, Balas, 1977.) For some $k \in N$, consider the valid elementary inequalities $x_k - \sum_{j \in N_{i,k}} x_j \leq 0$, for $i \in \overline{M}_k$. For each $j \in L(k)$, define $N(j) = \bigcup_{i \in \overline{M}_k \ni j \in N_{i,k}} N_{i,k} \setminus \{j\}$, and for each $i \in \overline{M}_k$, let*

$T_i = \{j \in N_{i,k}: N_{h,k} \subseteq N(j)$ *for some* $h \in \overline{M}_k\}$. *Then, the inequalities* $x_k - \sum_{j \in N_{i,k} \setminus T_i} x_j \leq 0, \ \forall \ i \in \overline{M}_k$, *are valid for SPP.* □

Since for any $\ell \in T_i$, $i \in \overline{M}_k$, we have $N_{h,k} \subseteq N(\ell)$ for some $h \in \overline{M}_k$, we then have that $N_{h,\{k,\ell\}} = \emptyset$. Moreover, $h \in \overline{M}_\ell$ or else we would have $\ell \in N_{h,k}$, while $\ell \notin N(\ell)$. From (5.47c) written for this $h \in \overline{M}_{\{k,\ell\}}$, we get $w_{(k\ell)} = \sum_{j \in N_{h,\{k,\ell\}}} w_{(jk\ell)} = 0$. Hence, Proposition (5.13) is a trivial sufficient condition to guarantee that $z_{(k\ell)} = 0$ in (5.49). In particular, using $w_{(jk)} = 0 \ \forall \ j \in T_i$ in (5.46b), and applying the argument of Proposition 5.12, we see that the strengthened valid inequality of Proposition 5.13 is implied by SPP_2. Hence, SPP_2 automatically incorporates such strengthened versions of (5.48) within itself.

To further generalize this discussion related to Proposition 5.13, consider the following result.

Proposition 5.14. *Consider any* $k \in N$ *and* $i \in \overline{M}_k$. *Then, given a* $Q \subseteq N_{i,k}$, *the inequality*

$$x_k - \sum_{j \in Q} x_j \leq 0 \tag{5.50}$$

is valid for SPP if and only if $w_{(jk)} = 0 \ \forall \ j \in (N_{i,k} - Q)$ *for any feasible solution (x, w) to* SPP_1 *having x binary.*

Proof. Note from Theorem 2.1 that SPP_1 with the added restriction that x is binary valued (call this problem SPP_1 (x binary)) is equivalent to *SP*. Hence, if (5.50) is valid for SPP, then it is also valid for SPP_1 (x binary), and so by multiplying this with x_k and linearizing, the constraint $x_k - \sum_{j \in Q} w_{(jk)} \leq 0$ is valid for SPP_1 (x binary). From (5.46b), it then follows that $w_{(jk)} = 0 \; \forall \; j \in (N_{i,k} - Q)$, because $w_{(jk)} \geq 0 \; \forall \; j, k$. Conversely, if $w_{(jk)} = 0 \; \forall \; j \in (N_{i,k} - Q)$ in SPP_1 (x binary), and this is directly imposed in SPP_1, then (5.46b) becomes $\sum_{j \in Q} w_{(jk)} = x_k$. As in Proposition 5.12, this implies that $x_k - \sum_{j \in Q} x_j \leq 0$ is valid for SPP_1 (x binary), and hence, for SPP. This completes the proof. □

In the light of Proposition 5.14, for some $k \in N$, if we were given instead in Proposition 5.13 that the inequalities $x_k - \sum_{j \in Q} x_j \leq 0$ for $i \in \overline{M}_k$, for some index sets $Q_i \subseteq N_{i,k}$, are valid for SPP, then a similar tightening of these inequalities is possible, with the tightened versions being automatically subsumed within SPP_2, by simply replacing $N_{i,k}$ by Q_i in SPP_1 and SPP_2, after fixing $w_{(jk)} = 0 \; \forall \; j \in (N_{i,k} - Q_i)$, $i \in \overline{M}_k$. This reconstructs Balas' Proposition 3.1.

The following examples illustrate that not only does SPP_2 subsume the tightened inequalities of Proposition 5.13, but because this proposition is only a sufficient condition for $z_{(k\ell)}$ to be zero in (5.49), it inherently accommodates other strengthened versions of (5.48) as well.

Example 5.4. (Example 3.1 in Balas, 1977.) Consider the following coefficient matrix A (where the blank spaces are zeros) for a set partitioning polytope having $m = 5$ and $n = 15$.

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15
\end{array} \\
\left[\begin{array}{ccccccccccccccc}
1 & & & & & 1 & 1 & 1 & 1 & & & & 1 & 1 & \\
1 & 1 & & & & & & & & 1 & 1 & & & & 1 \\
 & 1 & 1 & & & 1 & 1 & 1 & & & & 1 & & & \\
 & 1 & 1 & 1 & & & & 1 & 1 & & 1 & & & & \\
 & & 1 & 1 & 1 & & 1 & & & 1 & & 1 & 1 & &
\end{array}\right]
\end{array}
$$

Consider $k = 1$, for which $\overline{M}_1 = \{3, 4, 5\}$, and $N_{3,1} = \{3, 12\}$, $N_{4,1} = \{3, 4\}$, and $N_{5,1} = \{3, 4, 5, 12\}$. Let us examine the procedure of Proposition 5.13 to strengthen the inequality $x_1 - x_3 - x_{12} \le 0$ associated with $N_{3,1}$. We have that $N(3) = \{4, 5, 12\}$, $N(12) = \{3, 4, 5\}$, and we find that $N_{4,1} \subseteq N(12)$. Hence, $T_3 = \{12\}$, and the above inequality can be replaced by $x_1 - x_3 \le 0$.

We now demonstrate how this strengthened inequality is automatically implied by SPP_2. Consider the equalities of type (5.47b) for $k = 1$: $x_1 = w_{1,3} + w_{1,12}$ for $i = 3$, $x_1 = w_{1,3} + w_{1,4}$ for $i = 4$, and $x_1 = w_{1,3} + w_{1,4} + w_{1,5} + w_{1,12}$ for $i = 5$. Let us examine the form of equality (5.47c) for $\ell = 12$ in this second-level RLT formulation SPP_2. Since $\overline{M}_{12} = \{1, 2, 4\}$, we have that $\overline{M}_{\{1,12\}} = \overline{M}_1 \cap \overline{M}_{12} = \{4\}$, and for $h = 4$, we get $N_{4,1} = \{3, 4\}$ and $N_{4,12} = \{9, 11\}$. Therefore, $N_{4,\{1,12\}} = N_{4,1} \cap N_{4,12} = \varnothing$. Hence, the inequality of type (5.47c) for $i = 4$, $k = 1$ and $\ell = 12$ is $w_{1,12} = \sum_{j \in N_{4,\{1,12\}}} w_{(j,1,12)} = 0$. In particular, this yields in the above constraint (5.47b) written for $i = 3$ that $x_1 = w_{1,3} \le x_3$. □

Example 5.5. (Example 3.2 in Balas, 1977.) This example illustrates that SPP_2 captures strengthened versions of (5.48) beyond that of Proposition 5.13. Consider the following coefficient matrix A for a set partitioning polytope having $m = 7$ and $n = 10$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\begin{bmatrix}
1 & & & & & & & & 1 & 1 \\
1 & & & & & & & & & 1 \\
 & 1 & & & 1 & & 1 & & 1 & \\
 & 1 & & & & 1 & & 1 & & \\
 & & 1 & & 1 & & & 1 & & 1 \\
 & & 1 & 1 & & 1 & & & 1 & \\
 & & & 1 & 1 & & 1 & & &
\end{bmatrix}
\end{array}$$

For $k = 1$, we have that $\overline{M}_1 = \{3, 4, 5, 6, 7\}$, and $N_{3,1} = \{2, 5, 7\}$, $N_{4,1} = \{2, 6, 8\}$, $N_{5,1} = \{3, 5, 8\}$, $N_{6,1} = \{3, 4, 6\}$, and $N_{7,1} = \{4, 5, 7\}$. The reader can verify that Balas' first strengthening procedure does not apply to any of the elementary inequalities associated with $k = 1$. On the other hand, consider SPP_2. The constraints (5.47b) for $k = 1$ can be stated as follows: $x_1 = w_{1,2} + w_{1,5} + w_{1,7}$, $x_1 = w_{1,2} + w_{1,6} + w_{1,8}$, $x_1 = w_{1,3} + w_{1,5} + w_{1,8}$, $x_1 = w_{1,3} + w_{1,4} + w_{1,6}$, and $x_1 = w_{1,4} + w_{1,5} + w_{1,7}$. Furthermore, consider the second-level constraints of type (5.47c) for $\ell = 2$ and $k = 1$, where $\overline{M}_2 = \{1, 2, 5, 6, 7\}$, so that $\overline{M}_{\{1,2\}} = \overline{M}_1 \cap \overline{M}_2 = \{5, 6, 7\}$. Also, we have, $N_{5,2} = \{3, 10\}$, $N_{6,2} = \{3, 4\}$, and $N_{7,2} = \{4\}$. This gives $N_{5,\{1,2\}} = \{3\}$, $N_{6,\{1,2\}} = \{3, 4\}$, and $N_{7,\{1,2\}} = \{4\}$. Consequently, the constraints of type (5.47c) for $k = 1$ and $\ell = 2$ are of the form $w_{1,2} = w_{1,2,3}$, $w_{1,2} = w_{1,2,3} + w_{1,2,4}$, and $w_{1,2} = w_{1,2,4}$.

This system implies that $w_{1,2} = w_{1,2,3} = w_{1,2,4} = 0$. Hence, the elementary inequality associated with $N_{3,1}$, namely, $x_1 - x_2 - x_5 - x_7 \leq 0$ can be strengthened to $x_1 - x_5 - x_7 \leq 0$ using (5.47b) for $k = 1$ and $i = 3$. Note again that this strengthened inequality is automatically implied within SPP_2. □

We now consider Balas' second strengthening procedure.

Proposition 5.15. *(Proposition 3.2, Balas, 1977.) For some $k \in N$, let the index sets $Q_{ik} \subseteq N_{i,k}$, $i \in \overline{M}_k$, be such that the inequalities $x_k - \sum_{j \in Q_{ik}} x_j \leq 0$, $i \in \overline{M}_k$, are satisfied by all $x \in SPP$. For each $i \in \overline{M}_k$, define*

$$U_{ik} = \{j \in Q_{ik} \colon \ Q_{hk} \cap Q_{hj} = \varnothing \ \text{ for some } \ h \in \overline{M}_{\{k,j\}}\}. \tag{5.51}$$

Then, the inequalities $x_k - \sum_{j \in Q_{ik} \setminus U_{ik}} x_j \leq 0$, $i \in \overline{M}_k$, are satisfied by all $x \in SPP$. □

The foregoing strengthening procedure of Proposition 5.15 can be easily verified to be inherent within SPP_2 as follows. As before, given the validity of $x_k - \sum_{j \in Q_{ik}} x_j \leq 0$, we can set $w_{(jk)} = 0 \ \forall \ j \in (N_{i,k} - Q_{ik})$ for each $i \in \overline{M}_k$, so that the revised $N_{i,k} \equiv Q_{ik}$. Now, condition (5.51) of Proposition 5.15 is a trivial sufficient condition for ensuring that the corresponding problem (5.49) written for indices j and k, where $j \in U_{ik}$, has an objective value of zero. This follows because (5.49) directly includes the simple constraints $w_{(jk)} = 0$ for any $j \in U_{ik}$, noting that $N_{h,\{k,j\}} \equiv N_{h,k} \cap N_{h,j} \equiv Q_{hk} \cap Q_{hj} = \varnothing$ for some $h \in \overline{M}_{\{k,j\}}$. Consequently, with $w_{(jk)} = 0 \ \forall \ j \notin Q_{ik} \setminus U_{ik}$ that appear in (5.47b), by using the argument of Proposition 5.12, we see that the strengthened valid inequality of Proposition 5.15 is also implied by SPP_2.

Note that in a similar spirit, we can employ RLT relaxations higher than the second-level to further strengthen the valid inequalities obtained from the first-level formulation. For example, in the second-level RLT formulation, suppose that $w_{(jk)} = w_{(jk\ell)}$ for some ℓ, and consider the third-level formulation constraint $w_{(jk\ell)} = \sum_{t \in N_{i,\{k,\ell,j\}}} w_{(tjk\ell)}$, for $i \in \overline{M}_{\{j,k,\ell\}}$. If $N_{i,\{k,\ell,j\}} = \emptyset$, then $w_{(jk\ell)} = 0$, and consequently, $w_{(jk)} = 0$. This information can be transferred to SPP_1 to further tighten its formulation. Also, note that Balas' strengthening procedures use only *partial* information regarding the logical implications of SPP_2 for tightening SPP_1. On the other hand, if one has the facility to handle SPP_2 itself directly, then stronger relaxations can be enforced via such an explicit representation of SPP_2.

Balas (1977) has also developed a particular *composition rule* that considers known valid inequalities of the type

$$x_k - \sum_{j \in S} x_j \le 0 \tag{5.52}$$

where $S \subseteq L(k)$, $k \in N$, and composes specific pairs of such inequalities, deriving for each pair another valid inequality of this same type (5.52) that is tighter than the sum of the two inequalities that generated it. Similar to the above analysis, it can be shown that when the two parent inequalities are of the type (5.48) or (5.50), then the resulting inequality obtained by applying this composition rule is implied by SPP_2. Hence, SPP_2 automatically accommodates such additional valid inequalities as well. However, as one might guess, if such a composition is repeatedly applied sequentially to a pair of inequalities selected from the combined set (5.48), (5.50) and (5.52) thus generated, then suitable higher-level RLT representations need to be considered to automatically imply the new composed valid inequalities. Otherwise, if only SPP_1 is considered, then a coefficient reduction step needs to be interspersed in order to derive such inequalities. Details on this demonstration can be found in Sherali and Lee (1996).

5.3.3. Insights into Deleting Fractional Vertices and Generating Manageable Relaxations

We now provide some additional insights into the polyhedral structure of SPP_1, the first level RLT relaxation, as well as suggest guidelines for managing its size in practice. While these comments apply to other higher order relaxations, it is the first level relaxation that would most likely be used in practical implementations.

To begin with, let us show that there exists a family of cutting planes inherent in SPP_1, which in particular, delete all the fractional vertices of $\overline{SP}$. This is a consequence of a more generally applicable result. To see this, consider any zero-one programming problem whose continuous feasible region is given by $X = \{x\colon\ Ax = b,\ 0 \le x \le e\}$. Let $\bar{x}$ be a fractional vertex of X for which a variable x_k, say, is fractional. Suppose that we apply the RLT procedure by multiplying each equation in $Ax = b$ by x_k, and by multiplying each of the inequalities $x_j \ge 0$ and $(1 - x_j) \ge 0\ \forall\ j \ne k$ by x_k and by

$(1 - x_k)$, and then we linearize by letting $x_k^2 = x_k$ and by denoting $x_j x_k = w_{(jk)}$ $\forall\ j \neq k$. Then, by Theorem 2.2, the projection of the resulting constraint set onto the x-variable space represents the convex hull of those vertices of X for which x_k is binary valued, and hence, $\bar{x}$ is no longer feasible to this constraint set. If this RLT process is simultaneously applied to X for each fractional variable, then all the fractional vertices of X will be deleted. In particular, since SPP_1 is constructed by applying the foregoing RLT process using all the variables (and then simplifying), the following proposition follows as a special case.

Proposition 5.16. *Let* $\bar{x} = (\bar{x}_F, \bar{x}_{\bar{F}})$, *where* $F = \{j \in N\text{: } \bar{x}_j$ *is fractional*$\}$ *and* $\bar{F} = N - F$, *be any basic feasible solution for* $\overline{SP}$ *with* $F \neq \varnothing$. *Then* SPP_1 *cuts off* $\bar{x}$. □

Thus far, we have shown that the first and second-level RLT formulations SPP_1 and SPP_2 contain some rich structural properties with respect to generating tight representations for Problem SP. However, in the case of large problem instances, we may not afford the luxury of being able to cope with the size of these resulting reformulations. In such a case, we might wish to construct only a *partial* first or second-level reformulation, viewing only the fractional variables at an optimum basic feasible solution to $\overline{SP}$ as being binary valued, and treating the remaining variables as being continuous in light of the development in Chapter 2.

Below, we derive several classes of such reduced or partial first-level relaxations of SPP, all guaranteed to delete the obtained fractional linear programming solution $\bar{x}$ to Problem $\overline{SP}$, following the preceding argument that supports Proposition 5.16. (Similar constructs can be used to derive partial second or higher level relaxations, in the same manner as SPP_2 was derived from SPP_1.)

Reduced Partial Relaxation $SPP_1(F)$:

Let $\bar{x}$ be an optimal basic feasible solution to $\overline{SP}$, and as before, denote $F = \{j \in N\colon \bar{x}_j \text{ is fractional}\}$ and $\bar{F} = N - F$. Now, multiply each equality in (5.38) by each x_k, $k \in F$, and also construct the *bound-factor product constraints* by multiplying each of the bounding constraints $x_j \geq 0$, $j \in N$, defining $\overline{SP}$, by each x_k and $(1 - x_k)$ for $k \in F$. Having done this, linearize by letting $x_k^2 \equiv x_k \ \forall\ k \in F$, and substituting $w_{jk} = x_j x_k \ \forall\ j < k \in F$, and also $v_{jk} = x_j x_k \ \forall\ j \in \bar{F},\ k \in F$. Note that when both j and k belong to F, the product constraint $[x_j(1 - x_k)]_L \geq 0$, i.e., $w_{jk} \leq x_j$ is implied by the constraints derived when multiplying (5.38) with x_k along with the nonnegativity restrictions, as seen in the proof of Proposition 5.12, and so, these RLT constraints can be additionally deleted from the relaxation. Also, note that there is no need to generate RLT constraints by taking the products of $(1 - x_j) \geq 0$, $j \in N$, with x_k and $(1 - x_k)$ for each $k \in F$, since (5.38) implies that $(1 - x_j) \geq 0$, and so, these RLT constraints would be implied even in a continuous sense by the foregoing RLT constraints (see Chapter 8 for general results related to implied inequalities via RLT constructions). Hence, the resulting first-level RLT formulation with this restricted set of products would then tighten the formulation of $\overline{SP}$, and in particular, would delete $\bar{x}$ by the proof for Proposition 5.16.

Reduced Partial Relaxation $SPP_1'(F)$**:**

Motivated by the foregoing development, if we wanted to further reduce the size of the reformulated, tightened relaxation that would, in particular, delete the linear programming optimum $\bar{x}$ to $\overline{SP}$, we can further restrict the multiplication of each fractional variable, x_k, $k \in F$, to a subset of the constraints in $\overline{M}_k$ as follows. Suppose that we fix all the variables x_j in $\overline{F}$ at their respective binary values $\bar{x}_j$, and examine the separable sets of surviving non-trivial constraints produced thereby. Let $S_1, \ldots, S_p$ be the partition of F corresponding to the fractional variables that appear in each of the, say, p separable sets $1, \ldots, p$ produced in this manner. Now, multiply each $x_k \in S_t$ with each of the *original* (not restricted) set partitioning constraints in set t alone, and then construct the bound-factor product constraints of the type generated for $SPP_1(F)$ above, corresponding to all the product terms that are thus created. This produces the desired partial relaxation $SPP_1'(F)$ that deletes $\bar{x}$. The motivation here is that when x_j is fixed at $\bar{x}_j$ for $j \in \overline{F}$, the separable linear programs for each t produce complete fractional solutions, and so RLT can be used to eliminate these particular solutions while providing a tighter representation with respect to these partial fractionating constraint sets. (Note that in general, the RLT product term v_{jk} that represents $x_j x_k$ for $j \in \overline{F}$, $k \in F$ takes on precisely the value $v_{jk} = \bar{x}_j x_k$ in any *feasible* solution to the RLT relaxation whenever

x_j is fixed at a binary value $\bar{x}_j$. This produces the desired effect of essentially applying RLT to the reduced system when x_j is fixed at $\bar{x}_j$ $\forall\ j \in \bar{F}$.)

In fact, a minimal RLT extension of $\overline{SP}$ that would guarantee the deletion of $\bar{x}$ could be constructed as follows. Suppose that for any $t \in \{1, \ldots, p\}$ and $k \in S_t$, we generate an RLT formulation $SPP_1'(x_k)$ by multiplying each original set partitioning constraint in set t alone by x_k, and by including bound-factor product (constraints as above) for each product term thus created upon linearization. Then again, by the foregoing arguments, we would obtain a tighter reformulation with the desired property that it deletes $\bar{x}$. The following example illustrates this situation.

Example 5.6. Consider the following set-partitioning problem:

SP: Maximize$\{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \colon\ x_1 + x_2 + x_3 + x_7 = 1,$

$x_1 + x_3 + x_4 + x_8 = 1,\ x_1 + x_4 + x_5 + x_9 = 1,\ x_1 + x_5 + x_6 + x_{10} = 1,$

$x_1 + x_2 + x_6 + x_{11} = 1,\ x_j \in (0, 1)\ \forall\ j = 1, \ldots, 11\}.$

An optimal solution $\bar{x}$ to $\overline{SP}$ is obtained as follows:

$$\left(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6, \bar{x}_7, \bar{x}_8, \bar{x}_9, \bar{x}_{10}, \bar{x}_{11}\right) = \left(0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, 0\right).$$

It can be verified that no elementary inequality cuts off this optimal basic feasible solution $\bar{x}$. Moreover, when x_j is fixed at $\bar{x}_j$ $\forall\ j \in \bar{F}$, we obtain only one set $(p = 1)$, with $S_{p=1} = \{2, 3, 4, 5, 6\} \equiv F$. Consider $2 \in S_1$ and let us construct

$SPP'(x_2)$. To do this, we would multiply the equality partitioning constraints $i \in \overline{M}_2 = \{2, 3, 4\}$ by x_2, as well as each $x_j \geq 0$ for $j \in L(2) \equiv \{4, 5, 8, 9, 10\}$ by x_2 and by $(1 - x_2)$ and then re-linearize the resulting problem by substituting $v_{2,j}$ for $x_2 x_j$ $\forall$ $j \in L(2)$, recognizing that $x_2 x_j \equiv 0$ $\forall$ $j \in \overline{L}(2) = \{1, 3, 6, 7, 11\}$.

The optimal solution to $SPP_1'(x_2)$ turns out to be all integer. In particular, to see why $\bar{x}$ is deleted, note that when we fix $x_j = \bar{x}_j$ for $j \in \overline{F}$, we get $v_{2,8} = v_{2,9} = v_{2,10} = 0$ since $\bar{x}_8 = \bar{x}_9 = \bar{x}_{10} = 0$, and then the product of x_2 with the constraints in $\overline{M}_2$ reduce to $v_{2,4} = x_2$, $v_{2,4} + v_{2,5} = x_2$, and $v_{2,5} = x_2$, which has no solution when $x_2 = \bar{x}_2 = 1/2$. Furthermore, as a point of interest, using the optimal dual vector for $\overline{SP}$, we can construct the surrogate valid inequality $x_2 - x_8 - x_{10} \leq 0$ which also happens to delete $\bar{x}$. It is interesting to note here that by substituting the set partitioning constraints $x_8 = 1 - x_1 - x_3 - x_4$ and $x_{10} = 1 - x_1 - x_5 - x_6$ into this inequality, we obtain $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2$, which happens to be a lifting of the odd hole inequality for the underlying intersection graph of SP. These inequalities are all automatically inherent within $SPP_1'(x_2)$ in this instance. □

In conclusion, observe that there exists a hierarchy of relaxations that one can construct between the extremes of $SPP_1'(x_k)$ and $SPP_1'(F)$, based on generating RLT representations using products of any subset of fractional variables within each corresponding block $t = 1, \ldots, p$. The choice of the RLT constructed can be guided by the nature and size of each block, and the values of the fractional variables. Furthermore, tighter relaxations are available via $SPP_1(F)$ (or its similar partial constructions) and via the complete first-level representation SPP_1 itself. In addition, as mentioned before, (partial) higher-level constraints can be generated, as for example, from SPP_2 based on constructing RLT products using second-order factors composed with a subset of the fractional variables. Such strategies are applicable to other (mixed-integer) zero-one programming problems as well, and hold considerable computational promise, as evident from several related computational studies (see Chapter 10).

6

PERSISTENCY IN DISCRETE OPTIMIZATION

Given a mixed 0-1 linear program in n binary variables, we studied in Chapter 2 the construction of an $n + 1$ level hierarchy of polyhedral approximations that ranges from the usual continuous relaxation at level 0 to an explicit algebraic representation of the convex hull of feasible solutions at level-n. While exponential in both the number of variables and constraints, level-n was shown in the previous chapter to serve as a useful tool for promoting valid inequalities and facets via a projection operation onto the original variable space. In this chapter, we once again invoke the convex hull representation at level-n, this time to provide a direct linkage between discrete sets and their polyhedral relaxations. Specifically, we study conditions under which a binary variable realizing a value of either 0 or 1 in an optimal solution to the linear programming relaxation of a mixed 0-1 linear program will *persist* in maintaining that same value in some discrete optimum.

More formally, an optimal solution to the continuous relaxation of a mixed 0-1 linear programming problem is defined to be *persistent* if the set of 0-1 variables realizing binary values retains those same binary values in at least one integer optimum. A mixed 0-1 linear program is said to possess the persistency property, or equivalently to be persistent, if *every* optimal solution to the continuous relaxation is a persistent solution. The utility of establishing a given formulation as being persistent is evident; the feasible region to the complex discrete optimization problem can be potentially reduced via the solving of a far simpler linear programming approximation. In general, however, it is not an easy task to either recognize a given solution as being persistent or to define and/or verify a persistent formulation.

Given a mixed 0-1 linear program in n binary variables, we will use the RLT to generalize the published knowledge on persistency. Our methodology is to treat the hierarchical levels as stepping stones between the usual linear programming relaxation and the level-n representation, with the highest-level serving as a "theoretical bridge" between the continuous and discrete sets. Suppose that for an arbitrary $d \in \{0, ..., n-1\}$, we establish conditions on an optimal solution to the level-d relaxation of some mixed 0-1 linear (polynomial) program sufficient to ensure that there will exist an optimal solution to the level $d+1$ representation such that every variable realizing a binary value at level d will maintain that same binary value at level $d+1$. Moreover, suppose we also show that this newly-defined solution necessarily satisfies the same sufficiency conditions at level $d+1$. Then we will have effectively established an inductive argument for progressing up the hierarchical levels, with the level-n convex hull representation establishing the desired persistency results. Such arguments are motivated by a study of the unconstrained 0-1 polynomial programming problem, relying on an exploitation of the mathematical structure of certain inequalities found in the sets Z_d of (2.22).

It turns out that our sufficiency conditions for persistency will always be satisfied for special families of problems, and thus we will have that these select classes of problems are persistent. Indeed, these persistent programs directly result from level-1 RLT formulations, and subsume all known persistent formulations for 0-1 linear and quadratic programs. As we will show, by considering the level-1 representations and projections of faces of the associated feasible regions, all published persistency results dealing with the well-studied vertex packing problem and relatives fall out as special cases.

In order to obtain as general a family of persistent formulations as possible, we then rethink through the basic constructs of reformulation and linearization to create a new implementation of the RLT for mixed 0-1 linear (polynomial) programs. Here, we once again use product factors to generate additional implied nonlinear inequalities, and we also linearize by substituting a continuous variable for each distinct nonlinear term. However, the product factors differ from those presented in any of Chapters 2 through 4. They were specially designed for the specific purpose of achieving persistency.

This chapter is organized as follows. We present in Section 6.1 our inductive persistency arguments for unconstrained 0-1 polynomial programs. Thereafter, in Section 6.2 we show that by considering projections of faces of the formulations from Section 6.1, families of persistent 0-1 linear programs emerge. We broaden our concept of reformulation and linearization in Section 6.3 by introducing an entirely different implementation of the RLT for unconstrained 0-1 polynomial programs. We show in Subsection 6.3.1 that the formulations resulting from this modified RLT are persistent. We then consider 0-1 polynomial programs whose linear programming feasible regions consist of faces of the formulations from Subsection 6.3.1, and show in Subsection 6.3.2 these problems to be persistent. Finally, in Section 6.4 we review the literature on persistency and explain how such published results can be viewed within the context of this chapter.

6.1. RLT-Based Persistency for Unconstrained 0-1 Polynomial Programs

We begin our study on persistency by focusing on the unconstrained 0-1 polynomial program. This problem is stated mathematically as follows.

PP: Minimize $\sum_{J \subseteq N} c_J \prod_{j \in J} x_j$

subject to x binary

where $N = \{1, 2, ..., n\}$, $x = (x_1, x_2, ..., x_n)$ is the vector of binary decision variables, and where for each $J \subseteq N$, c_J is the objective function coefficient on the term $\prod_{j \in J} x_j$. We will refer to the degree of Problem PP as the degree of the objective function: that is Problem PP will be said to have degree t if $c_J = 0 \; \forall \; J \subseteq N$ with $|J| > t$ and there exists at least one $J \subseteq N$ with $|J| = t$ such that $c_J \neq 0$. Moreover, we will assume that Problem PP is of fixed degree $t \geq 2$, since otherwise an optimal solution can be trivially computed.

Observe that upon applying the RLT to this problem, the set X_d for each $d \in \{0, ..., n-1\}$ defined in (2.5a - 2.5c) reduces to $X_d = \{(x, w) : f_{d+1}(J_1, J_2) \geq 0$ for each (J_1, J_2) of order $(d+1)$ where, as in (2.4a -2.4b), $w_J = \prod_{j \in J} x_j$ for each $J \subseteq N$, $w_j = x_j$ for $j = 1, ..., n$, and $w_\varnothing = 1$.

For notational convenience, we state the linearized form of Problem PP as Problem IP(d) below where, for consistency with Chapter 2, the parameter $\underline{d}$ indicates that the level $(d-1)$ RLT has been employed, $d \in \{2, ..., n\}$.

IP(d): Minimize $\sum_{J \subseteq N} c_J w_J$

subject to $f_d(J_1, J_2) \geq 0$ for each (J_1, J_2) of order d (6.1)

x binary.

Throughout Sections 6.1 and 6.2 we will not consider the formulation IP($n+1$), effectively restricting $d \leq n$, since Problems IP(n) and IP($n+1$) are identical. Of course, as indicated in Section 2.5, in order to achieve an equivalent linear reformulation of Problem PP, we must have that $d \geq t$, and we make this assumption throughout.

We will henceforth denote the continuous relaxation of Problem IP(d), for each $d \in \{2, ..., n\}$, obtained by deleting the x binary restrictions as Problem LP(d). Recall from Lemma 2.1 that the restrictions $0 \leq x \leq 1$ are implied by the inequalities in IP(d), and consequently we do not explicitly include them within the continuous relaxations. In fact, for each $d \in \{2, ..., n\}$, inequalities (6.1) comprise a special instance of Z_d in (2.22) where the variables y and v are absent.

For our purposes in this chapter, we draw attention here to two properties of inequalities of the type (6.1). First, it follows from (2.2) that for any $d \in \{0, ..., n\}$ and any $(J_1, J_2)_d$ we have

$$f_d(J_1, J_2) = \sum_{J' \subseteq J_2} (-1)^{|J'|} w_{J_1 \cup J'}. \tag{6.2}$$

Second, for any $d \in \{0,...,n - 1\}$ and any $p \in \{1,...,n - d\}$, using a simple inductive argument on the parameter p, the first set of equations in (2.11) can be shown to enforce that for any $(J_1, J_2)_d$ and any $S \subseteq N - (J_1 \cup J_2)$ with $|S| = p$, the inequalities in (6.1) imply that

$$f_d(J_1, J_2) = \sum_{\substack{(S_1, S_2)_p \\ S_1 \cup S_2 = S}} f_{d+p}(J_1 \cup S_1, J_2 \cup S_2). \tag{6.3}$$

In fact, the first equation in (2.11) for given sets (J_1, J_2) of order d is precisely (6.3) when $S = \{t\}$ for $t \notin J_1 \cup J_2$.

As a final preliminary remark, we recall that the arguments of Lemma 2.3 effectively define Problem IP(d), for each $d \in \{t,...,n\}$, to be an equivalent linear reformulation of Problem PP by showing that for each $J \subseteq N$, $\hat{w}_J = \prod_{j \in J} \hat{x}_J$ for every $(\hat{x}, \hat{w})$ feasible to Problem IP(d). In fact, by applying a similar logic to that found in the proof, this argument can be modified to accommodate Problems LP(d) for $d \in \{t, ..., n\}$ as demonstrated in the Lemma below for arbitrary d.

Lemma 6.1. *Consider any feasible solution* $(\hat{x}, \hat{w})$ *to Problem LP(d) and any* $J \subseteq N$ *with* $1 \le |J| \le d$. *If there exists an* $i \in J$ *with* $\hat{x}_i = 1$ *then* $\hat{w}_J = \hat{w}_{J-i}$. *Moreover, if there exists an* $i \in J$ *with* $\hat{x}_i = 0$ *then* $\hat{w}_J = 0$.

Proof. Since $\hat{w}_\varnothing = 1$ and $\hat{w}_j = \hat{x}_j$ for all $j = 1, ..., n$, the results hold trivially for all sets $J \subseteq N$ with $|J| = 1$. Now by induction assume the results hold true for all sets J with $|J| \le r - 1$ for a given r satisfying $2 \le r \le d$. Consider the first statement, and any $J \subseteq N$ with $|J| = r$ such that there exists an $i \in J$ with $\hat{x}_i = 1$. At the point $(\hat{x}, \hat{w})$, $f_r(J - i, i) = \hat{w}_{J-i} - \hat{w}_J \ge 0$ gives $\hat{w}_J \le \hat{w}_{J-i}$ while for any $k \in J$, $k \ne i$, $f_r(J - i - k, \{i, k\}) = \hat{w}_{J-i-k} - \hat{w}_{J-i} - \hat{w}_{J-k} + \hat{w}_J \ge 0$ gives, by the inductive hypothesis, $\hat{w}_J \ge \hat{w}_{J-i}$. Hence $\hat{w}_J = \hat{w}_{J-i}$. Now consider the second statement and any $J \subseteq N$ with $|J| = r$ such that there exists an $i \in J$ with $\hat{x}_i = 0$. At the point $(\hat{x}, \hat{w})$, $f_r(J, \varnothing) = \hat{w}_J \ge 0$ gives $\hat{w}_J \ge 0$ while for any $k \in J$, $k \ne i$, $f_r(J - k, k) = \hat{w}_{J-k} - \hat{w}_J \ge 0$, gives, by the inductive hypothesis, $\hat{w}_J \le 0$. Hence $\hat{w}_J = 0$. □

We are now in position to present our first persistency result, which will provide the basis for upcoming arguments. Given an optimal solution to Problem LP(d) for some $d \in \{t,...,n\}$, if the number of variables realizing a fractional value is no greater than d, then we have that the computed solution is persistent. The logic is that if all variables do not realize values of 0 or 1 to trivially make the solution persistent, then there must exist an alternate optimal solution to the linear program obtainable by assigning binary values to the fractional-valued variables. This is due to the convex hull representation available at the level-n RLT representation. The formal statement and proof is given in the Theorem below.

Theorem 6.1. *Let $(\hat{x}, \hat{w})$ be an optimal primal solution to Problem LP(d), and define subsets N^+, N^-, and N^f of N in terms of $\hat{x}$ as $N^+ = \{j : \hat{x}_j = 1\}$, $N^- = \{j : \hat{x}_j = 0\}$, and $N^f = N - (N^+ \cup N^-)$. If $|N^f| \le d$, then $(\hat{x}, \hat{w})$ is a persistent solution to Problem LP(d). Consequently, there exists an optimal solution x^* to Problem PP with $x_j^* = 1 \;\forall\; j \in N^+$ and $x_j^* = 0 \;\forall\; j \in N^-$.*

Proof. Consider the linear programming problem obtained by fixing $x_j = 1 \;\forall\; j \in N^+$ and $x_j = 0 \;\forall\; j \in N^-$ in Problem LP(d). This reduced linear program by virtue of Lemma 6.1 can be expressed in terms of those variables w_J for which $J \subseteq N^f$, so that the level $|N^f|$ linearization effectively results for a problem having $|N^f|$ binary x variables. As this highest level has x binary at all extreme points by Theorem 2.2, an optimal binary solution, say $(\tilde{x}, \tilde{w})$, exists to this linear program. The solution (x^*, w^*), with x^* defined as $x_j^* = 1 \;\forall\; j \in N^+$, $x_j^* = 0 \;\forall\; j \in N^-$ and $x_j^* = \tilde{x}_j \;\forall\; j \in N^f$, and with w^* defined as $w_J^* = \prod_{j \in J} x_j^* \;\forall\; J \subseteq N$ is therefore optimal to Problem LP(d) (since $\tilde{x}$ was obtained by fixing a subset of the variables at their optimal values and re-solving the resulting problem, and since $w_J^* \;\forall\; J \subseteq N$ is uniquely defined in terms of the binary x^* due to Lemma 2.3) and hence to Problem

IP(d). Thus, $(\hat{x}, \hat{w})$ is a persistent solution to Problem LP(d). Since Problems LP(d) and PP are equivalent for binary x as shown in Lemma 2.3, x^* is an optimal solution to Problem PP. $\square$

A consequence of Theorem 6.1 is that Problem IP($n-1$) possesses the persistency property since, given any optimal solution to Problem LP($n-1$), if at least one variable takes a binary value, the associated set N^f must satisfy $|N^f| \le n-1$. Unfortunately, as we later demonstrate via example, Problems IP(d) for $d \in \{t,\ldots,n-2\}$ do not in general possess the persistency property. Our intent now is to examine the mathematical structure of these formulations for the specific purpose of establishing conditions which, when satisfied, identify persistent solutions. We begin by showing that the constraints of Problem LP(d) have a special structure which allow us, without actually solving the linear programming problem, to determine conditions under which predesignated partial primal and dual feasible solutions are in fact partial primal and dual optimal solutions. This result is stated formally below.

Lemma 6.2. *Consider any partition of the set N into three subsets N^+, N^-, and N^f, and the corresponding partition of the constraints of Problem* LP(d) *as follows.*

$$f_d(J_1, J_2) \ge 0 \quad \forall \quad (J_1, J_2)_d \ \text{ such that } J_1 \cup J_2 \subseteq N^f \tag{6.4}$$

$$f_d(J_1, J_2) \ge 0 \quad \forall \quad (J_1, J_2)_d \ \text{ such that } (J_1 \cap N^-) \cup (J_2 \cap N^+) \ne \emptyset \tag{6.5}$$

$$f_d(J_1, J_2) \ge 0 \quad \forall \quad (J_1, J_2)_d \ \text{ such that } J_1 \cup J_2 \not\subseteq N^f \text{ and }$$
$$(J_1 \cap N^-) \cup (J_2 \cap N^+) = \emptyset \tag{6.6}$$

If there exist multipliers $\hat{\pi}_d(J_1, J_2) \geq 0$ associated with inequalities (6.5) satisfying

$$\sum_{\substack{(J_1, J_2)_d \\ (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \emptyset}} \hat{\pi}_d(J_1, J_2) f_d(J_1, J_2) = \sum_{J \subseteq N} b_J w_J, \text{ where } b_J = c_J \ \forall \ J \not\subseteq N^f \tag{6.7}$$

and b_J is arbitrary otherwise, then there exists an optimal primal solution $(\tilde{x}, \tilde{w})$ to Problem LP(d) having $\tilde{x}_j = 1 \ \forall \ j \in N^+$ and $\tilde{x}_j = 0 \ \forall \ j \in N^-$. Furthermore, if $|N^f| \geq d$, then there exists an optimal dual solution $\tilde{\pi}_d$ with the multipliers $\tilde{\pi}_d(J_1, J_2) \ \forall \ (J_1, J_2)_d$ associated with inequalities (6.6) all 0 and with the multipliers $\tilde{\pi}_d(J_1, J_2) \ \forall \ (J_1, J_2)_d$ associated with inequalities (6.5) having $\tilde{\pi}_d(J_1, J_2) = \hat{\pi}_d(J_1, J_2)$.

Proof. The proof is to construct feasible primal and dual solutions to Problem LP(d) as prescribed in the statement of the Lemma, and then to prove optimality using complementary slackness. We consider the two cases $|N^f| \geq d$ and $|N^f| < d$ separately.

Case 1: $|N^f| \geq d$.

Dual solution. Define $\tilde{\pi}_d(J_1, J_2) = \hat{\pi}_d(J_1, J_2) \ \forall \ (J_1, J_2)_d$ associated with inequalities (6.5), $\tilde{\pi}_d(J_1, J_2) = 0 \ \forall \ (J_1, J_2)_d$ associated with inequalities (6.6), and compute $\tilde{\pi}_d(J_1, J_2) \ \forall \ (J_1, J_2)_d$ corresponding to (6.4) as a set of optimal dual multipliers to the feasible and bounded linear program

$$\text{minimize } \{ \sum_{J \subseteq N} \bar{c}_J w_J \ : \text{subject to (6.4)}\} \tag{6.8}$$

obtained as a Lagrangian dual to Problem LP(d) formed by placing constraints (6.5) and (6.6) into the objective function using the corresponding multipliers $\tilde{\pi}_d(J_1, J_2)$ $\forall$ $(J_1, J_2)_d$ such that $J_1 \cup J_2 \not\subseteq N^f$. Inequalities (6.4) contain a variable w_J of LP(d) if and only if $J \subseteq N^f$, and they define a polytope in these variables (since from Lemma 2.1 the restrictions $0 \le w_J \le 1$ $\forall$ $J \subseteq N^f$ are implied). Consequently, since (6.7) ensures that all dual constraints corresponding to variables w_J with $J \not\subseteq N^f$ are satisfied, solving (6.8) ensures that the remaining dual constraints are satisfied, and since $\tilde{\pi}_d(J_1, J_2) \ge 0$ $\forall$ $(J_1, J_2)_d$, $\tilde{\pi}_d$ is a dual feasible solution to Problem LP(d). (Note that in addition to nonnegativity of the variables, we require

$$\sum_{(J_1, J_2)_d} \pi_d(J_1, J_2) f_d(J_1, J_2) = \sum_{J \subseteq N} c_J w_J$$

for dual feasibility, and this is equivalent to a dual problem in which there is a constraint corresponding to each variable w_J, since the left-hand side coefficient on a particular w_J must be equal to that of the right-hand side.)

Primal solution. Define the solution $(\tilde{x}, \tilde{w})$ in terms of N^+, N^-, and $(\bar{x}, \bar{w})$, an optimal primal solution to (6.8), as follows: $\tilde{x}_j = 1$ $\forall$ $j \in N^+$, $\tilde{x}_j = 0$ $\forall$ $j \in N^-$, $\tilde{x}_j = \bar{x}_j$ $\forall$ $j \in N^f$, and for those $J \subseteq N$ with $2 \le |J| \le d$,

$$\tilde{w}_J = \bar{w}_J \qquad \forall\ J \subseteq N^f, \tag{6.9}$$

$$\tilde{w}_J = 0 \qquad \forall\ J \subseteq N \text{ such that } J \cap N^- \neq \varnothing \tag{6.10}$$

$$\tilde{w}_J = \bar{w}_{J-N^+} \qquad \forall\ J \subseteq N \text{ such that } J \cap N^+ \neq \varnothing. \tag{6.11}$$

$(\tilde{x}, \tilde{w})$ satisfies (6.4) by primal feasibility to (6.8). Next, consider any $(J_1, J_2)_d$ such that $J_1 \cap N^- \neq \emptyset$, and observe from (6.2) that $f_d(J_1, J_2)$ evaluated at $(\tilde{x}, \tilde{w})$ equals 0 by (6.10) since $\tilde{x}_j$ is defined to be 0 $\forall\ j \in N^-$. Similarly, for any $(J_1, J_2)_d$ such that $J_2 \cap N^+ \neq \emptyset$ and any $k \in J_2 \cap N^+$, $f_d(J_1, J_2)$ evaluated at $(\tilde{x}, \tilde{w})$ gives

$$f_d(J_1, J_2) = \sum_{J' \subseteq J_2} (-1)^{|J'|} \tilde{w}_{J_1 \cup J'} = \sum_{J' \subseteq J_2 - k} (-1)^{|J'|} (\tilde{w}_{J_1 \cup J'} - \tilde{w}_{J_1 \cup J' \cup k}) = 0 \qquad (6.12)$$

by (6.2) and (6.11). Hence $(\tilde{x}, \tilde{w})$ is feasible to (6.5). Finally, consider any $(J_1, J_2)_d$ corresponding to an inequality in (6.6). The expression $f_d(J_1, J_2)$ evaluated at $(\tilde{x}, \tilde{w})$ gives

$$f_d(J_1, J_2) = \sum_{J' \subseteq J_2} (-1)^{|J'|} \tilde{w}_{J_1 \cup J'} = \sum_{J' \subseteq J_2 - N^-} (-1)^{|J'|} \tilde{w}_{J_1 \cup J'} =$$

$$\sum_{J' \subseteq J_2 - N^-} (-1)^{|J'|} \tilde{w}_{J_1 - N^+ \cup J'} = f_r(J_1 - N^+, J_2 - N^-) \geq 0 \qquad (6.13)$$

where $r = d - \left|(J_1 \cap N^+) \cup (J_2 \cap N^-)\right| < d$ and $f_r(J_1 - N^+, J_2 - N^-)$ is evaluated at $(\tilde{x}, \tilde{w})$. Here, the first and fourth equalities follow from (6.2), the second equality from (6.10), and the third equality from (6.9) and (6.11). The nonnegativity follows from (6.4) and (6.9), since $(J_1 - N^+) \cup (J_2 - N^-) \subseteq N^f$ in this case.

Complementary Slackness. Complementary slackness holds for inequalities (6.4) by optimality to (6.8). Relative to (6.5), $f_d(J_1, J_2)$ evaluated at $(\tilde{x}, \tilde{w})$ equals 0 for these constraints as in the argument for primal feasibility of $(\tilde{x}, \tilde{w})$. Finally, by definition, $\tilde{\pi}_d(J_1, J_2) = 0 \ \forall \ (J_1, J_2)_d$ corresponding to inequalities (6.6).

Case 2: $|N^f| < d$.

In this case, (6.4) is empty since there exist no $(J_1, J_2)_d$ such that $J_1 \cup J_2 \subseteq N^f$. Let $m = |N^f|$, and replace (6.4) throughout Case 1 with (6.14) defined as

$$f_m(J_1, J_2) \geq 0 \ \forall \ (J_1, J_2)_m \text{ such that } J_1 \cup J_2 = N^f. \tag{6.14}$$

Inequalities (6.14) are implied by (6.5) and (6.6) due to (6.3), and hence the above argument holds relative to the computed optimal primal solution $(\tilde{x}, \tilde{w})$ to Problem LP(d). □

Observe in Case 2 of the above proof that the computed vector $\tilde{\pi}_d$ is dual optimal to Problem LP(d) with the redundant constraints (6.14) included within the problem. While an optimal dual solution to LP(d) for this case can be readily recovered via (6.3) by appropriately increasing the values of the dual variables corresponding to (6.5) and (6.6), such a vector is not explicitly required in our persistency study since (6.8), when solved over (6.14), has an optimal binary solution. Hence, in this case, as well as in the case $|N^f| = d$, we can assume without loss of generality that $(\tilde{x}, \tilde{w})$ is binary with $\tilde{w}_J = \prod_{j \in J} \tilde{x}_j \ \forall \ J \subseteq N$ such that $|J| \geq 2$ so that $(\tilde{x}, \tilde{w})$ is a persistent solution to Problem LP(d) with $\tilde{x}$ optimal to Problem PP.

We now use Lemma 6.2, together with Theorem 6.1, to prove our second persistency result. Given an optimal solution to Problem LP(d), Theorem 6.2 provides conditions sufficient for identifying this solution as persistent. This is accomplished by first defining the subsets N^+, N^-, and N^f of N in terms of the optimal solution to LP(d) as in Theorem 6.1, and then using these sets in the context of Lemma 6.2.

Theorem 6.2. *Let* $(\hat{x}, \hat{w})$ *be an optimal primal solution to Problem LP(d). Define the subsets* N^+, N^-, *and* N^f *of* N *in terms of* $\hat{x}$ *as in Theorem 6.1, and partition the constraints of Problem LP(d) exactly as in Lemma 6.2. If either (i)* $\left|N^f\right| \le d$ *or (ii)* $\left|N^f\right| > d$ *and there exists a dual feasible solution* $\hat{\pi}_d$ *to Problem LP(d) with the multipliers corresponding to inequalities (6.6) all taking value 0, then* $(\hat{x}, \hat{w})$ *is a persistent solution to Problem LP(d). Consequently, there exists an optimal solution* x^* *to Problem PP having* $x_j^* = 1 \ \forall \ j \in N^+$ *and* $x_j^* = 0 \ \forall \ j \in N^-$.

Proof. If $\left|N^f\right| \le d$ then the result holds by Theorem 6.1 while if $\left|N^+ \cup N^-\right| = 0$ the result holds trivially. Otherwise, $1 \le \left|N^+ \cup N^-\right| < n - d$ and by induction the proof reduces to showing that for $p = d + 1$, there exists a nonnegative vector $\tilde{\pi}_p(J_1, J_2) \ \forall \ (J_1, J_2)_p$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \ne \varnothing$ satisfying

$$\sum_{\substack{(J_1, J_2)_p \\ (J_1 \cap N^-) \cup (J_2 \cap N^+) \ne \varnothing}} \tilde{\pi}_p(J_1, J_2) f_p(J_1, J_2) = \sum_{J \subseteq N} b_J w_J \text{ where } b_J = c_J \ \forall \ J \not\subseteq N^f. \tag{6.15}$$

This logic follows since by Lemma 6.2 there must then exist an (optimal) dual solution $\hat{\pi}_p$ to Problem LP(p) with $\hat{\pi}_p(J_1, J_2) = 0 \ \forall \ (J_1, J_2)_p$ such that $J_1 \cup J_2 \not\subseteq N^f$ and

$(J_1 \cap N^-) \cup (J_2 \cap N^+) = \varnothing$. Inductively applying this argument, we get that for each $p \in [d, |N^f|]$ there exists a nonnegative vector $\tilde{\pi}_p(J_1, J_2)\ \forall\ (J_1, J_2)_p$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing$ satisfying (6.15). Consequently, Lemma 6.2 will assure us that for every such p there exists an optimal primal solution $(\hat{x}, \hat{w})$ to Problem LP(p) having $\tilde{x}_j = 1\ \forall\ j \in N^+$ and $\tilde{x}_j = 0\ \forall\ j \in N^-$. Persistency then follows directly from Theorem 6.1 by setting $p = |N^f|$.

To establish (6.15) for $p = d + 1$, we initialize with $\tilde{\pi}_p(J_1, J_2) = 0\ \forall\ (J_1, J_2)_p$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing$ and increment these variables as follows: systematically progress through each $(J_1, J_2)_d$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing$, selecting for each such $(J_1, J_2)_d$ any $k \notin J_1 \cup J_2$, say $k(J_1, J_2)$, and setting $\tilde{\pi}_p(J_1 \cup k(J_1, J_2), J_2) = \tilde{\pi}_p(J_1 \cup k(J_1, J_2), J_2) + \hat{\pi}_d(J_1, J_2)$ and $\tilde{\pi}_p(J_1, J_2 \cup k(J_1, J_2)) = \tilde{\pi}_p(J_1, J_2 \cup k(J_1, J_2)) + \hat{\pi}_d(J_1, J_2)$. Clearly, for each such $(J_1, J_2)_p$, $\tilde{\pi}_p(J_1, J_2) \geq 0$ and, by (2.11), $f_p(J_1 \cup k(J_1, J_2), J_2) + f_p(J_1, J_2 \cup k(J_1, J_2)) = f_d(J_1, J_2)$. It then follows that

$$\sum_{\substack{(J_1, J_2)_p \\ (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing}} \tilde{\pi}_p(J_1, J_2) f_p(J_1, J_2) = \sum_{\substack{(J_1, J_2)_d \\ (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing}} \hat{\pi}_d(J_1, J_2) f_d(J_1, J_2).$$

Hence, since $\hat{\pi}_d(J_1, J_2)\ \forall\ (J_1, J_2)_d$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing$ satisfies (6.7), by dual feasibility of $\hat{\pi}_d$ to LP(d) with the multipliers corresponding to inequalities (6.6) all taking value 0, the multipliers $\tilde{\pi}_p(J_1, J_2)$ satisfy (6.15). $\square$

The above proof of case (ii) essentially shows that if $(\hat{x}, \hat{w})$ is an optimal primal solution to Problem LP(d), and the sufficiency conditions of Lemma 6.2 with respect to this problem are satisfied when N^+, N^-, and N^f are defined in terms of $\hat{x}$ as prescribed in the statement of the Theorem, then there exists an optimal primal solution $(\tilde{x}, \tilde{w})$ to Problem LP(d + 1) with $\tilde{x}_j = \hat{x}_j \ \forall \ j \in N^+ \cup N^-$ such that the sufficiency conditions to Problem LP(d + 1), defined in terms of N^+ and N^-, are also satisfied. In fact, this proof, together with the proof of Lemma 6.2 (with d incremented by 1), provides a linear programming problem of the form (6.8) whose optimal solution can be used to construct such a solution $(\tilde{x}, \tilde{w})$ to Problem LP(d + 1) as prescribed in (6.9), (6.10), and (6.11). This constructive argument may prove useful in developing solution strategies for Problem PP which are designed to exploit the strength of the higher-level linearizations. Consider, for example, a scenario in which the sufficiency conditions for Problem LP(d) are satisfied so that $x_j = 1 \ \forall \ j \in N^+$ and $x_j = 0 \ \forall \ j \in N^-$ constitutes part of an optimal solution to Problem PP. (These sufficiency conditions are always satisfied for Problem LP(2) as we show later in this section.) The analyst has the option at this point to solve Problem LP(d + 1) via a reduced problem (6.8), to redefine N^+ and N^- in terms of the computed optimal solution $(\tilde{x}, \tilde{w})$, and then to check whether the sufficiency conditions to Problem LP(d + 1) are satisfied with respect to these new sets. In this manner, additional variables may be fixed at binary values.

While the statement of case (ii) of Theorem 6.2 is based upon the sets N^+ and N^- computed in terms of an optimal primal solution to Problem LP(d), as suggested by

Lemma 6.2, the proof will hold so long as the sufficiency conditions are satisfied for *any* disjoint sets N^+ and N^-. This observation would allow us to consider, for any optimal solution $(\hat{x}, \hat{w})$ to Problem LP(d), subsets of N^+ and N^- for which the sufficiency conditions might hold even when they fail on the sets themselves or, in specific instances where the problem contains some special structure, to define the sets N^+ and N^- independent of an optimal primal solution. Throughout this and the following section, however, we deal exclusively with those sets N^+ and N^- defined in terms of an optimal primal solution $(\hat{x}, \hat{w})$.

We now present an example to illustrate how Theorem 6.2 can be used to reduce the size of a 0-1 polynomial program by allowing a variable to be fixed at a binary value.

Example 6.1. Consider the instance of Problem PP having degree $t = 3$ in $n = 5$ binary variables given below.

$$\begin{aligned}
\text{Minimize} \quad & x_1 + x_5 - x_1x_2 - x_1x_3 - x_1x_4 + x_1x_5 - x_2x_5 + x_3x_4 - x_3x_5 + \\
& x_1x_2x_3 + x_1x_2x_4 - x_1x_4x_5 - x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 \\
\text{subject to} \quad & x_i \text{ binary} \qquad i = 1, \ldots, 5.
\end{aligned}$$

The reader can verify that the optimal objective function value to Problem LP(3) is $-\frac{1}{3}$ with optimal primal and dual solutions as given below.

Primal Solution. $(x_1, x_2, x_3, x_4, x_5) = \left(\frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 0\right)$, $w_{13} = w_{14} = w_{23} = w_{24} = w_{34} = w_{134} = w_{234} = \frac{1}{3}$, and all other primal variables take value 0.

Dual Solution. $\pi_3(\{1\},\{2,3\}) = \pi_3(\{3\},\{1,2\}) = \pi_3(\{1,2,3\},\emptyset) = \pi_3(\{1\},\{2,4\}) = \pi_3(\{2\},\{1,4\}) = \pi_3(\{4\},\{1,2\}) = \pi_3(\{1\},\{3,4\}) = \pi_3(\{3,4\},\{1\}) = \pi_3(\emptyset,\{2,3,4\}) = \pi_3(\{2,4\},\{3\}) = \pi_3(\{3,4\},\{2\}) = \frac{1}{3}$, $\pi_3(\{1,5\},\{4\}) = \pi_3(\{5\},\{2,3\}) = \pi_3(\{2,4,5\},\emptyset) = \pi_3(\{3,4,5\},\emptyset) = 1$, and all other dual variables take value 0.

Since the hypothesis of case (ii) of Theorem 6.2 is satisfied: namely, $4 = |N^f| > d = 3$ and $N^+ = \emptyset$ and $N^- = \{5\}$ with $\pi_3(J_1, J_2) = 0$ for all $(J_1, J_2)_3$ such that $5 \in J_2$, x_5 must equal 0 at some optimal binary solution to the problem. In fact, $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 0, 0)$ is one such optimal binary solution, having objective function value 0. (x_5 need not be 0 at all optimal solutions, however, as $(0, 1, 0, 0, 1)$ is also optimal with $x_5 = 1$.)

A natural question to arise at this point is whether the sufficient conditions of Theorem 6.2 for recognizing an optimal solution $(\hat{x}, \hat{w})$ to Problem LP(d) as being persistent are also necessary: that is, given an optimal solution $(\hat{x}, \hat{w})$ to Problem LP(d) which is also persistent, must either case (i) or case (ii) of the Theorem hold true? The answer to this question is no, as the following example demonstrates.

Example 6.2. Consider the instance of Problem PP having degree $t = 3$ in $n = 5$ binary variables given below.

Minimize $\quad -2x_1 - 2x_2 + 2x_3 + 2x_4 - x_5 + 5x_1x_2 - 2x_1x_3 - 2x_1x_4 +$

$\quad 4x_1x_5 - 2x_2x_3 - -2x_2x_4 + 4x_2x_5 - 4x_3x_4 + 3x_3x_5 + 3x_4x_5 -$

$\quad 2x_1x_2x_5 + 4x_1x_3x_4 - x_1x_3x_5 - x_1x_4x_5 + 4x_2x_3x_4 - x_2x_3x_5 -$

$x_2x_4x_5 - x_3x_4x_5$

subject to $\quad x_i$ binary $\qquad i = 1, \ldots, 5.$

The optimal objective function value to Problem LP(3) is -3 with an optimal primal solution having $(x_1, x_2, x_3, x_4, x_5) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0\right)$. In this example, $N^+ = \emptyset$ and $N^- = \{5\}$ so that case (i) fails since $4 = \left|N^f\right| > d = 3$. Moreover, since the objective function coefficient on x_5 is -1 and all linear expressions $f_d(J_1, J_2)$ found on the left-hand side of (6.7) have $5 \in J_1$ and therefore have nonnegative coefficients for x_5, no nonnegative solution exists to equations (6.7) so that case (ii) also fails. It can be verified that there exist eight optimal binary solutions, $(x_1, x_2, x_3, x_4, x_5) = (0, 1, *, *, 0)$ and $(x_1, x_2, x_3, x_4, x_5) = (1, 0, *, *, 0)$, all having objective function value -2.

In each of the two above examples the optimal solution provided to Problem LP(d) is a persistent solution. In general, though, the formulations IP(d), $d \in [t, n]$, are not persistent as the next example illustrates.

Example 6.3. Consider the instance of Problem PP having degree $t = 3$ in $n = 5$ binary variables given below.

Minimize $\quad -7x_1 - 8x_2 + 15x_3 + 23x_4 - x_5 + 30x_1x_2 - 16x_1x_3 - 24x_1x_4 +$

$x_1x_5 - 14x_2x_3 - 22x_2x_4 + x_2x_5 - 46x_3x_4 - 7x_1x_2x_3 - 2x_1x_2x_5 +$

$48x_1x_3x_4 + 45x_2x_3x_4$

subject to $\quad x_i$ binary $\qquad i = 1, \ldots, 5.$

The optimal objective function value to Problem LP(3) is -19 with an optimal primal solution having $(x_1, x_2, x_3, x_4, x_5) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0)$. Even though $x_5 = 0$ in this solution, neither of the two optimal solutions to the original integer problem has $x_5 = 0$; the two optimal integer solutions are $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 1, 1)$ and $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 1, 1, 1)$, both giving objective function value -9.

We mention here that, given an optimal solution $(\hat{x}, \hat{w})$ to Problem LP(d), the sufficient conditions of case (ii) of Theorem 6.2 can be made somewhat less restrictive by exploiting the structure of inequalities (6.1). As an example, one can show that given multipliers $\bar{\pi}_d(J_1, J_2)\ \forall\ (J_1, J_2)_d$ which satisfy

$$\sum_{(J_1,J_2)_d} \bar{\pi}_d(J_1, J_2) f_d(J_1, J_2) = \sum_{J \subseteq N} c_J w_J,$$

these multipliers must also satisfy

$$\sum_{\substack{(J_1,J_2)_d \\ (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing}} f_d(J_1, J_2)[\bar{\pi}_d(J_1, J_2) - \bar{\pi}_d([J_1 - N^-] \cup [J_2 \cap N^+],$$

$$[J_2 - N^+] \cup [J_1 \cap N^-])] = \sum_{J \subseteq N} b_J w_J$$

where $b_J = c_J\ \forall\ J \not\subseteq N^f$. Thus, by Lemma 6.2 the restriction that all multipliers corresponding to inequalities (6.6) must take value 0 in Theorem 6.2 can be relaxed to

$$\hat{\pi}_d(J_1, J_2) \geq \hat{\pi}_d([J_1 - N^-] \cup [J_2 \cap N^+], [J_2 - N^+] \cup [J_1 \cap N^-])$$

$$\forall\ (J_1, J_2)_d \text{ with } (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing,$$

since for each such $(J_1, J_2)_d$, the multipliers corresponding to the associated inequality in (6.5) can be defined as the difference between the left- and right-hand sides of this inequality. In effect then, an alternate optimal dual solution will be computed that satisfies Theorem 6.2. Other relaxations of case (ii) can also be obtained by exploiting the increasing strength of the linear programming problems LP(d) as d progresses from t to n. One can, for example, take a computed optimal dual solution $\hat{\pi}_d$ to Problem LP(d) for which the sufficiency conditions in case (ii) fail, and attempt to construct, using (6.3), multipliers $\hat{\pi}_p(J_1, J_2) \geq 0 \; \forall \; (J_1, J_2)_p$ with $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing$ for some $p \in \{d+1,\ldots,n\}$ that satisfy (6.7) with d replaced by p, so that Lemma 6.2 can be invoked for Problem LP(p). For specifics, the interested reader is referred to Adams *et al.* (1998).

We now direct attention to a related persistency issue. Given an instance of Problem PP, Theorems 6.1 and 6.2 provide conditions for identifying persistent solutions to Problems LP(d), $d \in \{t,\ldots,n\}$. Recall that an optimal solution to Problem LP(d) is defined to be persistent if the set of 0-1 variables realizing binary values retains those same binary values in *at least one* integer optimum. Given an optimal primal and dual solution to Problem LP(d), Theorem 6.3 below stipulates conditions under which *all* optimal solutions to Problems LP(p), $p \in \{d,\ldots,n\}$,must lie on computed faces of the feasible region to Problem LP(d), so that all optimal solutions to Problem PP must satisfy $F_d(J_1, J_2) = 0$ for certain $(J_1, J_2)_d$. A special case of Theorem 6.3 provides a set of conditions sufficient for identifying 0-1 variables that realize a given binary value in *all* optimal solutions to Problem PP.

Theorem 6.3. *Let* $(\hat{x}, \hat{w})$ *and* $\hat{\pi}_d$ *be optimal primal and dual solutions, respectively, to Problem LP(d). If* $|N^f| \leq d$, *then for each* $p \in \{d,...,n\}$ *and for every* $(J_1, J_2)_d$ *such that* $\hat{\pi}_d(J_1, J_2) > 0$, *we have that* $f_d(J_1, J_2) = 0$ *at all optimal solutions to Problem LP(p). Consequently, for every such* $(J_1, J_2)_d$, $F_d(J_1, J_2) = 0$ *at all optimal solutions to Problem PP. If* $|N^f| > d$ *and there exist multipliers* $\bar{\pi}_d(J_1, J_2) \geq 0 \;\; \forall \; (J_1, J_2)_d$ *associated with inequalities (6.5) that satisfy (6.7), then for each* $p \in \{d,...,n\}$ *and for every* $(J_1, J_2)_d$ *associated with an inequality in (6.5) for which* $\bar{\pi}_d(J_1, J_2) > 0$, *we have that* $f_d(J_1, J_2) = 0$ *at all optimal solutions to Problem LP(p). Thus, for each such* $(J_1, J_2)_d$, $F_d(J_1, J_2) = 0$ *at all optimal solutions to Problem PP.*

Proof. We consider the two cases $|N^f| \leq d$ and $|N^f| > d$ separately.

Case 1: $|N^f| \leq d$.

Since the hierarchy of linearizations produces a nondecreasing sequence of lower bounds on the optimal objective function value to Problem PP, the proof of Theorem 6.1 ensures that the optimal objective function values to Problems LP(p) $\forall \; p \in \{d,...,n\}$ are the same. Now, by contradiction suppose there exists a $p \in \{d,...,n\}$ and an optimal solution $(\tilde{x}, \tilde{w})$ to Problem LP(p) such that for some $(J_1, J_2)_d$ with $\hat{\pi}_d(J_1, J_2) > 0$, say $(K_1, K_2)_d$, the expression $f_d(K_1, K_2)$ evaluated at $(\tilde{x}, \tilde{w})$ is positive. The point $(\tilde{x}, \overline{w})$ with $\overline{w}$ defined as $\overline{w}_J = \tilde{w}_J \;\; \forall \; J \subseteq N$ such that $2 \leq |J| \leq d$ is feasible to

Problem LP(d) by (6.2) and (6.3) with $f_d(K_1, K_2)$ evaluated at $(\tilde{x}, \overline{w})$ equaling $f_d(K_1, K_2)$ evaluated at $(\tilde{x}, \tilde{w})$. Moreover, since $c_J = 0 \ \forall \ J \subseteq N$ with $|J| > d$, $(\tilde{x}, \overline{w})$ so defined must have the same objective function value in LP(d) as does $(\tilde{x}, \tilde{w})$ in LP(p). Hence, $(\tilde{x}, \overline{w})$ is optimal to Problem LP(d). However, since $\hat{\pi}_d(K_1, K_2) > 0$, the positivity of $f_d(K_1, K_2)$ at the optimal solution $(\tilde{x}, \overline{w})$ contradicts complementary slackness to Problem LP(d). Consequently, for each $p \in \{d, ..., n\}$ and for every $(J_1, J_2)_d$ such that $\hat{\pi}_d(J_1, J_2) > 0$, we have that $f_d(J_1, J_2) = 0$ at all optimal solutions to Problem LP(p). Since the level-n hierarchical formulation has all binary extreme points, $F_d(J_1, J_2) = 0$ at all optimal solutions to Problem PP.

Case 2: $|N^f| > d$.

By induction the proof is to show for arbitrary $p \in \{d, ..., n-1\}$ and $(K_1, K_2)_d$ corresponding to an inequality in (6.5) that if there exists an optimal dual solution $\hat{\pi}_p$ to Problem LP(p) that satisfies (6.7) (for $(d = p)$) with $\hat{\pi}_p(J_1, J_2) > 0 \ \forall \ (J_1, J_2)_p$ such that $K_1 \subseteq J_1$ and $K_2 \subseteq J_2$, then there exists an optimal dual solution $\hat{\pi}_{p+1}$ to Problem LP($p+1$) that satisfies (6.7) (for $d = p+1$) with $\hat{\pi}_{p+1}(J_1, J_2) > 0 \ \forall$ $(J_1, J_2)_{p+1}$ such that $K_1 \subseteq J_1$ and $K_2 \subseteq J_2$. (By Lemma 6.2, regardless of the chosen $(K_1, K_2)_d$, such an optimal dual solution exists for LP(d).) Upon showing this, we will then have by complementary slackness that for each $p \in \{d, ..., n\}$, every optimal solution to LP(p) must satisfy

$$0 = \sum_{\substack{(J_1, J_2)_p \\ K_1 \subseteq J_1, K_2 \subseteq J_2}} f_p(J_1, J_2), \text{ which in turn equals } \binom{n-d}{p-d} f_d(K_1, K_2)$$

by (6.3), yielding the desired result.

To show this inductive step, suppose the premise holds true for LP(p), for some $p \in \{d, ..., n-1\}$, so that such a dual solution $\hat{\pi}_p$ exists. By convexity of the dual region, we need only show for an arbitrarily-selected $(H_1, H_2)_{p+1}$ with $K_1 \subseteq H_1$ and $K_2 \subseteq H_2$ that there exists an optimal dual solution $\hat{\pi}_{p+1}$ to Problem LP(p + 1) having $\hat{\pi}_{p+1}(H_1, H_2) > 0$. Toward this end, define a vector $\hat{\pi}_{p+1}(J_1, J_2)$ in terms of $\hat{\pi}_p$ in the same spirit as the algorithmic process found in the proof of Theorem 6.2. That is, systematically progress through each $(J_1, J_2)_p$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \emptyset$, selecting for each such $(J_1, J_2)_p$ any $k \notin J_1 \cup J_2$, say $k(J_1, J_2)$, and setting

$$\hat{\pi}_{p+1}(J_1 \cup k(J_1, J_2), J_2) = \hat{\pi}_{p+1}(J_1 \cup k(J_1, J_2), J_2) + \hat{\pi}_p(J_1, J_2) \text{ and}$$

$$\hat{\pi}_{p+1}(J_1, J_2 \cup k(J_1, J_2)) = \hat{\pi}_{p+1}(J_1, J_2 \cup k(J_1, J_2)) + \hat{\pi}_p(J_1, J_2),$$

with one notable exception. For the chosen $(H_1, H_2)_{p+1}$, there must exist a $(J_1, J_2)_p$ with $K_1 \subseteq J_1 \subseteq H_1$ and $K_2 \subseteq J_2 \subseteq H_2$: for such a $(J_1, J_2)_p$, select $k(J_1, J_2) = (H_1 \cup H_2) - (J_1 \cup J_2)$ so that by the hypothesis, $\hat{\pi}_{p+1}(H_1, H_2) > 0$. Invoking Lemma 6.2 exactly as in the proof of Theorem 6.2, we have that $\hat{\pi}_{p+1}(J_1, J_2)$ $\forall\ (J_1, J_2)_{p+1}$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \emptyset$ constitutes part of an optimal dual solution to LP(p + 1) satisfying (6.7) (for $d = p + 1$), as desired. Since the level-n hierarchical formulation has all binary extreme points, $F_d(J_1, J_2) = 0$ at all optimal solutions to Problem PP. □

Observe how Theorem 6.3 allows us to recognize instances in which a 0-1 variable realizes a given binary value at all optimal solutions to Problem PP. Let $(\hat{x}, \hat{w})$ and $\hat{\pi}_d$ be, respectively, optimal primal and dual solutions to Problem LP(d). If $|N^f| \leq d$ and there exists some $S \subseteq N$ having $|S| = d$ and a $j \in S$ satisfying $\hat{\pi}_d(J_1, J_2) > 0$ $\forall\ (J_1, J_2)_d$ such that $j \in J_i$, $i = 1$ or 2, and $J_1 \cup J_2 = S$, then Theorem 6.3 and (6.3) enforce that

$$0 = \sum_{\substack{(J_1, J_2)_d \\ J_1 \cup J_2 = S, j \in J_i}} f_d(J_1, J_2) = \begin{cases} x_j & \text{if } i = 1 \\ 1 - x_j & \text{if } i = 2 \end{cases}$$

for all optimal solutions to Problems LP(p), $p \in \{d, ..., n\}$, and at all optimal solutions to Problem PP. If, however, $|N^f| > d$ and there exist multipliers $\bar{\pi}_d(J_1, J_2) \geq 0$ $\forall\ (J_1, J_2)_d$ associated with inequalities (6.5) that satisfy (6.7), then if the conditions given above are true for some subset of these multipliers $\bar{\pi}_d(J_1, J_2)$ (where $\bar{\pi}_d(J_1, J_2)$ is substituted for $\hat{\pi}_d(J_1, J_2)$), the same result holds.

It turns out that the conditions posed in Theorem 6.2 for identifying a solution as persistent are satisfied at all optimal solutions to Problem LP(2), and therefore that Problem IP(2) possesses the persistency property. In addition, it is also true that whenever a variable x_j takes the same binary value in *all* optimal solutions to LP(2), it takes that same binary value in *all* optimal integer solutions. We conclude this section by establishing these two results in Theorem 6.4. First, however, we present Lemma 6.3 which shows that for any feasible solution $(\hat{x}, \hat{w})$ to Problem LP(2), restrictions

(6.6) of this linear program are satisfied strictly. This result is used in the proof of Theorem 6.4.

Lemma 6.3. *Consider any feasible solution* $(\hat{x}, \hat{w})$ *to Problem LP(2). For each* $(J_1, J_2)_2$ *corresponding to an inequality in (6.6), we have that* $f_2(J_1, J_2) > 0$.

Proof. Observe that the linear inequalities $f_2(J_1, J_2) \geq 0$ associated with the sets $(J_1, J_2)_2$ that satisfy the hypothesis are of the form

$$1 - x_i - x_j + w_J \geq 0 \qquad \text{where } i \in N^- \text{ and } j \notin N^+, \tag{6.16}$$

$$x_i - w_J \geq 0 \qquad \text{where } i \in N^+ \text{ and } j \notin N^+ \tag{6.17}$$

$$x_i - w_J \geq 0 \qquad \text{where } i \notin N^- \text{ and } j \in N^- \tag{6.18}$$

$$w_J \geq 0 \qquad \text{where } i \in N^+ \text{ and } j \notin N^- \tag{6.19}$$

where, in each expression, $J = \{i, j\}$. Now, recalling from Lemma 6.1 that $\hat{w}_J = 0$ if $J \cap N^- \neq \emptyset$, and observing that $j \notin N^+$ in (6.16) and that $i \notin N^-$ in (6.18), it follows that inequalities (6.16) and (6.18) are satisfied with strict inequality at the solution $(\hat{x}, \hat{w})$. Similarly, recalling from Lemma 6.1 that $\hat{w}_J = \hat{w}_{J-i}$ if $i \in J \cap N^+$, and observing that $j \notin N^+$ in (6.17) and that $j \notin N^-$ in (6.19), it follows that inequalities (6.17) and (6.19) are satisfied with strict inequality at the solution $(\hat{x}, \hat{w})$. □

Theorem 6.4. *Problem IP(2) possesses the persistency property. Moreover, letting $K^+ \subseteq N$ and $K^- \subseteq N$ denote the index sets of x variables which take values 1 and 0, respectively, at all optimal solutions to Problem LP(2), we have that $x_i = 1 \ \forall \ i \in K^+$ and $x_i = 0 \ \forall \ i \in K^-$ at all optimal solutions to Problems LP(p), $p \in \{2, \ldots, n\}$. Thus, $x_i = 1 \ \forall \ i \in K^+$ and $x_i = 0 \ \forall \ i \in K^-$ at all optimal solutions to Problem PP.*

Proof. Let $(\hat{x}, \hat{w})$ and $\hat{\pi}_2$ be any optimal primal and dual solutions, respectively, to Problem LP(2). If $|N^f| \leq 2$, then $(\hat{x}, \hat{w})$ is a persistent solution to Problem LP(2) by Theorem 6.1. If $|N^f| > 2$, then $(\hat{x}, \hat{w})$ is a persistent solution to Problem LP(2) by case (ii) of Theorem 6.2 since Lemma 6.3 and complementary slackness to LP(2) ensure that $\hat{\pi}_2(J_1, J_2) = 0 \ \forall \ (J_1, J_2)_2$ corresponding to inequalities (6.6). Since $(\hat{x}, \hat{w})$ is an arbitrarily-selected optimal solution to Problem LP(2), Problem IP(2) possesses the persistency property. Next, observe that Lemma 6.1 enforces that $f_2(J_1, J_2) = 0$ $\forall \ (J_1, J_2)_2$ such that $(J_1 \cap K^-) \cup (J_2 \cap K^+) \neq \emptyset$ at *all* optimal solutions to Problem LP(2). By the Strong Theorem of Complementary Slackness, there must therefore exist an optimal dual solution $\tilde{\pi}_2$ to Problem LP(2) having $\tilde{\pi}_2(J_1, J_2) > 0$ $\forall \ (J_1, J_2)_2$ such that $(J_1 \cap K^-) \cup (J_2 \cap K^+) \neq \emptyset$. Since $\tilde{\pi}_2(J_1, J_2) = 0$ $\forall \ (J_1, J_2)_2$ corresponding to inequalities (6.6) from Lemma 6.3 and complementary slackness to LP(2), it follows from dual feasibility of $\tilde{\pi}_2$ to Problem LP(2) and Theorem 6.3 that for all $(J_1, J_2)_2$ with $(J_1 \cap K^-) \cup (J_2 \cap K^+) \neq \emptyset$, (i) for each $p \in \{2, \ldots, n\}$, $f_2(J_1, J_2) = 0$ at *all* optimal solutions to Problem LP(p) and (ii)

$F_2(J_1, J_2) = 0$ at *all* optimal solutions to Problem PP. Invoking (6.3) as in the discussion following the proof of Theorem 6.3, we have for each $i \in K^+$ that

$$0 = \sum_{\substack{(J_1, J_2)_2 \\ i \in J_2}} f_2(J_1, J_2) = (n-1)(1-x_i)$$

and for each $i \in K^-$ that

$$0 = \sum_{\substack{(J_1, J_2)_2 \\ i \in J_1}} f_2(J_1, J_2) = (n-1)x_i$$

at *all* optimal solutions to Problems LP(p), $p \in \{2, ..., n\}$. Consequently, we have that $x_i = 1 \ \forall \ i \in K^+$ and $x_i = 0 \ \forall \ i \in K^-$ at *all* optimal solutions to Problem PP, and the proof is complete. □

6.2. RLT-Based Persistency for Constrained 0-1 Polynomial Programs

The persistency results of the previous section focused entirely upon unconstrained 0-1 polynomial programs. In this section, we broaden our study by showing how this work can be used to motivate and prove persistency results for certain *constrained* such programs. In particular, we introduce a class of these programs and present, for each problem in this class, a hierarchy of equivalent mixed 0-1 linear programs. This hierarchy is obtained by including within Problems IP(d), $d \in \{2, ..., n\}$, equality restrictions which, relative to the continuous relaxations, restrict attention to faces of the respective feasible regions. By suitably modifying the arguments of the previous section, we are able to extend the results to include this constrained hierarchy of linearizations and, consequently, the underlying family of restricted problems.

We begin by considering restricted 0-1 polynomial programs for which the constraints are of the form $F_r(J_1, J_2) = 0 \ \forall \ (J_1, J_2)_r \in E$, where $E \subseteq \{(J_1, J_2)_p$ for $p \in \{2, \ldots, n\}$. Here, we assume that no restriction is of the type $F_1(J_1, J_2) = 0$ since otherwise the problem can be reduced by fixing the corresponding variable x_j at 0 or 1, according as $j \in J_1$ or $j \in J_2$, respectively. We also assume without loss of generality that there do not exist two sets $(J_1, J_2)_p \in E$ and $(K_1, K_2)_q \in E$ with $J_1 \subseteq K_1$ and $J_2 \subseteq K_2$ since any x satisfying $F_p(J_1, J_2) = 0$ would necessarily satisfy $F_q(K_1, K_2) = 0$. Specifically, the general form of the problem is as follows.

CPP: Minimize $\sum_{J \subseteq N} c_J \prod_{j \in J} x_j$

subject to $F_r(J_1, J_2) = 0 \qquad \forall \ (J_1, J_2)_r \in E$ (6.20)

x binary.

Problem CPP is a restricted (constrained) version of Problem PP in that Equations (6.20) are not present in PP. Consistent with the earlier notation, we let t denote the maximum degree of any polynomial term appearing with nonzero coefficient, either in the objective function or the constraints. We then define the restricted analog to Problem IP(d) for each $d \in \{t, \ldots, n\}$ as the following mixed 0-1 linear program.

RIP(d): Minimize $\sum_{J \subseteq N} c_J w_J$

$$\text{subject to} \quad f_d(J_1, J_2) \geq 0 \qquad \forall\ (J_1, J_2)_d \notin E_d \tag{6.21}$$

$$f_d(J_1, J_2) = 0 \qquad \forall\ (J_1, J_2)_d \in E_d \tag{6.22}$$

$$x \text{ binary}$$

where $E_d = \{(J_1, J_2)_d$ such that there exists a $(K_1, K_2)_r \in E$ with $K_1 \subseteq J_1$ and $K_2 \subseteq J_2\}$. Problem RIP(d) is obtained from Problem CPP by including the restrictions (6.21), which constitute a subset of (6.1), within the constraint set, replacing equations (6.20) with (6.22), and linearizing the objective function by substituting for each product term $\prod_{j \in J} x_j$ the continuous variable w_J. Denote, for each $d \in \{t, ..., n\}$, the continuous relaxation of Problem RIP(d) obtained by omitting the x binary restrictions as Problem RLP(d).

Given a 0-1 polynomial program of the form of Problem CPP, for each $d \in \{t, ..., n\}$, the linear reformulation RIP(d) is equivalent to CPP in that for any feasible solution to one problem there exists a feasible solution to the other problem with the same objective function value. Recall from the previous section (and Lemma 2.3) that every (x, w), x binary, which satisfies $f_d(J_1, J_2) \geq 0\ \forall\ (J_1, J_2)_d$ must have $w_J = \prod_{j \in J} x_j\ \forall\ J \subseteq N$ such that $|J| \geq 2$. Thus, $F_r(J_1, J_2) = f_r(J_1, J_2)\ \forall\ (J_1, J_2)_r \in E$ at every point (x, w) feasible to Problem RIP(d). Since for each $(J_1, J_2)_r \in E$ and for any $S \subseteq N - (J_1 \cup J_2)$ with $|S| = d - r$, (6.3) enforces

$$f_r(J_1, J_2) = \sum_{\substack{(S_1, S_2)_{d-r} \\ S_1 \cup S_2 = S}} f_d(J_1 \cup S_1, J_2 \cup S_2),$$

we have that at every such point (x, w), equations (6.20) must be satisfied by x. On the other hand, if a given binary vector $\hat{x}$ is feasible to (6.20), then $(\hat{x}, \hat{w})$ with $\hat{w}_J = \prod_{j \in J} \hat{x}_j \; \forall \; J \subseteq N$ such that $|J| \geq 2$ by definition satisfies $f_d(J_1, J_2) \geq 0 \; \forall \; (J_1, J_2)_d$ and thus satisfies (6.24). Once again using (6.3) and $f_d(J_1, J_2) \geq 0 \; \forall \; (J_1, J_2)_d$, we have that $(\hat{x}, \hat{w})$ satisfies (6.22). Moreover, at every point $(\hat{x}, \hat{w})$ satisfying $\hat{w}_J = \prod_{j \in J} \hat{x}_j \; \forall \; J \subseteq N$ it trivially follows that $\sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j = \sum_{J \subseteq N} c_J \hat{w}_J$.

We now focus attention on extending the persistency results to include problems of the form of RIP(d), $d \in \{t, ..., n\}$, and hence problems of the form of Problem CPP. As the results closely relate to those of the previous section, we suffix each Theorem and Lemma with the letter C to denote the constrained analog of the corresponding result of Section 6.1. We begin by presenting Theorem 6.1C below.

Theorem 6.1C. *Let $(\hat{x}, \hat{w})$ be an optimal primal solution to Problem RLP(d), and define subsets N^+, N^-, and N^f of N in terms of $\hat{x}$ as $N^+ = \{j : \hat{x}_j = 1\}$, $N^- = \{j : \hat{x}_j = 0\}$, and $N^f = N - (N^+ \cup N^-)$. If $|N^f| \leq d$, then $(\hat{x}, \hat{w})$ is a persistent solution to Problem RLP(d). Consequently, there exists an optimal solution x^* to Problem CPP having $x^*_j = 1 \; \forall \; j \in N^+$ and $x^*_j = 0 \; \forall \; j \in N^-$.*

Proof. The proof follows that of Theorem 6.1 exactly, once we note that the feasible region to Problem RLP(n) constitutes a face of that of LP(n), and thus has all binary extreme points. □

A direct consequence of Theorem 6.1C is that Problems RIP(n) and RIP($n-1$) possess the persistency property since, for these problems, if $N^+ \cup N^- \neq \varnothing$, then $\left|N^f\right| \leq d$. In order to extend the remainder of the persistency results to include Problems RIP(d), $d \in \{t, \ldots, n-2\}$, we present Lemma 6.2C, which exploits the special structure of the constraint sets. For Lemma 6.2C, as well as the upcoming Theorems 6.2C, 6.3C, 6.4C, and 6.5, we consider only feasible instances of Problem CPP. This assures us that the relaxations RLP(d), $d \in \{t, \ldots, n\}$, are also necessarily feasible.

Lemma 6.2C. *Consider any feasible instance of Problem CPP and any partition of the set N into three subsets N^+, N^-, and N^f, with the corresponding partition of the constraints of Problem RLP(d) as follows.*

$$f_d(J_1, J_2) \geq 0 \quad \forall\ (J_1, J_2)_d \notin E_d \text{ such that } J_1 \cup J_2 \subseteq N^f \tag{6.23}$$

$$f_d(J_1, J_2) = 0 \quad \forall\ (J_1, J_2)_d \in E_d \text{ such that } J_1 \cup J_2 \subseteq N^f \tag{6.24}$$

$$f_d(J_1, J_2) \geq 0 \quad \forall\ (J_1, J_2)_d \notin E_d \text{ such that } (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing \tag{6.25}$$

$$f_d(J_1, J_2) = 0 \quad \forall\ (J_1, J_2)_d \in E_d \text{ such that } (J_1 \cap N^-) \cup (J_2 \cap N^+) \neq \varnothing \tag{6.26}$$

$$f_d(J_1, J_2) \geq 0 \quad \forall\ (J_1, J_2)_d \notin E_d \text{ such that } J_1 \cup J_2 \not\subseteq N^f \text{ and } (J_1 \cap N^-) \cup (J_2 \cap N^+) = \varnothing \tag{6.27}$$

$$f_d(J_1, J_2) = 0 \quad \forall\ (J_1, J_2)_d \in E_d \text{ such that } J_1 \cup J_2 \not\subseteq N^f \text{ and } (J_1 \cap N^-) \cup (J_2 \cap N^+) = \varnothing \tag{6.28}$$

If $E' \equiv \{(J_1, J_2)_r \in E : J_1 \cup J_2 \not\subseteq N^f$ *and* $(J_1 \cap N^-) \cup (J_2 \cap N^+) = \varnothing\}$ *is empty and there exist multipliers* $\hat{\pi}_d(J_1, J_2)\ \forall\ (J_1, J_2)_d$ *corresponding to (6.25) and (6.26) with* $\hat{\pi}_d(J_1, J_2) \geq 0$ *for all such* $(J_1, J_2)_d \notin E_d$, *and these multipliers satisfy (6.7), then there exists an optimal primal solution* $(\tilde{x}, \tilde{w})$ *to Problem RLP(d) having* $\tilde{x}_j = 1\ \forall\ j \in N^+$ *and* $\tilde{x}_j = 0\ \forall\ j \in N^-$. *Furthermore, if* $|N^f| \geq d$, *then there exists an optimal dual solution* $\tilde{\pi}_d$ *for which* $\tilde{\pi}_d(J_1, J_2) = 0\ \forall\ (J_1, J_2)_d$ *associated with (6.27) and (6.28) and* $\tilde{\pi}_d(J_1, J_2) = \hat{\pi}_d(J_1, J_2)\ \forall\ (J_1, J_2)_d$ *associated with (6.25) and (6.26).*

Proof. The proof follows that of Lemma 6.2, where (6.23) and (6.24) replace (6.4), (6.25) and (6.26) replace (6.5), and (6.27) and (6.28) replace (6.6), given that the prescribed set of constraints E' is empty. Here, the multipliers corresponding to Equations (6.26) are not restricted to be nonnegative. The only portion warranting explanation is that, given $E' = \varnothing$, the primal solution $(\tilde{x}, \tilde{w})$ defined by (6.9), (6.10), and (6.11) in terms of an optimal solution to the modified (6.8) and the sets N^+ and N^- satisfies (6.28). Toward this end, consider any $(J_1, J_2)_d$ associated with an equation in (6.28). (We assume here that $|N^f| \geq d$ since the case where $|N^f| < d$ is then obvious.) Observe that by (6.13), $f_d(J_1, J_2) = f_q(J_1 - N^+, J_2 - N^-)$ when evaluated at $(\tilde{x}, \tilde{w})$, where $q = d - |(J_1 \cap N^+) \cup (J_2 \cap N^-)| < d$. Consider any set $S \subseteq N^f - ((J_1 - N^+) \cup (J_2 - N^-))$ with $|S| = d - q$ and recall from (6.3) that

$$\sum_{\substack{(S_1, S_2)_{d-q} \\ S_1 \cup S_2 = S}} f_d((J_1 - N^+) \cup S_1, (J_2 - N^-) \cup S_2) = f_q(J_1 - N^+, J_2 - N^-).$$

Since by definition, $((J_1 - N^+) \cup S_1,\ (J_2 - N^-) \cup S_2) \in E_d\ \forall\ (S_1, S_2)_{d-q}$ and since $((J_1 - N^+) \cup (J_2 - N^-) \cup S) \subseteq N^f$, $f_d(J_1, J_2) = 0$ at $(\tilde{x}, \tilde{w})$ is implied by (6.24). This completes the proof. □

Theorem 6.1C and Lemma 6.2C motivate an inductive persistency argument for Problems RIP(d), $d \in \{t, ..., n\}$, in exactly the same manner that Theorem 6.1 and Lemma 6.2 approached Problems IP(d). Suppose we solve Problem RLP(d) for some $d \in \{t, ..., n\}$ to obtain an optimal primal solution $(\hat{x}, \hat{w})$. By Theorem 6.1C, if $|N^f| \le d$, then $(\hat{x}, \hat{w})$ is a persistent solution to Problem RLP(d). Otherwise, $|N^f| > d$. In this case, if the set E' is empty and there exists a dual feasible solution $\hat{\pi}_d$ to Problem RLP(d) with the multipliers corresponding to constraints (6.27) and (6.28) all taking value 0, by Lemma 6.2C the task of identifying $(\hat{x}, \hat{w})$ as persistent reduces to establishing, for $p = d + 1$, the existence of multipliers $\tilde{\pi}_p(J_1, J_2)\ \forall\ (J_1, J_2)_p$ such that $(J_1 \cap N^-) \cup (J_2 \cap N^+) \ne \varnothing$ with $\tilde{\pi}_p(J_1, J_2) \ge 0$ for all such $(J_1, J_2)_p \notin E_p$ satisfying (6.15). Additionally, Theorems 6.3 and 6.4 both readily extend to accommodate the constrained case. The persistency results of the previous section, with minor modifications, thus all follow for Problems CPP and RIP(d), $d \in \{t, ..., n\}$, and are presented below without formal proofs. For rigorous proofs of these results on equivalent formulations, the interested reader is referred to Lassiter (1993).

Theorem 6.2C. *Consider any feasible instance of Problem CPP, and let* $(\hat{x}, \hat{w})$ *be an optimal primal solution to Problem RLP(d). Define the subsets* N^+, N^-, *and* N^f *of* N

in terms of $\hat{x}$ *as in Theorem 6.1C, and partition the constraints of Problem RLP(d) exactly as in Lemma 6.2C. If either (i)* $|N^f| \le d$ *or (ii)* $|N^f| > d$, *the set* E' *is empty, and there exists a dual feasible solution* $\hat{\pi}_d$ *to Problem RLP(d) with the multipliers corresponding to constraints (6.27) and (6.28) all taking value 0, then* $(\hat{x}, \hat{w})$ *is a persistent solution to Problem RLP(d). Consequently, there exists an optimal solution* x^* *to Problem CPP having* $x_j^* = 1 \ \forall \ j \in N^+$ *and* $x_j^* = 0 \ \forall \ j \in N^-$.

Theorem 6.3C. *Consider any feasible instance of Problem CPP, and let* $(\hat{x}, \hat{w})$ *and* $\hat{\pi}_d$ *be optimal primal and dual solutions, respectively, to Problem RLP(d). If* $|N^f| \le d$, *then for each* $p \in \{d, ..., n\}$ *and for every* $(J_1, J_2)_d$ *such that* $\hat{\pi}_d(J_1, J_2) > 0$, *we have that* $f_d(J_1, J_2) = 0$ *at all optimal solutions to Problem RLP(p). Consequently, for every such* $(J_1, J_2)_d$, $F_d(J_1, J_2) = 0$ *at all optimal solutions to Problem CPP. If* $|N^f| > d$, *the set* E' *is empty, and there exist multipliers* $\tilde{\pi}_d(J_1, J_2) \ \forall \ (J_1, J_2)_d$ *corresponding to constraints (6.25) and (6.26) with* $\tilde{\pi}_d(J_1, J_2) \ge 0$ *for all such* $(J_1, J_2)_d \notin E_d$, *and these multipliers satisfy (6.7), then for each* $p \in \{d, ..., n\}$ *and for every* $(J_1, J_2)_d \notin E_d$ *associated with an inequality in (6.25) for which* $\tilde{\pi}_d(J_1, J_2) > 0$, *it follows that* $f_d(J_1, J_2) = 0$ *at all optimal solutions to Problem RLP(p). Thus, for each such* $(J_1, J_2)_d$, $F_d(J_1, J_2) = 0$ *at all optimal solutions to Problem CPP.*

Theorem 6.4C. *Provided it is feasible, Problem RIP(2) possesses the persistency property. Moreover, letting* $K^+ \subseteq N$ *and* $K^- \subseteq N$ *denote the index sets of x variables which take values 1 and 0, respectively, at all optimal solutions to Problem RLP(2), we* *have that* $x_i = 1 \ \forall\ i \in K^+$ *and* $x_i = 0 \ \forall\ i \in K^-$ *at all optimal solutions to Problems RLP(p),* $p \in \{2, ..., n\}$. *Thus,* $x_i = 1 \ \forall\ i \in K^+$ *and* $x_i = 0 \ \forall\ i \in K^-$ *at all optimal solutions to Problem CPP.*

Two remarks are in order relative to Theorem 6.4C. First, we included the requirement that Problem RIP(2) be feasible since it is possible for RLP(2) to be feasible and RIP(2) infeasible. A simple example has $n = 3$ and $E = E_2 = \{(\emptyset,\{1,2\}), (\emptyset,\{1,3\}), (\emptyset,\{2,3\}), (\{1,2\},\emptyset), (\{1,3\},\emptyset), (\{2,3\},\emptyset)\}$. Here, $(\hat{x}, \hat{w})$ defined as $\hat{x}_1 = \hat{x}_2 = \hat{x}_3 = \frac{1}{2}$ and $\hat{w}_{12} = \hat{w}_{13} = \hat{w}_{23} = 0$ satisfies the restrictions of RLP(2), though RIP(2) has no solution. Second, we did not consider the set E' since it is necessarily empty at all feasible solutions to RLP(2).

We have to this point demonstrated how the persistency results of the previous section can be modified to accommodate Problems RLP(d), $d \in \{t, ..., n\}$. The proofs of these results, however, require the additional condition that the set E', defined in terms of the pertinent solution $(\hat{x}, \hat{w})$, be empty. The question arises as to whether we can relax this additional condition. In the example below we show that Theorem 6.2C, our fundamental result, fails in the absence of this condition.

Example 6.4. Consider the instance of Problem CPP having degree $t = 3$ in $n = 5$ binary variables given below.

Minimize $\quad x_3 - x_1x_3 + x_2x_4 + x_1x_2x_3 - x_1x_2x_4 + x_1x_3x_4 - x_2x_3x_5 - x_3x_4x_5$

subject to $\quad F_3(\{1\}, \{2, 3\}) = 0 \qquad F_3(\{3\}, \{1, 2\}) = 0$

$$
\begin{array}{ll}
F_3(\{1,2\},\{4\}) = 0 & F_3(\varnothing,\{1,2,4\}) = 0 \\
F_3(\{3\},\{1,4\}) = 0 & F_3(\{1,4,5\},\varnothing) = 0 \\
F_3(\{3\},\{2,4\}) = 0 & F_3(\{2\},\{3,4\}) = 0 \\
F_3(\{4\},\{2,3\}) = 0 & F_3(\{2,3,4\},\varnothing) = 0 \\
x_i \text{ binary} & i = 1,\ldots,5.
\end{array}
$$

Hence, $E = E_3 = \{(\{1\},\{2,3\}), (\{1,2\},\{4\}), (\{3\},\{1,4\}), (\{3\},\{2,4\}), (\{4\},\{2,3\}), (\{3\},\{1,2\}), (\varnothing,\{1,2,4\}), (\{1,4,5\},\varnothing), (\{2\},\{3,4\}), (\{2,3,4\},\varnothing)\}$. The optimal objective function value to Problem RLP(3) is $-\frac{1}{3}$, with optimal primal and dual solutions as listed below.

Primal Solution. $(x_1, x_2, x_3, x_4, x_5) = (\frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{1}{3}, 1)$, $w_{35} = \frac{2}{3}$, $w_{13} = w_{15} = w_{23} = w_{25} = w_{34} = w_{45} = w_{135} = w_{235} = w_{345} = \frac{1}{3}$, and all other primal variables take value 0.

Dual Solution. $\pi_3(\{1,2\},\{3\}) = \pi_3(\{2,4\},\{1\}) = \pi_3(\{1\},\{3,4\}) = \pi_3(\{1,3,4\},\varnothing) = \frac{1}{3}$, $\pi_3(\{2\},\{1,3\}) = \pi_3(\{2,3\},\{5\}) = \pi_3(\{3,4\},\{5\}) = 1$, $\pi_3(\{2\},\{3,4\}) = -\frac{2}{3}$, $\pi_3(\varnothing,\{1,2,4\}) = \pi_3(\{1,2\},\{4\}) = \pi_3(\{3\},\{1,2\}) = \pi_3(\{3\},\{2,4\}) = \pi_3(\{4\},\{2,3\}) = \pi_3(\{3\},\{1,4\}) = \frac{1}{3}$, and all other dual variables take value 0.

For this example, $N^+ = \{5\}$ and $N^- = \varnothing$, with $|N^f| = 4 > 3 = d$. Note that even though the dual multipliers corresponding to constraints (6.27) and (6.28) all take value 0, the conditions of Theorem 6.2C are not satisfied since $(\{1,4,5\},\varnothing) \in E'$. The unique optimal solution to this problem is $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 1, 0)$ with objective function value 0.

Throughout, our persistency results for constrained 0-1 programs have been based upon the mixed 0-1 linear formulations RIP(d), $d \in \{t, ..., n\}$, defined in a higher-dimensional space than the original problem. We now focus attention on a family of 0-1 linear programs. To begin, we present below two related problems: the first is a special instance of Problem RIP(2), referred to as Problem RIP$'$(2) in which the objective function is linear in the decision variables x, and the second is a specially-structured 0-1 linear program defined in terms of the first problem. (The continuous relaxation of RIP$'$(2) will be denoted as RIP$'$(2).)

$$
\begin{array}{llll}
\textbf{RIP}'\ \textbf{(2):} & \text{Minimize} & \sum_{i=1}^{n} c_i x_i & \\
& \text{subject to} & f_2(J_1, J_2) \geq 0 \quad \forall\ (J_1, J_2)_2 & (6.29) \\
& & f_2(\varnothing, \{i, j\}) = 0 \quad \forall\ (i, j) \in I_1 & (6.30) \\
& & f_2(\{i\}, \{j\}) = 0 \quad \forall\ (i, j) \in I_2 & (6.31) \\
& & f_2(\{i, j\}, \varnothing) = 0 \quad \forall\ (i, j) \in I_3 & (6.32) \\
& & x_i \text{ binary} \quad \forall\ i \in N. &
\end{array}
$$

Here, the sets I_1, I_2, and I_3 are defined, using obvious notation, in terms of the set E, and the implied restrictions $f_2(J_1, J_2) \geq 0\ \ \forall\ (J_1, J_2)_2 \in E_2$ are included in (6.29) for convenience. Consider now the constrained 0-1 linear problem below having the same objective function and the same sets I_1, I_2, and I_3.

C01: Minimize $\displaystyle\sum_{i=1}^{n} c_i x_i$

subject to

$$x_i + x_j \geq 1 \quad \forall\ (i, j) \in I_1 \tag{6.33}$$

$$x_i \leq x_j \quad \forall\ (i, j) \in I_2 \tag{6.34}$$

$$x_i + x_j \leq 1 \quad \forall\ (i, j) \in I_3 \tag{6.35}$$

$$0 \leq x_i \leq 1 \quad \forall\ i \in N \tag{6.36}$$

$$x_i \text{ binary} \quad \forall\ i \in N.$$

Denote as Problem LC01 the continuous relaxation of C01 obtained by deleting the x binary restrictions.

Problems RIP′(2) and LC01 (and consequently RIP′(2) and C01) are related, as demonstrated below, in the sense that the feasible region to the latter problem is the projection of the feasible region to the former onto the x-variable space. In light of this observation, and since the objective function coefficients on all variables w_J with $|J| = 2$ have value 0 in Problem RIP′(2), it follows that a point $\hat{x}$ is part of an optimal solution $(\hat{x}, \hat{w})$ to RIP′(2) if and only if $\hat{x}$ is optimal to LC01. Moreover, this optimality relationship holds for Problems RIP′(2) and C01. Consequently, provided it is feasible, by Theorem 6.4C, Problem C01 possesses the persistency property, and any x variable which takes value 1 (0) at *all* optimal solutions to LC01 realizes that same value at *all* optimal solutions to Problem C01. A formal statement and proof of these persistency results are given below.

Theorem 6.5. *Provided it is feasible, Problem C01 possesses the persistency property. Moreover, letting $K^+ \subseteq N$ and $K^- \subseteq N$ denote the index sets of x variables which take values 1 and 0, respectively, at all optimal solutions to Problem LC01, we have that $x_i = 1 \; \forall \; i \in K^+$ and $x_i = 0 \; \forall \; i \in K^-$ at all optimal solutions to Problem C01.*

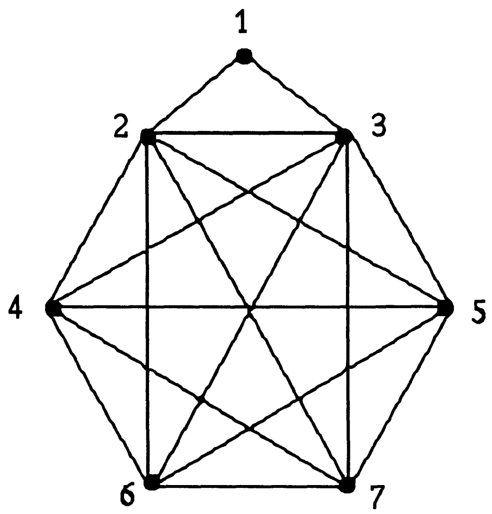
Proof. Since the objective functions to Problems C01 and RIP′(2) are identical and expressed entirely in terms of the variables x, and since the result is true for RIP′(2) by Theorem 6.4C, it is sufficient to show that a point $\hat{x}$ is part of a feasible solution $(\hat{x}, \hat{w})$ to RLP′(2) if and only if $\hat{x}$ is feasible to LC01. Thus, let $(\hat{x}, \hat{w})$ be feasible to Problem RLP′(2). For each $(i, j) \in I_1$, $f_2(\{i, j\}, \emptyset) \geq 0$ of (6.29) and $f_2(\emptyset, \{i, j\}) = 0$ of (6.30) imply that $\hat{x}_i + \hat{x}_j \geq 1$ while for each $(i, j) \in I_2$, $f_2(\{j\}, \{i\}) \geq 0$ of (6.29) and $f_2(\{i\}, \{j\}) = 0$ of (6.31) imply that $\hat{x}_i \leq \hat{x}_j$. Finally, for each $(i, j) \in I_3$, $f_2(\emptyset, \{i, j\}) \geq 0$ of (6.29) and $f_2(\{i, j\}, \emptyset) = 0$ of (6.32) imply that $\hat{x}_i \leq \hat{x}_j \leq 1$. Hence, constraints (6.33) through (6.35) of LC01 are satisfied. Lemma 2.1 has that (6.36) is implied by (6.29) so $\hat{x}$ is feasible to LC01. Conversely, let $\hat{x}$ be feasible to LC01 and define for each $J \subseteq N$ with $|J| = 2$, $\hat{w}_J$ where $J = \{i, j\}$, as $\hat{w}_J = \hat{x}_i + \hat{x}_j - 1$ if $(i, j) \in I_1$, $\hat{w}_J = \hat{x}_i$ if $(i, j) \in I_2$, $\hat{w}_J = 0$ if $(i, j) \in I_3$, and $\hat{w}_J = \hat{x}_i \hat{x}_j$ if $(i, j) \notin I_1 \cup I_2 \cup I_3$. It can be readily verified that $(\hat{x}, \hat{w})$ with $\hat{w}$ so defined is feasible to RLP′(2). □

As is discussed in Section 6.4, Nemhauser and Trotter (1975) showed that a classical formulation of the vertex packing problem possesses the persistency property. Given a graph $G = (V, \mathcal{E})$ with vertex set $V = \{v_1, v_2, ..., v_n\}$ and edge set $\mathcal{E}$, and a weight c_k associated with each vertex v_k, $k = 1, ..., n$, the vertex packing problem is to select a set of vertices that are pairwise nonadjacent for which the sum of the weights is a maximum. The formulation of concern is as follows.

VPP: Maximize $\{\sum_{i=1}^{n} c_i x_i : x_i + x_j \leq 1 \ \forall \ (i, j) \in \mathcal{E}, \ x_i \text{ binary} \ \forall \ i = 1, ..., n\}$

where for each $i = 1, ..., n$, $x_i = 1$ if vertex v_i is selected as part of a packing and $x_i = 0$ otherwise. Clearly, Problem VPP is a special case of Problem C01 in which $I_1 = I_2 = \varnothing$, so that Nemhauser and Trotter's result is a special case of the first statement in Theorem 6.5. The persistency results found within this section, however, can be more useful for fixing variables of this problem at binary values than Problem VPP since our study of the hierarchy of linearizations permits us to look at a richer class of formulations. We conclude this section with an example illustrating this usefulness.

Example 6.5. Consider the vertex packing problem on $n = 7$ vertices below, where we wish to select the maximum number of vertices such that no two are connected by an edge.



The classical formulation, Problem VPP, is given below.

Maximize $\sum_{i=1}^{7} x_i$

subject to $x_1 + x_2 \le 1$ $x_1 + x_3 \le 1$ $x_2 + x_3 \le 1$ $x_2 + x_4 \le 1$ $x_2 + x_5 \le 1$

$x_2 + x_6 \le 1$ $x_2 + x_7 \le 1$ $x_3 + x_4 \le 1$ $x_3 + x_5 \le 1$ $x_3 + x_6 \le 1$

$x_3 + x_7 \le 1$ $x_4 + x_5 \le 1$ $x_4 + x_6 \le 1$ $x_4 + x_7 \le 1$ $x_5 + x_6 \le 1$

$x_5 + x_7 \le 1$ $x_6 + x_7 \le 1$

x_i binary $\quad i = 1, \ldots, 7.$

Solving the continuous relaxation of the above problem, which by the proof of Theorem 6.5 is equivalent to solving Problem RLP(2) with $E = \{(\{1,2\},\emptyset), (\{1,3\},\emptyset), (\{2,3\},\emptyset), (\{2,4\},\emptyset), (\{2,5\},\emptyset), (\{2,6\},\emptyset), (\{2,7\},\emptyset), (\{3,4\},\emptyset), (\{3,5\},\emptyset), (\{3,6\},\emptyset), (\{3,7\},\emptyset),$

$(\{4,5\},\varnothing)$, $(\{4,6\},\varnothing)$, $(\{4,7\},\varnothing)$, $(\{5,6\},\varnothing)$, $(\{5,7\},\varnothing)$, $(\{6,7\},\varnothing)\}$, we obtain the unique optimal primal solution $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ with optimal objective function value $\frac{7}{2}$. Persistency does not allow us to fix any variables at binary values. However, solving Problem RLP(3), we obtain an optimal objective function value of $\frac{7}{3}$ along with optimal primal and dual solutions given below. (Since all theorems have been presented in terms of minimization problems, we look at the equivalent problem of minimizing $-\sum_{i=1}^{I} x_i$ and define the dual multipliers accordingly.)

Primal Solution. $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (1, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $w_{14} = w_{15} = w_{16} = w_{17} = \frac{1}{3}$, and all other primal variables take value 0.

Dual Solution. $\pi_3(\varnothing,\{1,2,3\}) = 1$, $\pi_3(\{1,2,3\},\varnothing) = -2$, $\pi_3(\varnothing,\{4,5,6\}) = \pi_3(\varnothing,\{4,5,7\}) = \pi_3(\varnothing,\{4,6,7\}) = \pi_3(\varnothing,\{5,6,7\}) = \pi_3(\{5,6,7\},\varnothing) = \frac{1}{3}$, $\pi_3(\{4,5,6\},\varnothing) = \pi_3(\{4,5,7\},\varnothing) = \pi_3(\{4,6,7\},\varnothing) = \pi_3(\{1,2\},\{3\}) = \pi_3(\{1,3\},\{2\}) = \pi_3(\{2,3\},\{1\}) = -1$, $\pi_3(\{4,5\},\{6\}) = \pi_3(\{4,6\},\{7\}) = \pi_3(\{4,7\},\{5\}) = \pi_3(\{5,6\},\{4\}) = \pi_3(\{5,7\},\{4\}) = \pi_3(\{6,7\},\{4\}) = -\frac{2}{3}$, and all other dual variables take value 0.

Since $N^+ = \{1\}$, $N^- = \{2,3\}$, and $N^f = \{4,5,6,7\}$, we have that $E' = \varnothing$, and that $\pi_3(J_1, J_2) = 0 \quad \forall\ (J_1, J_2)_3$ such that $(J_1 \cup J_2) \not\subseteq \{4,5,6,7\}$ and $(J_1 \cap \{2, 3\}) \cup (J_2 \cap \{1\}) = \varnothing$. Therefore, as $|N^f| = 4 > 3 = d$, the conditions of case (ii) of Theorem 6.2C are satisfied (since Problem VPP is clearly feasible) and we can conclude that there must exist an optimal binary solution having $x_1 = 1$ and $x_2 = x_3 = 0$. It can be readily verified that the optimal objective function value to this problem is 2 with alternate optimal solutions $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) =$ (1,0,0,1,0,0,0), (1,0,0,0,1,0,0), (1,0,0,0,0,1,0), and (1,0,0,0,0,0,1).

6.3. A Modified RLT Approach

As demonstrated in Section 6.1, the hierarchy of linearizations available from applying the RLT to unconstrained 0-1 polynomial programs allows us to establish, for the different levels, conditions sufficient for recognizing a computed solution as persistent. As a result of this study, and by examining faces of the resulting polytopes, we were able in Section 6.2 to identify certain persistent families of 0-1 linear programs. Our most general such formulation is given by Problem RIP(2), so that we are able to produce a persistent linear reformulation of Problems PP and CPP when the degree $t = 2$. However, for those instances in which $t \geq 3$, we do not have a persistent formulation: the structures admitted by the RLT do not permit such results for either Problems IP(d) or Problems RIP(d), $d \in \{3, ..., n - 2\}$.

Bearing in mind that the RLT does not yield persistent linear reformulations of Problem PP when $t \geq 3$, we devise in this section an alternate linearization method that gives rise to such persistent programs for *any* degree t. We use the same basic constructs of *reformulation* and *linearization*, first reformulating the problem in a higher-variable space and then linearizing the resulting problem by substituting a continuous variable for each distinct product term. Here, however, the nonlinear products take a completely different form from those introduced in Chapter 2. (We postpone to Subection 6.3.2 the treatment of constrained 0-1 polynomial programs.)

For convenience, we restate Problem PP as

PP: Minimize $\sum_{J \subseteq N} c_J \prod_{j \in J} x_j$

x binary

where, as before, $N = \{1, 2, ..., n\}$, $x = (x_1, x_2, ..., x_n)$ is the vector of binary decision variables, and where for each $J \subseteq N$, c_J is the objective function coefficient on the term $\prod_{j \in J} x_j$. Once again, we refer to the degree of Problem PP as t, and assume that $t \geq 2$. We find it useful to adopt the notation that $c_{\varnothing} = 0$ and $\prod_{j \in \varnothing} x_j = 1$, with $\prod_{j \in \varnothing} (1 - x_j) = 1$.

Intuitively speaking, our linear reformulations of Problem PP must possess two key ingredients. First, each such formulation must be equivalent to Problem PP in that an optimal solution to either problem readily admits an optimal solution to the other. Second, the constraint structure of the linear problem must be such that every optimal solution to the continuous relaxation is a persistent solution. In order to acquire formulations that capture these two ingredients, we introduce the following modified RLT for Problem PP.

Reformulation Step: For each $d \in \{2, ..., t\}$ define the 4-tuple (J_1, J_2, J_3, J_4) to be of order d, denoted $(J_1, J_2, J_3, J_4)_d$, if (a) $J_1 \cup J_2 \cup J_3 \cup J_4 \subseteq N$, (b)

$(J_1 \cup J_3) \cap (J_2 \cup J_4) = \emptyset$, (c) $|J_1 \cup J_2 \cup J_3 \cup J_4| = d$, and (d) $J_1 \cup J_2 \neq \emptyset$ and $J_3 \cup J_4 \neq \emptyset$. (Only the first and third conditions are needed, as the other two will simply serve to eliminate implied inequalities.) Compute the following products of binary variables

$$F_d(J_1, J_2, J_3, J_4) = [1 - \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j)]\ [1 - \prod_{j \in J_3} x_j \prod_{j \in J_4} (1 - x_j)]$$

$$\forall\ (J_1, J_2, J_3, J_4)_d,\ d = 2, \ldots, t$$

and note that for all binary x, we have

$$F_d(J_1, J_2, J_3, J_4) \geq 0 \quad \forall\ (J_1, J_2, J_3, J_4)_d,\ \ d = 2, \ldots, t, \tag{6.37}$$

Include within Problem PP all inequalities found in (6.37).

Linearization Step: Define for each $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, \ldots, t\}$, the expression $f_d(J_1, J_2, J_3, J_4)$ to be the linearized form of $F_d(J_1, J_2, J_3, J_4)$ obtained by substituting, as in (2.4a), the continuous variable w_J for each occurrence of the product term $\prod_{j \in J} x_j$ for all J such that $2 \leq |J| \leq t$, keeping in mind that $x_i^2 = x_i$ for all i. As in (2.4b), we let $w_j = x_j$ for all j and, consistent with our definition of $\prod_{j \in \emptyset} x_j$, $w_\emptyset = 1$. Perform the same substitution of variables in the objective function as was applied to the expressions $F_d(J_1, J_2, J_3, J_4)$ to obtain $f_d(J_1, J_2, J_3, J_4)$, to arrive at the following mixed 0-1 linear program.

BP: Minimize $\displaystyle\sum_{J \subseteq N} c_J w_J$

subject to $\quad f_d(J_1, J_2, J_3, J_4) \geq 0 \ \forall \ (J_1, J_2, J_3, J_4)_d, \ d = 2, \ldots, t \qquad (6.38)$

x binary.

We denote the continuous relaxation of Problem BP obtained by omitting the x binary restrictions as Problem BLP. We refer to the inequalities $f_d(J_1, J_2, J_3, J_4) \geq 0$ for all $(J_1, J_2, J_3, J_4)_d$, for any given value of $d \in \{2, \ldots, t\}$, as the level-d restrictions. We also adopt the notation that, given any polynomial $[\cdot]$ in 0-1 variables, the expression $[\cdot]_L$ represents the linearization of $[\cdot]$ analogous to that used on $F_d(J_1, J_2, J_3, J_4)$ to obtain $f_d(J_1, J_2, J_3, J_4)$, so that $f_d(J_1, J_2, J_3, J_4) \equiv [F_d(J_1, J_2, J_3, J_4)]_L$.

We now make five observations relative to the inequalities (6.38).

1) Each of the inequalities present in (6.38) appears twice since $f_d(J_1, J_2, J_3, J_4) = f_d(J_3, J_4, J_1, J_2)$ for all $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, \ldots, t\}$. While we can impose additional restrictions on the sets $(J_1, J_2, J_3, J_4)_d$ to effectively eliminate such replications, we opt for notational convenience to maintain these duplicate restrictions, noting here that they can be trivially omitted.

2) The inequalities $0 \leq x_i \leq 1$ for $i = 1, \ldots, n$ do not explicitly appear in Problems BP or BLP because they are implied by the level $d = 2$ restrictions of (6.38). To see this, observe that for any i, $x_i = f_2(\emptyset, \{i\}, \{j\}, \emptyset) + f_2(\emptyset, \{i\}, \emptyset, \{j\})$ and $1 - x_i = f_2(\{i\}, \emptyset, \{j\}, \emptyset) + f_2(\{i\}, \emptyset, \emptyset, \{j\})$ for every $j \neq i$.

3) For any subsets J_1 and J_2 of N with $|J_1 \cup J_2| \geq 2$, we do not include inequalities of the form $[1 - \prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)]_L \geq 0$ in the constraint set for Problems BP or BLP

since they are implied by constraints (6.38). Given any such sets J_1 and J_2 with $J_1 \cap J_2 \neq \emptyset$, the binary identity $x_i^2 = x_i$ for each i gives $[1 - \prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)]_L = 1$. Given any such disjoint sets J_1 and J_2, if $J_1 \neq \emptyset$ then $[1 - \prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)]_L = 1 - x_j + f_{|J_1 \cup J_2|}(\emptyset, \{j\}, J_1 - \{j\}, J_2)$ for any $j \in J_1$, while if $J_1 = \emptyset$ then $[1 - \prod_{i \in J_2} (1 - x_i)]_L = x_j + f_{|J_2|}(\{j\}, \emptyset, \emptyset, J_2 - \{j\})$ for any $j \in J_2$. Hence, these inequalities are implied since, as shown above, for every $j \in N$, $1 - x_j \geq 0$ and $x_j \geq 0$ are implied by (6.38).

4) We only consider those inequalities of the form (6.38) for which the sets J_1, J_2, J_3, and J_4 satisfy $(J_1 \cup J_3) \cap (J_2 \cup J_4) = \emptyset$ since, again by the binary identity $x_i^2 = x_i$ for each i, all other inequalities are necessarily implied. To demonstrate, suppose that for some such sets of order d, we allow $J_1 \cap (J_2 \cup J_4) \neq \emptyset$, and consider $F_d(J_1, J_2, J_3, J_4)$. If $J_1 \cap J_2 \neq \emptyset$, then by the previous remark, $f_d(J_1, J_2, J_3, J_4) = [1 - \prod_{i \in J_3} x_i \prod_{i \in J_4} (1 - x_i)]_L$, and $f_d(J_1, J_2, J_3, J_4) \geq 0$ is implied. If $J_1 \cap J_4 \neq \emptyset$ so that $j \in (J_1 \cap J_4)$, then $F_d(J_1, J_2, J_3, J_4) = (1 - x_j)[1 - \prod_{i \in J_3} x_i \prod_{i \in J_4 - \{j\}} (1 - x_i)] + x_j[1 - \prod_{i \in J_1 - \{j\}} x_i \prod_{i \in J_2} (1 - x_i)]$, so that $f_d(J_1, J_2, J_3, J_4) = f_{|J_3 \cup J_4|}(\{j\}, \emptyset, J_3, J_4 - \{j\}) + f_{|J_1 \cup J_2|}(J_1 - \{j\}, J_2, \emptyset, \{j\})$, with nonnegativity implied. An identical argument holds for the case where $J_3 \cap (J_2 \cup J_4) \neq \emptyset$. Consequently, any inequality $f_d(J_1, J_2, J_3, J_4) \geq 0$ having $|(J_1 \cup J_3) \cap (J_2 \cup J_4)| \neq 0$ can be expressed as a sum of inequalities in (6.38).

5) For $t = 2$, Problems BP and IP(2) are identical in that they have precisely the same set of constraints. This follows since for any $(J_1, J_2, J_3, J_4)_2$, by definition we have

$$F_2(J_1, J_2, J_3, J_4) = \prod_{j \in J_2 \cup J_4} x_j \prod_{j \in J_1 \cup J_3} (1 - x_j).$$

Given any instance of Problem PP (of degree t), Problem BP is our equivalent mixed 0-1 linear reformulation of PP that possesses the persistency property. By observation 5 above, we know that persistency holds true for $t = 2$ by Theorem 6.4. To show this for $t \geq 3$, we begin by presenting Lemma 6.4 and Theorem 6.6 in order to establish the equivalence between Problems PP and BP that an optimal solution to either problem readily promotes an optimal solution to the other problem. Observe the similarity between Lemma 6.1 of Section 6.1 and Lemma 6.4 below.

Lemma 6.4. *Consider any feasible solution* $(\hat{x}, \hat{w})$ *to Problem BLP and any* $J \subseteq N$ *with* $1 \leq |J| \leq t$. *If there exists an* $i \in J$ *with* $\hat{x}_i = 1$ *then* $\hat{w}_{J-i}$. *Moreover, if there exists an* $i \in J$ *with* $\hat{x}_i = 0$ *then* $\hat{w}_J = 0$.

Proof. Since $\hat{w}_\varnothing = 1$ and $\hat{w}_j = \hat{x}_j$ for all $j = 1, \ldots, n$, the results hold trivially for all sets $J \subseteq N$ with $|J| = 1$. Now by induction assume the results hold true for all sets J with $|J| \leq r - 1$ for a given r satisfying $2 \leq r \leq t$. Consider the first statement, and any $J \subseteq N$ with $|J| = r$ such that there exists an $i \in J$ with $\hat{x}_i = 1$. At the point $(\hat{x}, \hat{w})$, $f_r(\{i\}, \varnothing, J - \{i\}, \varnothing) = 1 - \hat{x}_i - \hat{w}_{J-i} + \hat{w}_J \geq 0$ gives $\hat{w}_{J-i} \leq \hat{w}_J$ while for any $k \in J$, $k \neq i$, $f_r(\{i\}, \varnothing, J - \{i, k\}, \{k\}) = 1 - \hat{x}_i - \hat{w}_{J-\{i,k\}} + \hat{w}_{J-i} + \hat{w}_{J-k} - \hat{w}_J \geq 0$ gives, by the inductive hypothesis, $\hat{w}_{J-i} \geq \hat{w}_J$. Hence $\hat{w}_{J-i} = \hat{w}_J$. Now consider the second statement and any $J \subseteq N$ with $|J| = r$ such that there exists an $i \in J$ with $\hat{x}_i = 0$. At the point $(\hat{x}, \hat{w})$, $f_r(\varnothing, \{i\}, J - \{i\}, \varnothing) = \hat{x}_i - \hat{w}_J \geq 0$ gives

$\hat{w}_J \leq 0$ while for any $k \in J$, $k \neq i$, $f_r(\emptyset, \{i\}, J - \{i, k\}, \{k\}) = \hat{x}_i - \hat{w}_{J-k} + \hat{w}_J \geq 0$ gives, by the inductive hypothesis, $\hat{w}_J \geq 0$. Hence $\hat{w}_J = 0$. □

We now establish the desired equivalence between Problems PP and BP in Theorem 6.6 below.

Theorem 6.6. *Given any binary $\hat{x}$, there exists a $\hat{w}$ so that $(\hat{x}, \hat{w})$ is feasible to Problem BP with objective function value $\sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$. Moreover, given any $(\hat{x}, \hat{w})$ feasible to Problem BP, the objective function value is $\sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$.*

Proof. Let $\hat{x}$ be a binary vector. Then, by construction, $(\hat{x}, \hat{w})$, where $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$ with $|J| \leq t$, is feasible to Problem BP with objective function value $\sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$. Now, let $(\hat{x}, \hat{w})$ be feasible to BP. The proof is to show that $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$ with $1 \leq |J| \leq t$. The result holds trivially for any $J \subseteq N$ with $|J| = 1$. Since the feasible region to Problem BP is contained within the feasible region to BLP, the result follows directly from Lemma 6.4. □

The above theorem and proof provide a bijective mapping between the feasible regions of Problems PP and BP under which the associated objective function values are invariant. Indeed, a direct consequence is that a point $(\hat{x}, \hat{w})$ is optimal to Problem BP if and only if $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$ with $2 \leq |J| \leq t$ and $\hat{x}$ is an optimal solution to Problem PP. In the upcoming subsection we prove that for any $(\tilde{x}, \tilde{w})$ optimal to BLP, those components of $\tilde{x}$ realizing binary values will retain the same binary values in an optimal solution to PP (and therefore BP). Consequently, we will have proven that Problem BP possesses the persistency property. Problem BP thus becomes our persistent mixed 0-1 linear reformulation of PP.

Before proceeding to Subsection 6.3.1 and proving that Problem BP is a persistent reformulation of Problem PP, we present two lemmas that focus upon the feasible region to Problem BLP. Lemma 6.5 is used in the persistency proofs of Theorems 6.7 and 6.9 while Lemma 6.6 is invoked in proving Lemma 6.7 and the persistency argument of Corollary 6.1.

Lemma 6.5. *Problem BLP is feasible and bounded.*

Proof. Since Problem BLP is clearly feasible, the proof is to show that it is bounded. Toward this end, consider any $J \subseteq N$ with $2 \leq |J| \leq t$. For each $i \in J$, the inequality $f_{|J|}(\emptyset, \{i\}, J - \{i\}, \emptyset) \geq 0$ enforces that $w_J \leq x_i$. Next, given any $J \subseteq N$ with $|J| = 2$, so that $J = \{i, j\}$, $f_2(\{i\}, \emptyset, \{j\}, \emptyset) \geq 0$ assures that $w_J \geq x_i + x_j - 1$. If $t > 2$, assume by induction that for a given r satisfying $3 \leq r \leq t$, if $J \subseteq N$ with $2 \leq |J| \leq r - 1$, then $w_J \geq \sum_{j \in J} x_j - |J| + 1$. Now consider any $J \subseteq N$ with $|J| = r$, and observe that for any $i \in J$, $f_r(\{i\}, \emptyset, J - \{i\}, \emptyset) \geq 0$ gives $w_J \geq w_{J-i} + x_i - 1$. By the inductive hypothesis, we have $w_{J-i} \geq \sum_{j \in J-\{i\}} x_j - |J - \{i\}| + 1$ so that $w_J \geq \sum_{j \in J} x_j - |J| + 1$ for all $J \subseteq N$ with $2 \leq |J| \leq t$. Since the restrictions $0 \leq x_i \leq 1$ for all i are implied by (6.38) as earlier noted, we have that the feasible region is bounded. $\square$

Lemma 6.6. *Given any feasible solution* $(\hat{x}, \hat{w})$ *to Problem BLP and any* $J \subseteq N$, $K \subseteq N$, *with* $1 \leq |J \cup K| \leq t$ *and* $J \cap K = \emptyset$,

(a) if there exists either a $j \in K$ *with* $\hat{x}_j = 0$ *or a* $j \in J$ *with* $\hat{x}_j = 1$ *then, when evaluated at* $(\hat{x}, \hat{w})$, $[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L = 0$, *and*

(b) if there exists a $j \in K$ *with* $\hat{x}_j \neq 1$ *or there exists a* $j \in J$ *with* $\hat{x}_j \neq 0$ *then, when evaluated at* $(\hat{x}, \hat{w})$, $[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L < 1$.

Proof. Consider any point $(\hat{x}, \hat{w})$ and any sets J and K satisfying the conditions of the lemma.

(a) Observe that when evaluated at $(\hat{x}, \hat{w})$, $[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L = \sum_{J' \subseteq J} (-1)^{|J'|} \hat{w}_{J' \cup K}$ as in (6.2) so that if there exists a $j \in K$ with $\hat{x}_j = 0$ the proof follows directly from the second statement in Lemma 6.4. Otherwise, if there exists a $j \in J$ with $\hat{x}_j = 1$ then, when evaluated at $(\hat{x}, \hat{w})$,

$$[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L = \sum_{J' \subseteq J} (-1)^{|J'|} \hat{w}_{J' \cup K} = \sum_{J' \subseteq J - \{j\}} (-1)^{|J'|} (\hat{w}_{J' \cup K} - \hat{w}_{J' \cup K \cup \{j\}}),$$

and the proof follows from the first statement in Lemma 6.4.

(b) The proof follows trivially if $|J \cup K| = 1$, and so we consider the cases where $|J \cup K| \geq 2$. Given such sets J and K, if there exists a $j \in K$ with $\hat{x}_j \neq 1$ (so that $\hat{x}_j < 1$) then the constraint from BLP, $f_{|J \cup K|}(\emptyset, \{j\}, K - \{j\}, J) \geq 0$, evaluated at $(\hat{x}, \hat{w})$, gives $[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L \leq \hat{x}_j < 1$. Similarly, if there exists a $j \in J$ with

$\hat{x}_j \neq 0$ (so that $\hat{x}_j > 0$) then the constraint $f_{|J \cup K|}(\{j\}, \varnothing, K, J - \{j\}) \geq 0$, evaluated at $(\hat{x}, \hat{w})$, gives $[\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L \leq 1 - \hat{x}_j < 1$. □

Clearly, since by definition we have that for any $J \subseteq N$, $K \subseteq N$, with $1 \leq |J \cup K| \leq t$ and $J \cap K = \varnothing$, $f_{|J \cup K|}(K, J) = [\prod_{i \in K} x_i \prod_{i \in J} (1 - x_i)]_L$ the proof of part (a) is in the same spirit as that portion of the proof of Lemma 6.2 establishing primal feasibility. In fact, the second statement in the proof of part (a) above uses precisely the same argument as in (6.12).

As a final remark, we note that the proofs of Lemmas 6.4, 6.5, and 6.6, as well as that of Theorem 6.6, rely only on those inequalities in (6.38) for which the associated sets $(J_1, J_2, J_3, J_4)_d$ have $|J_1 \cup J_2| = 1$ and $|J_4| \leq 1$. Indeed, we can redefine Problem BP so that (6.38) includes just such restrictions. As opposed to computing as succinct a persistent reformulation as possible, our intent here is to present the broadest possible set of inequalities that permit persistency, letting the analyst judiciously choose subsets that maintain equivalence and boundedness (and, as we will see, persistency will necessarily continue to hold).

6.3.1. Persistency for Unconstrained 0-1 Polynomial Programs

In this subsection we show that Problem BP is an equivalent, persistent mixed 0-1 linear reformulation of Problem PP, regardless of the degree t of Problem PP. The equivalence between Problems PP and BP was established in Theorem 6.6, so we focus here on showing that Problem BP is persistent. Unlike the standard RLT approach introduced in Chapter 2 and invoked throughout Sections 6.1 and 6.2, we do not have here at our disposal a hierarchy of linearizations leading to the convex hull representation. Consequently, an entirely different approach is needed.

Our proof of persistency is, in fact, based on an elementary idea. Suppose that for a given solution x^I to PP indexed by the set $I \subseteq N$ (that is, with $x_j^I = 1$ for all $j \in I$ and $x_j^I = 0$ otherwise) there exist disjoint subsets N^+ and N^- of N such that the solution $x^{I-N^- \cup N^+}$ yields an objective value no greater than x^I. Simply stated, this new solution is obtained by assigning value 1 to those variables indexed in N^+ and value 0 to those indexed in N^-, and allowing variables whose indices are not contained in $N^+ \cup N^-$ to retain their values from x^I. Notationally, the solution $x^{I-N^- \cup N^+}$ yields an objective value in PP less than or equal to that of x^I if and only if

$$\sum_{J \subseteq I} c_J - \sum_{J \subseteq I-N^- \cup N^+} c_J \geq 0. \tag{6.39}$$

Suppose, moreover, that for the selected sets N^+ and N^-, (6.39) holds for all $I \subseteq N$. Then the objective value at $x^{I-N^- \cup N^+}$ is less than or equal to that at x^I for every $I \subseteq N$, and there must exist an optimal solution $\hat{x}$ to Problem PP with $\hat{x}_j = 1$ for all $j \in N^+$ and $\hat{x}_j = 0$ for all $j \in N^-$.

Using (6.39) and this logic, we will prove that Problem BP possesses the persistency property: that is, those variables taking values of 1 or 0 in any optimal solution to BLP will persist in retaining those same values in an optimal solution to BP. Our approach is to show that the constraint structure of BLP dictates that (6.39) is indeed true for all $I \subseteq N$ when the sets N^+ and N^- are defined to be the index sets of those variables taking values 1 and 0, respectively, in an optimal primal solution. Thus, we will have shown that $\hat{x}_j = 1$ for all $j \in N^+$ and $\hat{x}_j = 0$ for all $j \in N^-$ in an optimal solution $\hat{x}$ to Problem PP and, from Theorem 6.6, that $\hat{x}$ is part of an optimal solution to BP.

An overview of our line of attack is as follows. We first show in Lemma 6.7 that given any feasible solution to Problem BLP, certain of the constraints defined in terms of the posed solution must be satisfied with strict inequality. This result is used in Lemma 6.8 to identify conditions under which select polynomial expressions $F_d(J_1, J_2, J_3, J_4)$, once again defined in terms of the posed solution, must realize the value 0. We then express (6.39) in terms of the dual region to Problem BLP. Finally, we use Lemma 6.8 and the Karush-Kuhn-Tucker complementary slackness conditions in Theorem 6.7 to show that, in the resulting representation of (6.39), any dual variable taking a strictly positive value in an optimal dual solution must have a nonnegative coefficient on the left-hand side of the inequality. This fact, along with the nonnegativity of the duals, proves the inequality for all $I \subseteq N$.

We begin by directing attention to inequalities (6.38) of Problem BLP, and deriving sufficient conditions on any given primal solution to ensure that certain of these constraints are satisfied with strict inequality. Here, we use the observation that the constraints in (6.38) can be expressed as

$$f_d(J_1, J_2, J_3, J_4) = 1 - [\prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)]_L - [\prod_{i \in J_3} x_i \prod_{i \in J_4} (1 - x_i)]_L +$$

$$[\prod_{i \in J_1 \cup J_3} x_i \prod_{i \in J_2 \cup J_4} (1 - x_i)]_L \geq 0 \text{ for each } (J_1, J_2, J_3, J_4)_d,\ d \in \{2, ..., t\}. \quad (6.40)$$

Lemma 6.7. *Let $(\hat{x}, \hat{w})$ be any feasible solution to Problem BLP, and define the subsets N^+ and N^- of N in terms of $\hat{x}$ as $N^+ = \{j : \hat{x}_j = 1\}$ and $N^- = \{j : \hat{x}_j = 0\}$. Given any $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, ..., t\}$, $f_d(J_1, J_2, J_3, J_4)$ evaluated at $(\hat{x}, \hat{w})$ is strictly positive in each of the following cases:*

(i) $J_1 \cap N^- \neq \emptyset$ *and* $J_3 \not\subseteq N^+$, *(ii)* $J_1 \cap N^- \neq \emptyset$ *and* $J_4 \not\subseteq N^-$,

(iii) $J_2 \cap N^+ \neq \emptyset$ *and* $J_3 \not\subseteq N^+$, *(iv)* $J_2 \cap N^+ \neq \emptyset$ *and* $J_4 \not\subseteq N^-$,

(v) $J_3 \cap N^- \neq \emptyset$ *and* $J_1 \not\subseteq N^+$, *(vi)* $J_3 \cap N^- \neq \emptyset$ *and* $J_2 \not\subseteq N^-$,

(vii) $J_4 \cap N^+ \neq \emptyset$ *and* $J_1 \not\subseteq N^+$, *and (viii)* $J_4 \cap N^+ \neq \emptyset$ *and* $J_2 \not\subseteq N^-$.

Proof. If $(J_1, J_2, J_3, J_4)_d$ satisfies any of cases (i) through (iv) then in light of (6.40), Lemma 6.6 enforces that, when evaluated at $(\hat{x}, \hat{w})$, $f_d(J_1, J_2, J_3, J_4) = 1 - [\prod_{i \in J_3} x_i \prod_{i \in J_4} (1 - x_i)]_L > 0$, with the equality following from statement (a) and the inequality from (b). Similarly, if $(J_1, J_2, J_3, J_4)_d$ satisfies any of cases (v) through (viii) then Lemma 6.6 enforces that, when evaluated at $(\hat{x}, \hat{w})$, $f_d(J_1, J_2, J_3, J_4) = 1 - [\prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)]_L > 0$. □

Lemma 6.8 uses the result of Lemma 6.7 to specify conditions under which select inequalities $F_d(J_1, J_2, J_3, J_4) \geq 0$ from (6.38) are satisfied with equality at $x^{I-N^- \cup N^+}$.

Lemma 6.8. *Let $(\hat{x}, \hat{w})$ be any feasible solution to Problem BLP, and define the subsets N^+ and N^- of N as in Lemma 6.7. Given any $I \subseteq N$, and any $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, ..., t\}$, such that $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I, if $f_d(J_1, J_2, J_3, J_4) = 0$ at $(\hat{x}, \hat{w})$ then $F_d(J_1, J_2, J_3, J_4)$ evaluated at $x^{I-N^- \cup N^+}$ gives a value of 0 as well.*

Proof. Clearly, the expression $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I if and only if either (a) $J_1 \subseteq I$ and $I \cap J_2 = \varnothing$ ($x_j = 1 \ \forall \ j \in J_1$ and $x_j = 0 \ \forall \ j \in J_2$) or (b) $J_3 \subseteq I$ and $I \cap J_4 = \varnothing$. Since these two cases are symmetric, we assume without loss of generality that (a) holds true. We address the contrapositive. Suppose for some $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, ..., t\}$, with $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I that $F_d(J_1, J_2, J_3, J_4) \neq 0$ at $x^{I-N^- \cup N^+}$. Then (i) either $J_1 \not\subseteq I - N^- \cup N^+$ or $J_2 \cap (I - N^- \cup N^+) \neq \varnothing$ and (ii) either $J_3 \not\subseteq I - N^- \cup N^+$ or $J_4 \cap (I - N^- \cup N^+) \neq \varnothing$. More simply, we are reduced to the following four possibilities:

1) $J_1 \subseteq I$, $J_2 \cap I = \varnothing$ and $J_1 \not\subseteq I - N^- \cup N^+$ and $J_3 \not\subseteq I - N^- \cup N^+$,

2) $J_1 \subseteq I$, $J_2 \cap I = \varnothing$ and $J_1 \not\subseteq I - N^- \cup N^+$ and $J_4 \cap (I - N^- \cup N^+) \neq \varnothing$,

3) $J_1 \subseteq I$, $J_2 \cap I = \varnothing$ and $J_2 \cap (I - N^- \cup N^+) \neq \varnothing$ and $J_3 \not\subseteq I - N^- \cup N^+$, or

4) $J_1 \subseteq I$, $J_2 \cap I = \varnothing$ and $J_2 \cap (I - N^- \cup N^+) \neq \varnothing$ and $J_4 \cap (I - N^- \cup N^+) \neq \varnothing$.

These four possibilities satisfy the conditions of the first four cases of Lemma 6.7, respectively, and therefore $f_d(J_1, J_2, J_3, J_4) > 0$ at $(\hat{x}, \hat{w})$. □

At this point, we observe that a suitably-dimensioned vector π constitutes a dual feasible solution to BLP if and only if

$$\pi_d(J_1, J_2, J_3, J_4) \geq 0 \qquad \forall\ (J_1, J_2, J_3, J_4)_d,\ d = 2, \ldots, t \tag{6.41}$$

and for all real (x, w), we have that

$$\sum_{d=2}^{t} \Big[\sum_{(J_1,J_2,J_3,J_4)_d} \pi_d(J_1, J_2, J_3, J_4) f_d(J_1, J_2, J_3, J_4) \Big] = \sum_{J \subseteq N} c_J w_J + \theta \tag{6.42}$$

where

$$\theta = \sum_{d=2}^{t} \sum_{\substack{(J_1,J_2,J_3,J_4)_d \\ J_1 \neq \varnothing, J_3 \neq \varnothing}} \pi_d(J_1, J_2, J_3, J_4) \tag{6.43}$$

and where π has components $\pi_d(J_1, J_2, J_3, J_4)$ for all $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, \ldots, t\}$, with each $\pi_d(J_1, J_2, J_3, J_4)$ multiplier corresponding, using obvious notation, to the associated constraint $f_d(J_1, J_2, J_3, J_4) \geq 0$. A key observation here is that since $f_d(J_1, J_2, J_3, J_4) = [F_d(J_1, J_2, J_3, J_4)]_L$, given any dual feasible solution $\hat{\pi}$ to Problem BLP, equation (6.42) enforces for all binary x that

$$\sum_{d=2}^{t}\Big[\sum_{(J_1,J_2,J_3,J_4)_d} \hat{\pi}_d(J_1,J_2,J_3,J_4)F_d(J_1,J_2,J_3,J_4)\Big] - \theta = \sum_{J\subseteq N} c_J \prod_{j\in J} x_j, \tag{6.44}$$

where the scalar θ is as defined in (6.43). Thus, to verify inequalities (6.39) and establish persistency for Problem BP, we need only show for the prescribed sets N^+ and N^- and a given dual solution $\hat{\pi}$ that for every $I \subseteq N$ the left-hand side of Equation (6.44) evaluated at x^I is no less than the left-hand side evaluated at $x^{I-N^- \cup N^+}$ (since $\sum_{J\subseteq I} c_J$ is just $\sum_{J\subseteq N} c_J \prod_{j\in J} x_j$ evaluated at x^I). This is precisely the argument used to prove our main persistency result of Theorem 6.7, with $\hat{\pi}$ representing any optimal dual solution to BLP.

Theorem 6.7. *Let $(\hat{x}, \hat{w})$ be any optimal solution to Problem BLP, and define the subsets N^+ and N^- of N as in Lemma 6.7. There exists an optimal solution x^* to Problem PP with $x_j^* = 1$ for all $j \in N^+$ and $x_j^* = 0$ for all $j \in N^-$. Hence, there exists a w^* such that (x^*, w^*) is optimal to Problem BP.*

Proof. Let $\hat{\pi}$ denote any optimal dual solution to BLP. By Lemma 6.5, such a solution must exist. Now, given any $I \subseteq N$, consider the associated inequality (6.39) expressed in terms of $\hat{\pi}$ via (6.44). From complementary slackness to BLP, if some $\hat{\pi}_d(J_1, J_2, J_3, J_4) \neq 0$ then $f_d(J_1, J_2, J_3, J_4) = 0$ at $(\hat{x}, \hat{w})$ so that from Lemma 6.8, if $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I then $F_d(J_1, J_2, J_3, J_4) = 0$ at $x^{I-N^- \cup N^+}$ Consequently, the coefficient on any $\hat{\pi}_d(J_1, J_2, J_3, J_4) \neq 0$ in the resulting expression is either 0 or 1. Since $\hat{\pi}_d(J_1, J_2, J_3, J_4) \geq 0$ for all $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, ..., t\}$, by dual feasibility to BLP, it follows that (6.39) holds for all $I \subseteq N$. Therefore, there exists an optimal solution x^* to Problem PP with $x_j^* = 1$ for all $j \in N^+$ and $x_j^* = 0$ for all $j \in N^-$ and, from Theorem 6.6, there exists a w^* such that (x^*, w^*) is optimal to Problem BP. □

Theorem 6.7 demonstrates that Problem BP possesses the persistency property, and the method of proof can be used to obtain related results specifying conditions under which certain products of 0-1 variables must retain prescribed values at *all* optimal solutions to Problems BP and PP. Observe for the computed sets N^+ and N^-, that given any $(J_1, J_2, J_3, J_4)_d$, $d \in \{2, ..., t\}$, with associated optimal dual multiplier $\hat{\pi}_d(J_1, J_2, J_3, J_4) > 0$, no x^I having $F_d(J_1, J_2, J_3, J_4) = 1$ at x^I and $F_d(J_1, J_2, J_3, J_4) = 0$ at $x^{I-N^- \cup N^+}$ can be optimal to PP since the inequality (6.39) would necessarily hold strictly. We can use this observation together with the Strong Theorem of Complementary Slackness to deduce that those variables realizing a given binary value at *all* optimal solutions to BLP must persist in retaining their same respective values at *all* optimal solutions to PP and BP. A formal statement and proof of this persistency result are given below as a corollary to Theorem 6.7.

Corollary 6.1. *Given an instance of Problem PP, define the subsets K^+ and K^- of N as $K^+ = \{i : x_i = 1$ at all optimal solutions to BLP$\}$ and $K^- = \{i : x_i = 0$ at all optimal solutions to BLP$\}$. Then $x_i = 1$ for all $i \in K^+$ and $x_i = 0$ for all $i \in K^-$ at all optimal solutions to PP, and consequently at all optimal solutions to BP.*

Proof. Define the subsets N^+ and N^- of N in terms of any optimal primal solution $(\hat{x}, \hat{w})$ to Problem BLP as in Lemma 6.7. Now, suppose there exists an $i \in K^+$. Consider any solution x^I to PP such that $x_i^I = 0$. Compute sets $(J_1, J_2, J_3, J_4)_2$ as follows. Fix $J_1 = \{i\}$ and $J_2 = \emptyset$. For any $k \in N$ with $k \neq i$, fix $J_3 = \emptyset$ and $J_4 = \{k\}$ if $x_k^I = 1$ to obtain $f_2(J_1, J_2, J_3, J_4) = (1 - x_i)x_k$, but fix $J_3 = \{k\}$ and $J_4 = \emptyset$ if $x_k^I = 0$ to obtain $f_2(J_1, J_2, J_3, J_4) = (1 - x_i)(1 - x_k)$. In either case, observe from part (a) of Lemma 6.6 that $f_2(J_1, J_2, J_3, J_4)$ must equal 0 at all optimal primal solutions to BLP. Consequently, by the Strong Theorem of Complementary Slackness, there must exist an optimal dual solution $\hat{\pi}$ to BLP having $\hat{\pi}_2(J_1, J_2, J_3, J_4) > 0$. Moreover, the expression $F_2(J_1, J_2, J_3, J_4)$ equals 1 when evaluated at x^I and, since $K^+ \subseteq N^+$, equals 0 when evaluated at $x^{I-N^- \cup N^+}$. Thus, from (6.44), the positive dual multiplier $\hat{\pi}_2(J_1, J_2, J_3, J_4)$ has a coefficient of 1 in the left-hand side of inequality (6.39) so that by the proof of Theorem 6.7,

$$\sum_{J \subseteq I} c_J > \sum_{J \subseteq I-N^- \cup N^+} c_J$$

and x^I cannot be optimal to PP. As the set I was arbitrarily selected so that $x_i^I = 0$, we have that $x_i = 1$ for all $i \in K^+$ at *all* optimal solutions to PP and, by Theorem 6.6, at *all* optimal solutions to BP. In a similar fashion, suppose there exists an $i \in K^-$ and consider any solution x^I to PP such that $x_i^I = 1$. The proof to show that x^I cannot be optimal to PP proceeds in an identical manner with $J_1 = \emptyset$ and $J_2 = \{i\}$, and since $K^- \subseteq N^-$. $\square$

We conclude this subsection with two examples to illustrate our persistency results and to address a related persistency issue. Our first example demonstrates how Theorem 6.7 can be used to reduce the size of a given 0-1 polynomial program by allowing a variable to be fixed at a binary value.

Example 6.6. Consider the instance of Problem PP having degree $t = 3$ in $n = 5$ binary variables given below.

Minimize $\quad x_3 + x_4 + x_1x_2 - x_1x_4 - x_2x_4 - x_3x_4 - x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4$

subject to $\quad x_i$ binary $\quad i = 1,\ldots,4.$

The reader can verify that the optimal objective function value to BLP is $-\frac{1}{2}$ with an optimal primal solution of $(x_1, x_2, x_3, x_4) = (\frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$, $w_{14} = w_{24} = \frac{1}{2}$, and all other variables taking value 0. Here, using the notation of the Theorem, $N = \{1,2,3,4\}$, $N^+ = \varnothing$, and $N^- = \{3\}$. From Theorem 6.7 we know that x_3 must be 0 at an optimal binary solution to the problem, and therefore the original problem reduces to minimizing $x_4 + x_1x_2 - x_1x_4 - x_2x_4$. By inspection we can see that the optimal objective value is 0 and that the trivial solution, $(x_1, x_2, x_3, x_4) = (0,0,0,0)$, is thus an optimal solution to the original problem.

In the proof of Theorem 6.7 we use the fact that, given any feasible solution x^I to PP and any optimal solution $(\hat{x}, \hat{w})$ to BLP defining the sets N^+ and N^-, the solution $x^{I-N^- \cup N^+}$ yields an objective function value no greater than that provided by x^I. The question arises as to whether this same relationship holds if we consider *subsets* of the sets N^+ and N^-. As the second example illustrates, such is not the case.

Example 6.7. Consider the instance of Problem PP having degree $t = 2$ in $n = 5$ binary variables given below.

Minimize $\quad x_1 + x_5 - x_1x_2 - x_1x_3 + x_2x_3 - 2x_4x_5$

subject to $\quad x_i$ binary $\quad i = 1,\ldots,5.$

An optimal solution (x, w) to BLP is $(x_1, x_2, x_3, x_4, x_5) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1)$, $w_{23} = 0$, $w_{45} = 1$ and $w_{ij} = \frac{1}{2}$ otherwise, with objective function value $-\frac{3}{2}$, so that $N^+ = \{4,5\}$ and $N^- = \varnothing$. Consider the feasible solution $x^I = x^{\varnothing} = (0,0,0,0,0)$ with objective value 0. As mentioned above, the solution $x^{I-N^- \cup N^+} = (0,0,0,1,1)$ must yield an objective value no greater than that of $x^{\varnothing}$; in this case we obtain -1. Complementing just one of the variables on which the two solutions do not agree, however, does not yield the same result; if only x_5 is changed to $x_5 = 1$ we obtain an objective value of 1, which is strictly greater than the 0 value obtained at $x^{\varnothing}$.

6.3.2. Persistency for Constrained 0-1 Polynomial Programs

In this section we show that certain specially-structured, constrained versions of Problem PP can be reformulated as persistent mixed 0-1 linear programs. The constraints consist of arbitrary subsets of the inequalities (6.37) restricted to hold at equality; interestingly, these constraints can be used to model, in Boolean form, pairwise disjunctions of conjunctions. For the special case in which the maximum degree of any term in the objective function and constraints is $t = 2$, we get Problem RIP(2) of Section 6.2.

Our approach is similar to that of the previous subsection; we in fact extend the arguments found there to accommodate constrained problems. We prove a persistency result analogous to that of Theorem 6.7, and then show that Corollary 6.1 dealing with the identification of variables realizing a given binary value at all optimal binary solutions can also be extended to the constrained case. Finally, we provide two examples. The first illustrates how our persistency results can be used to fix variables at binary values and the second shows that, even for a slight relaxation of our formulation structure, persistency will not necessarily hold.

In constructing our constrained 0-1 polynomial programs, we once again use the observation that a binary solution x^I satisfies the restriction $F_d(J_1, J_2, J_3, J_4) = 0$ for some $d \in \{2, ..., t\}$ and $(J_1, J_2, J_3, J_4)_d$ if and only if either (a) $J_1 \subseteq I$ and $I \cap J_2 = \emptyset$ or (b) $J_3 \subseteq I$ and $I \cap J_4 = \emptyset$, that is, x^I_j must be 1 for all $j \in J_1$ and 0 for all $j \in J_2$ or else x^I_j must be 1 for all $j \in J_3$ and 0 for all $j \in J_4$. This observation, together with Lemma 6.7, preserves the result of Lemma 6.8 and permits us to express a constrained problem, whose restrictions can be written in Boolean form as pairwise disjunctions of conjunctions (i.e. $\prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i) = 1$ or $\prod_{i \in J_3} x_i \prod_{i \in J_4} (1 - x_i) = 1$), as mixed 0-1 linear programs possessing the persistency property.

Formally, we define the restricted (constrained) 0-1 polynomial program as follows.

RPP: Minimize $\sum_{J \subseteq N} c_J \prod_{j \in J} x_j$

subject to $F_d(J_1, J_2, J_3, J_4) = 0 \quad \forall\ (J_1, J_2, J_3, J_4)_d \in E$ (6.45)

x binary

where $E \subseteq \{(J_1, J_2, J_3, J_4)_d,\ d \in \{2, ..., n\}$. Problem RPP is a restricted version of Problem PP in that Equations (6.45) have been included in the problem. (Observe that Problem RPP differs from Problem CPP of Section 6.2 in the form of the equality restrictions.) RPP is said to be of degree t if t is the maximum degree of any polynomial term appearing with nonzero coefficient, either in the objective function or the constraints. In the same spirit of Section 6.2, given an instance of RPP of degree t, we define the following mixed 0-1 linear program.

CBP: Minimize $\sum_{J \subseteq N} c_J w_J$

subject to $f_d(J_1, J_2, J_3, J_4) \geq 0 \quad \forall\ (J_1, J_2, J_3, J_4)_d \notin E,\ d = 2, ..., t$ (6.46)

$f_d(J_1, J_2, J_3, J_4) = 0 \quad \forall\ (J_1, J_2, J_3, J_4)_d \in E$ (6.47)

x binary.

Problem CBP is obtained from RPP by including the redundant restrictions (6.46) within the constraint set, relaxing Equations (6.45) to (6.47), and linearizing the objective function by substituting for each product term $\prod_{j \in J} x_j$, the continuous variable w_J. Denote by CBLP the continuous relaxation of CBP obtained by omitting the x binary restrictions.

Theorem 6.8 below addresses the equivalence between Problems RPP and CBP.

Theorem 6.8. *Given any $\hat{x}$ feasible to Problem RPP, there exists a $\hat{w}$ so that $(\hat{x}, \hat{w})$ is feasible to Problem CBP with objective function value $\sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$. Moreover, given any $(\hat{x}, \hat{w})$ feasible to CBP, $\hat{x}$ is feasible to RPP with objective function value $\sum_{J \subseteq N} c_J \hat{w}_J$.*

Proof. Given any $\hat{x}$ feasible to RPP, define $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$, $1 \le |J| \le t$ so that $\sum_{J \subseteq N} c_J \hat{w}_J = \sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$. By definition, at $(\hat{x}, \hat{w})$, $f_d(J_1, J_2, J_3, J_4) = F_d(J_1, J_2, J_3, J_4) \ge 0$ for all $(J_1, J_2, J_3, J_4)_d$, $d = 2, \ldots, t$, and $f_d(J_1, J_2, J_3, J_4) = F_d(J_1, J_2, J_3, J_4) = 0$ for all $(J_1, J_2, J_3, J_4)_d \in E$. To prove the second statement, we recall from the proof of Theorem 6.6 that any $(\hat{x}, \hat{w}), \hat{x}$ binary, which satisfies $f_d(J_1, J_2, J_3, J_4) \ge 0$ for all $(J_1, J_2, J_3, J_4)_d$, $d = 2, \ldots, t$, must have $\hat{w}_J = \prod_{j \in J} \hat{x}_j$ for all $J \subseteq N$ such that $1 \le |J| \le t$. Consequently, at any such point $(\hat{x}, \hat{w})$ which also satisfies (6.47), we have that $F_d(J_1, J_2, J_3, J_4) = f_d(J_1, J_2, J_3, J_4) = 0$ for all $(J_1, J_2, J_3, J_4)_d \in E$ and that $\sum_{J \subseteq N} c_J \hat{w}_J = \sum_{J \subseteq N} c_J \prod_{j \in J} \hat{x}_j$. □

In order to prove persistency for Problem CBP, we will show that given any subsets N^+ and N^- of N defined in terms of an optimal primal solution $(\hat{x}, \hat{w})$ to CBLP as $N^+ = \{j : \hat{x}_j = 1\}$ and $N^- = \{j : \hat{x}_j = 0\}$ and given any x^I feasible to RPP, then (i) $x^{I - N^- \cup N^+}$ is also feasible to RPP and (ii) the objective function value to RPP at $x^{I - N^- \cup N^+}$ is less than or equal to that at x^I. Part (i) will follow from Lemma 6.8. Part (ii), however, is somewhat more involved and translates into showing that (6.39) must hold for all sets I indexing feasible binary solutions to RPP. To show this second point, we make use of an optimal dual solution, this time to CBLP. Except for the nonnegativity restrictions, this dual problem is identical to that for BLP. Clearly, a vector π is dual feasible to CBLP if and only if for all real (x,w) we have that equations (6.42) and (6.43) are satisfied, with $\pi_d(J_1, J_2, J_3, J_4) \geq 0$ for all $(J_1, J_2, J_3, J_4)_d \notin E$, $d = \{2, ..., t\}$. Here, in the context of Problem CBLP, $\pi_d(J_1, J_2, J_3, J_4)$ corresponds to the constraint $f_d(J_1, J_2, J_3, J_4) \geq 0$ if $(J_1, J_2, J_3, J_4)_d \notin E$ and to $f_d(J_1, J_2, J_3, J_4) = 0$ if $(J_1, J_2, J_3, J_4)_d \in E$. Our persistency result is given below.

Theorem 6.9. *Let $(\hat{x}, \hat{w})$ be any optimal solution to Problem CBLP, and define the subsets N^+ and N^- of N in terms of $\hat{x}$ as $N^+ = \{j : \hat{x}_j = 1\}$ and $N^- = \{j : \hat{x}_j = 0\}$. Then, provided that Problem RPP is feasible, there exists an optimal solution x^* to RPP with $x_j^* = 1$ for all $j \in N^+$ and $x_j^* = 0$ for all $j \in N^-$ and, consequently, a w^* such that (x^*, w^*) is optimal to Problem CBP.*

Proof. Our first step is to show that given any x^I feasible to RPP, $x^{I-N^- \cup N^+}$ must be feasible as well. This follows immediately from Lemma 6.8 since $(\hat{x}, \hat{w})$ must also be feasible to BLP (implying that Lemmas 6.6 through 6.8 hold for Problem CBLP), and since for every x^I feasible to RPP and every $(J_1, J_2, J_3, J_4)_d \in E$, we have that $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I and $f_d(J_1, J_2, J_3, J_4) = 0$ at $(\hat{x}, \hat{w})$ by feasibility to CBLP, so that $F_d(J_1, J_2, J_3, J_4)$ must be 0 at $x^{I-N^- \cup N^+}$.

To finish the proof, we show that (6.39) is true for all $I \subseteq N$ such that x^I is feasible to RPP. We accomplish this via an optimal dual solution $\hat{\pi}$ to CBLP. Exactly as in the proof of Theorem 6.7, given any $I \subseteq N$, express (6.39) in terms of $\hat{\pi}$ via equations (6.44). Since the feasible region to CBLP is contained within that of BLP, Lemma 6.5 assures that any feasible instance of CBLP is also bounded so that such a $\hat{\pi}$ exists. From complementary slackness to CBLP and Lemma 6.8, the coefficient on any $\hat{\pi}_d(J_1, J_2, J_3, J_4) \neq 0$ in the resulting expression is either 0 or 1. If $\hat{\pi}_d(J_1, J_2, J_3, J_4) < 0$ then $(J_1, J_2, J_3, J_4)_d \in E$ which implies, since x^I is feasible to RPP, that $F_d(J_1, J_2, J_3, J_4) = 0$ at x^I. Again invoking Lemma 6.8, it follows from feasibility of $(\hat{x}, \hat{w})$ to CBLP that the coefficient in the difference (6.39) must be 0. □

Two remarks are in order. First, if Problem CBLP is infeasible, then so is CBP, as well as RPP by Theorem 6.8. We therefore implicitly assume in Theorem 6.9 that CBLP is feasible. Second, the above theorem, as well as the following Corollary 6.2, qualifies the persistency result to include only feasible instances of RPP. This is because it is possible, for given instances of RPP, to have CBLP feasible and RPP infeasible. A simple example is the instance where $n = 3$, $t = 2$, and the equations (6.45) are defined by the set $E = \{(J_1, J_2, J_3, J_4)_2 : |J_2 \cup J_4| \text{ is even}\}$. In this case, RPP is infeasible while $(x_1, x_2, x_3, w_{12}, w_{13}, w_{23}) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 0)$ is feasible to CBLP. (This is the same example found in the remark in Section 6.2 immediately following the proof of Theorem 6.4C.)

The persistency results of Corollary 6.1 dealing with the recognition of binary variables realizing a given value at *all* optimal solutions to Problems PP and BP carries over to

Problem RPP and the linear reformulation CBP. The statement and proof are given below.

Corollary 6.2. *Given a feasible instance of Problem RPP, define the subsets K^+ and K^- of N as $K^+ = \{i : x_i = 1$ at all optimal solutions to CBLP$\}$ and $K^- = \{i : x_i = 0$ at all optimal solutions to CBLP$\}$. Then $x_i = 1$ for all $i \in K^+$ and $x_i = 0$ for all $i \in K^-$ at all optimal solutions to RPP, and consequently at all optimal solutions to CBP.*

Proof. The proof follows in a similar manner to that of Corollary 6.1 with the sets N^+ and N^- defined in terms of Problem CBLP as in Theorem 6.9. Here, however, given an $i \in K^+$, we recognize the case in which no solution exists to RPP with $x_i = 0$; in this instance the result holds trivially. Otherwise, given that such a solution x^I exists, the constructed $F_2(J_1, J_2, J_3, J_4)$ expression must necessarily have $(J_1, J_2, J_3, J_4)_2 \notin E$. Thus, by replacing the references to Theorems 6.6 and 6.7 with that of Theorems 6.8 and 6.9 respectively, the result follows for $i \in K^+$. The analogous argument holds for $i \in K^-$. □

We now present our two examples. The first shows how we can use Theorem 6.9 to fix a variable in Problem RPP at a binary value.

Example 6.8. Consider the instance of Problem RPP having degree $t = 3$ in $n = 5$ binary variables that consists of Example 6.6 with the added restriction, not satisfied by the optimal binary solution given there, that either (i) $x_2 = 1$ or (ii) $x_1 = 1$ and $x_3 = 0$. This problem is as follows.

$$\text{Minimize} \quad x_3 + x_4 + x_1x_2 - x_1x_4 - x_2x_4 - x_3x_4 - x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4$$

$$\text{subject to} \quad F_3(\{2\},\varnothing,\{1\},\{3\}) = 0$$

$$x_i \text{ binary} \quad i = 1,\ldots,4.$$

The optimal objective value to CBLP, where $E = \{(\{2\},\varnothing,\{1\},\{3\})\}$, is $-\frac{1}{2}$ with $(x_1, x_2, x_3, x_4) = (\frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$ as in Example 6.6. Therefore, even with the added restriction, we know from Theorem 6.9 that $x_3 = 0$ at an optimal solution. The reader can verify that there exist five optimal solutions to RPP, given by $(x_1, x_2, x_3, x_4) =$ (0,1,0,0), (0,1,0,1), (1,0,0,0), (1,0,0,1), and (1,1,0,1).

Our next example shows that even for slight variations of Problem RPP, the persistency results do not generally hold. The packing problem of this example can be formulated in terms of Problem CBP provided a single additional "cubic" constraint is enforced. As is demonstrated, this single restriction causes persistency to fail.

Example 6.9. Consider the following 0-1 linear program in $n = 4$ binary variables.

$$\text{Minimize} \quad -\frac{3}{2}x_1 - 2x_2 - 2x_3 - 2x_4$$

$$\begin{aligned} \text{subject to} \quad x_1 + x_2 + x_3 &\le 2 \\ x_2 + x_4 &\le 1 \end{aligned}$$

$$x_3 + \quad x_4 \quad \leq 1$$

$$x_i \text{ binary} \quad i = 1,\ldots,4.$$

In the sense of Theorem 6.8, this problem is equivalent to the following.

$$\begin{array}{ll} \text{Minimize} & -\frac{3}{2}x_1 - 2x_2 - 2x_3 - 2x_4 \\ \text{subject to} & f_d(J_1, J_2, J_3, J_4) \geq 0 \quad \forall\ (J_1, J_2, J_3, J_4)_d \notin E,\ d = 2,\ldots,3 \\ & w_{123} = 0 \\ & f_2(\varnothing,\{2\},\varnothing,\{4\}) = 0 \\ & f_2(\varnothing,\{3\},\varnothing,\{4\}) = 0 \\ & x_i \text{ binary} \quad i = 1,\ldots,4. \end{array}$$

Here, $E = \{(\varnothing,\{2\},\varnothing,\{4\}), (\varnothing,\{3\},\varnothing,\{4\})\}$. Note that $w_{123} = 0$ cannot be written as $f_3(J_1, J_2, J_3, J_4) = 0$ and therefore is not part of the set of allowable constraints. The reader can verify that an optimal solution to this problem has objective value $-\frac{9}{2}$, realized at $(x_1, x_2, x_3, x_4) = (1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, $w_{12} = w_{13} = w_{14} = \frac{1}{2}$, and all other primal variables 0. The unique optimal solution to the binary packing problem is clearly $(x_1, x_2, x_3, x_4) = (0,1,1,0)$. Thus, this problem does not possess the persistency property.

We conclude this subsection by mentioning that the arguments presented throughout Section 6.3 and Subsections 6.3.1 and 6.3.2 are sufficiently general to promote alternate persistent linear reformulations of Problem RPP, that differ from Problem CBP in terms of both size and strength of the continuous relaxation. Observe that these arguments will continue to hold for Problem CBP when *any* subset of the inequalities (6.46) are enforced, provided that the equivalence of Theorem 6.8 is maintained. As alluded to in Section 6.3, we can, for example, include in (6.46) only those inequalities for which the associated sets $(J_1, J_2, J_3, J_4)_d$ have $|J_1 \cup J_2| = 1$ and $|J_4| \leq 1$. While reducing the problem size, such an elimination of constraints has the drawback of potentially weakening the linear programming relaxation. Our intent in this chapter, however, is to present the most general family of inequalities that permit persistency while pointing out that, if desired, subsets can also be used.

As opposed to devising more compact formulations of Problem RPP than Problem CBP, one may wish to obtain persistent formulations whose linear programming relaxations provide increased strength. Given an instance of Problem RPP of degree t, we can define the inequalities (6.46) for all $(J_1, J_2, J_3, J_4)_d$ with $d = 2, \ldots, s$, where $s \in \{t, \ldots, n\}$, and continue to maintain persistency since our arguments are not limited to the degree t of RPP. The additional inequalities resulting when $s > t$ can, if fact, be shown to tighten the linear programming relaxation. We refer the reader to Lassiter and Adams (1997) for such an example, as well as for a more detailed discussion on different persistent modifications of Problem CBP.

6.4. Relationships to Published Results

The published persistency results center around the vertex packing problem and variants. The seminal work is due to Nemhauser and Trotter (1975), who proved that a standard formulation of the vertex packing problem (described in Section 6.2 and found in Example 6.5) possesses the persistency property. These authors used a graph-theoretic approach to essentially show that any 0-1 linear program having constraints exclusively of the form $x_i + x_j \leq 1$ and $x_i \geq 0$ is persistent. Picard and Queyranne (1977) showed that, within the set of alternate optimal solutions to the continuous relaxation of the vertex packing problem, there exists a unique maximal set of variables which are integral. Hammer *et al.* (1982) proved that if a variable takes the same binary value in *all* optimal solutions to the continuous relaxation of the vertex packing problem then it realizes that binary value in *all* optimal binary solutions. Hammer *et al.* (1984) provided a relationship between the vertex packing problem and a linearization strategy due to Rhys (1970), resulting in a persistency proof for Rhys' formulation. All these authors used in their proofs a characterization by Balinski (1965) of the extreme point solutions of the continuous relaxation stating that, given any extreme point solution $\hat{x}$, each component of $\hat{x}$ must take the value 0, $\frac{1}{2}$, or 1.

Other works on persistency are as follows. Hochbaum (1983) used the persistency results of Nemhauser and Trotter (1975) in heuristics dealing with the vertex packing problem, as well as related vertex cover and set packing problems. In each case, she employed a preprocessing step of fixing variables via persistency. Bourjolly (1990) used nonsingular linear transformations between the vertex packing problem and each of the max-cut polytope and the Boolean quadric polytope which preserve at optimality the x variable values, to invoke Nemhauser and Trotter's (1975) result and establish persistency for these related problems. As a consequence, he affirmatively answered an open question of Padberg (1989) as to whether the persistency property holds for the Boolean quadric polytope. He also showed that the persistency results of Hammer *et al.* (1982) extend to these two problem classes. Hochbaum *et al.* (1993) extended the work of Nemhauser and Trotter (1975) to include problems with constraints of the form $x_i + x_j \geq 1$ and $x_i \leq x_j$. They also showed a logical equivalence between this problem class and any integer program with at most two variables per inequality, so that persistency can be applied to such integer problems. Both these latter works rely on an extreme point characterization stipulating that all components of their respective polytopes are 0, $\frac{1}{2}$, or 1.

Lu and Williams (1987) posed persistency results for higher-order (nonquadratic) 0-1 polynomial programs. For the family of unconstrained 0-1 polynomial programs expressed in maximization form, these authors used a suitable complementation of variables to rewrite the polynomial objective so that all coefficients are nonnegative. They then constructed a linear reformulation of the problem, equivalent to the nonlinear problem at all optimal solutions, so that the constraints are of the form $x_i + x_j \leq 1$, $x_i + x_j \geq 1$, and $x_i \leq x_j$. This yields a persistent formulation.

The results presented within this chapter unify and generalize all published persistent formulations, and employ completely different methods of proof. Recall that the most general persistent formulation in Sections 6.1 and 6.2 is Problem RIP(2). Based on the equivalence between Problems RIP$'$(2) and C01 established in the proof of Theorem 6.5, the persistency results of Nemhauser and Trotter (1975) and Hochbaum et al. (1993) fall out of Theorem 6.4 C where, unlike these other works, quadratic terms can appear in the objective function. This latter theorem also motivates the result of Lu and Williams (1987). Moreover, as the convex hull of feasible solutions to Problem IP(2) defines the Boolean quadric polytope of Padberg (1989), the result of Bourjolly (1990) emerges. The result of Hammer et al. (1982) identifying variables that take certain binary values in all optimal solutions also appears in Theorem 6.5. Interestingly, as opposed to relying on Balinski's (1965) extreme point characterization prevalent throughout these other efforts, the proofs in Sections 6.1 and 6.2 instead use the highest-level RLT convex hull representation to verify persistency.

Not only does Section 6.3 (Subsections 6.3.1 and 6.3.2 inclusive) pose a more general family of persistent formulations than Sections 6.1 and 6.2, it provides an altogether different method of proof, instead relying on inequality (6.39). Recall that Problem RIP(2) from Section 6.2 is a special case of Problem CBP when the degree of Problem RPP is restricted to $t = 2$. Consequently, the persistent formulations available in the literature can also be explained in terms of Theorem 6.9 and Corollary 6.2. In fact, the higher-level restrictions can be shown to envelope the polynomial inequalities of Lu and Williams (1987).

As a final comment, we note that not only do the constraints of Problem CBP admit persistency, but they also have a special structure that can be exploited in identifying integer optimal solutions. Upon solving the continuous relaxation of such a problem, one can determine whether a gap exists between the objective function values to the continuous and integer problems by checking the consistency of a 0-1 quadratic posiform having no more that $2n(n-1)$ terms. While such a study is beyond the scope of this chapter, we point out here that this result generalizes related work on roof duality by Adams *et al.* (1990); on node-weighted bidirected graphs that promote constraints of the form $x_i + x_j \leq 1$, $x_i + x_j \geq 1$, and $x_i \leq x_j$ by Bourjolly (1988); and on the Boolean quadric and max cut polytopes by Bourjolly (1990). The reader is referred to Lassiter and Adams (1997) for details.

PART II

CONTINUOUS NONCONVEX PROGRAMS

7

RLT-BASED GLOBAL OPTIMIZATION ALGORITHMS FOR NONCONVEX POLYNOMIAL PROGRAMMING PROBLEMS

Thus far, we have considered the generation of tight relaxations leading to the convex hull representation for linear and nonlinear (polynomial) discrete mixed-integer programming problems using the Reformulation-Linearization Technique (RLT). It turns out that because of its natural facility to enforce relationships between different polynomial terms, RLT can be gainfully employed to obtain global optimal solutions for *continuous*, nonconvex, polynomial programming problems as well. In this context, although a hierarchy of nested, tighter relaxations can be generated, except in special cases, convex hull representations are not necessarily produced at any particular level, or even in the limit as the number of applications tends to infinity. (See Chapter 8 for a discussion on a special case of bilinear programming problems for which RLT does produce convex hull or convex envelope representations.)

Note that the main focus of RLT in the present context is to produce tight *polyhedral* outer approximations or *linear programming* relaxations for the underlying nonlinear, nonconvex polynomial program, and as such, its principal purpose is to produce tight lower bounds on the problem. (Over the next two chapters, we modify RLT to generate convex, rather than simply linear, relaxations by incorporating certain useful convex variable bounding restrictions that enhance the quality of bounds obtained, while preserving ease in solving the corresponding relaxations.) The key to obtaining global optimal solutions is to embed this RLT construct in a branch-and-bound framework, coordinating with it a suitable partitioning strategy that would enable the gap between the lower and upper bounds thus generated to tend to zero in the limit. More specifically, this approach either solves the given polynomial program finitely, or else, an infinite branch-and-bound enumeration tree is generated such that along any infinite branch, any accumulation point of the sequence of solutions generated for the associated node relaxations solves the underlying polynomial program.

We begin our discussion in this chapter by first developing in Section 7.1, a global optimization algorithm for polynomial programming problems having integral exponents for all the nonlinear terms. Following this, in Section 7.2, we extend our analysis to accommodate situations in which variables might have rational exponents. As we shall see, in order to use RLT in such cases, we will need to insert an intermediate level of approximation, interposing an integer-exponent lower bounding polynomial program in order to solve this problem.

7.1. Polynomial Programs Having Integral Exponents

A polynomial programming problem seeks a minimum to a polynomial objective function subject to a set of polynomial constraint functions, all defined in terms of some bounded, continuous decision variables. A mathematical formulation of this problem is given below.

$$\textbf{PP}(\Omega)\text{: Minimize } \{\phi_0(x): \ x \in Z \cap \Omega\} , \tag{7.1a}$$

where

$$Z = \{x: \ \phi_r(x) \geq \beta_r \text{ for } r = 1,\ldots,R_1, \ \phi_r(x) = \beta_r, \text{ for } r = R_1 + 1,\ldots,R\}, \tag{7.1b}$$

$$\Omega = \{x: \ 0 \leq \ell_j \leq x_j \leq u_j < \infty, \text{ for } j = 1,\ldots,n\}, \tag{7.1c}$$

and where

$$\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[\prod_{j \in J_{rt}} x_j \right] \text{ for } r = 0,1,\ldots,R. \tag{7.1d}$$

Here, T_r is an index set for the terms defining $\phi_r(\cdot)$ and α_{rt} are real coefficients for the polynomial terms $(\prod_{j \in J_{rt}} x_j)$, $t \in T_r$, $r = 0,1,\ldots,R$. Note that we permit a repetition of indices within each set J_{rt}. For example, if $J_{rt} = \{1,2,2,3\}$, then the corresponding polynomial term is $x_1x_2^2x_3$. Hence, all the variables have integral exponents. In particular, let us denote $N = \{1,\ldots,n\}$, and let δ be the maximum degree of any polynomial term appearing in $\text{PP}(\Omega)$. Define $\overline{N} = \{N,\ldots,N\}$ to be composed of δ replicates of N. Then, each $J_{rt} \subseteq \overline{N}$, with $1 \leq |J_{rt}| \leq \delta$, for $t \in T_r$, $r = 0, 1, \ldots, R$.

Problems of this type find a wide range of applications in production planning, location, and distribution contexts (see Horst and Tuy, 1993) in risk management problems (see Sherali *et al.*, 1994), and in various chemical process design (pooling and blending) and engineering design situations [see Duffin *et al.* (1967), Peterson (1976), Floudas and Pardalos (1990), Floudas and Visweswaran (1990, 1995), Lasdon *et al.* (1979), and Shor (1990)].

To solve PP(Ω), we propose a branch-and-bound algorithm that utilizes specially constructed linear bounding problems using a Reformulation Linearization Technique (RLT). In this approach, we generate nonlinear implied constraints by taking the products of bounding terms in Ω up to a suitable order, and also possibly, products of other defining constraints of the problem. The resulting problem is subsequently linearized by defining new variables, one for each nonlinear term appearing in the problem. The straightforward mechanics of RLT automatically creates outer linearizations that approximate the closure of the convex hull of the feasible region.

More specifically, given Ω, in order to construct a linear programming bounding problem LP(Ω) using RLT, we begin by generating implied constraints using *distinct* products called *bound-factor products* of the *bounding factors* $(x_j - \ell_j) \geq 0$, $(u_j - x_j) \geq 0$, $j \in N$, taken δ at a time. These product constraints are of the form

$$F_\delta(J_1, J_2) \equiv \prod_{j \in J_1} (x_j - \ell_j) \prod_{j \in J_2} (u_j - x_j) \geq 0, \tag{7.2}$$

where $J_1 \cup J_2 \subseteq \overline{N}$, and $|J_1 \cup J_2| = \delta$. By using suitable dummy or blank slot choices to represent repetitions, it is readily verified that the number of distinct constraints of type (7.2) is given by

$$\sum_{k=0}^{\delta} \binom{n+k-1}{k} \binom{n+(\delta-k)-1}{(\delta-k)}.$$

Accordingly, the RLT process proceeds as follows.

Reformulation Phase

Include the constraints (7.2) in the problem PP(Ω).

Linearization Phase

In the reformulated augmented polynomial programming, substitute

$$X_J = \prod_{j \in J} x_j \qquad \forall\ J \subseteq \overline{N}, \tag{7.3}$$

where the indices in J are assumed to be sequenced in nondecreasing order, and where $X_{\{j\}} \equiv x_j\ \forall\ j \in N$, and $X_\emptyset \equiv 1$. The number of X-variables defined here, besides $X_{\{j\}}$, $j \in N$, and $X_\emptyset$ is $\binom{n+\delta}{\delta} - (n+1)$. Note that each distinct set J produces one distinct X_J variable, and that when we write $X_{(i,j,k)}$ or $X_{(J_1 \cup J_2)}$ for example, we assume that the indices within $(\cdot)$ are sequenced in nondecreasing order. This produces the required *RLT linear programming relaxation.*

Remark 7.1. *Optional Constraints for Generating Tighter Linear Programming Representations.* Evidently, the constraints of type (7.2) are implied by the set Ω prior to the linearization. However, following the linearization process, these constraints impose useful interrelationships among the product variables X_J. In a likewise manner, we can also generate additional implied constraints in the form of polynomials of degree less than or equal to δ, by taking suitable products of *constraint factors* $\phi_r(x) - \beta_r \geq 0,\ r = 1,\ldots,R_1$ and/or products of bounding factors with constraint factors wherever possible. In addition, we can multiply the equality constraints $\phi_r(x) = \beta_r,\ r = R_1 + 1,\ldots,R$, defining Z, by sets of products of variables of the type $\prod_{j \in J_1} x_j,\ J_1 \subseteq \overline{N}$, so long as the resulting polynomial expression is of degree no more than δ. Incorporating these additional constraints in LP(Ω) after making the substitutions (7.3), produces a tighter linear programming representation. Although this is theoretically admissible and may be computationally advantageous, it is not necessarily required for the results presented in this chapter. In the same spirit, as we shall be seeing in this and the next two chapters, tighter relaxations can be generated by incorporating bound or constraint factor products of order greater than δ. Of course, this would increase the size of the resulting relaxations, and must be done selectively and judiciously in order to keep the relaxations manageable. Nevertheless, we permit the inclusion of such constraints within LP(Ω) as a user or application driven option. $\square$

Lemma 7.1 below verifies that LP(Ω) is indeed a relaxation of PP(Ω), and gives an important characterization for this problem. As always, we will let $\nu[\cdot]$ denote the value at optimality of the corresponding problem $[\cdot]$.

Lemma 7.1. $\nu[\text{LP}(\Omega)] \le \nu[\text{PP}(\Omega)]$. *Moreover, if the optimal solution* (x^*, X^*) *obtained for* LP(Ω) *satisfies (7.3) for all* $J \in \cup_{r=0}^{R} \cup_{t \in T_r} \{J_{rt}\}$, *then* x^* *solves Problem* PP(Ω).

Proof. For any feasible solution $\bar{x}$ to PP(Ω), there exists a feasible solution $(\bar{x}, \bar{X})$ to LP(Ω) having the same objective function value which is constructed using the definition (7.3). Hence, $\nu[\text{LP}(\Omega)] \le \nu[\text{PP}(\Omega)]$. Moreover, if (7.3) holds for an optimal solution (x^*, X^*) to LP(Ω), for all $J \in \cup_{r=0}^{R} \cup_{t \in T_r} \{J_{rt}\}$, then x^* is feasible to PP(Ω), and $\nu[\text{LP}(\Omega)] = \sum_{t \in T_0} \alpha_{0t} X^*_{J_{0t}} = \sum_{t \in T_0} \alpha_{0t} [\prod_{j \in J_{0t}} x^*_j]$, which equals the objective value of PP(Ω) at $x = x^*$. Hence, x^* solves PP(Ω). □

Notice that LP(Ω) does not explicitly contain any constraint that can be generated by constructing products of bounding factors taken less than δ at a time. The next lemma shows that such constraints are actually implied by those already existing in LP(Ω).

Lemma 7.2. *Let* $[f(\cdot)]_L$ *denote the linearized version of a polynomial function* $f(\cdot)$ *after making the substitutions (7.3). Then, the constraints* $[F_{\delta'}(J_1, J_2)]_L \ge 0$, *where* $(J_1 \cup J_2) \subseteq \bar{N}$, $|J_1 \cup J_2| = \delta'$, $1 \le \delta' < \delta$, *are all implied by the constraints*

$[F_\delta(J_1, J_2)]_L \geq 0$ *generated via (7.2), for all distinct ordered pairs* $(J_1 \cup J_2) \subseteq \overline{N}$, $|J_1 \cup J_2| = \delta$.

Proof. For any δ', $1 \leq \delta' < \delta$, consider the surrogate of the constraints $[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_L \geq 0$ and $[F_{\delta'+1}(J_1, J_2 \cup \{t\})]_L \geq 0$, where $(J_1 \cup J_2) \subseteq \overline{N}$, $|J_1 \cup J_2| = \delta'$, and $t \in N$.

$$[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_L + [F_{\delta'+1}(J_1, J_2 \cup \{t\})]_L = [(x_t - \ell_t)F_{\delta'}(J_1, J_2)]_L$$

$$+ [(u_t - x_t)F_{\delta'}(J_1, J_2)]_L = [x_t F_{\delta'}(J_1, J_2)]_L - \ell_t [F_{\delta'}(J_1, J_2)]_L$$

$$+ u_t [F_{\delta'}(J_1, J_2)]_L - [x_t F_{\delta'}(J_1, J_2)]_L = (u_t - \ell_t)[F_{\delta'}(J_1, J_2)]_L \geq 0.$$

Since $(u_t - \ell_t) \geq 0$, it follows that $[F_{\delta'}(J_1, J_2)]_L \geq 0$ is implied by $[F_{\delta'+1}(J_1 \cup \{t\}, J_2)]_L \geq 0$ and $[F_{\delta'+1}(J_1, J_2 \cup \{t\})]_L \geq 0$. The required result follows by the principle of induction, and this completes the proof. □

The next result establishes an important interrelationship between the newly defined variables X_J, and prompts a partitioning strategy which drives the convergence argument.

Lemma 7.3. *Let* $(\bar{x}, \overline{X})$ *be a feasible solution to* LP(Ω). *Suppose that* $\bar{x}_p = \ell_p$. *Then*

$$\overline{X}_{(J \cup p)} = \ell_p \overline{X}_J \qquad \forall\ J \subseteq \overline{N}, \quad 1 \leq |J| \leq \delta - 1. \tag{7.4}$$

Similarly, if $\bar{x}_p = u_p$, *then*

$$\overline{X}_{(J \cup p)} = u_p \overline{X}_J \quad \forall\ J \subseteq \overline{N}, \quad 1 \le |J| \le \delta - 1. \tag{7.5}$$

Proof. First, consider the case $\overline{x}_p = \ell_p$. For $|J| = 1$, consider any $q \in N$ (possibly, $q \equiv p$). By Lemma 7.2, the following constraints are implied by $[(7.2)]_L$, where as before, $[\cdot]_L$ denotes the linearization of $[\cdot]$ under the substitution (7.3).

$$\begin{aligned} [(x_p - \ell_p)(x_q - \ell_q)]_L &= X_{(p,q)} - \ell_q x_p - \ell_p x_q + \ell_p \ell_q \ge 0 \\ [(x_p - \ell_p)(u_q - x_q)]_L &= -X_{(p,q)} + u_q x_p + \ell_p x_q - \ell_p u_q \ge 0. \end{aligned} \tag{7.6}$$

Hence, we get

$$\ell_q (x_p - \ell_p) + \ell_p x_q \le X_{(p,q)} \le \ell_p x_q + u_q (x_p - \ell_p). \tag{7.7}$$

By evaluating (7.7) at $(\overline{x}, \overline{X})$, we have $\overline{X}_{(p,q)} = \ell_p \overline{x}_q$.

Now, let us inductively assume that (7.4) is true for $|J| = 1, \ldots, (t-1)$, and consider $|J| = t$, where $2 \le t \le \delta - 1$. For any $q \in J$ (possibly $q \equiv p$), by Lemma 7.2, the following constraints are implied by $[(7.2)]_L$.

$$\begin{aligned} &\left[(x_p - \ell_p)(x_q - \ell_q) \prod_{j \in J-q} (x_j - \ell_j) \right]_L \ge 0 \\ &\left[(x_p - \ell_p)(u_q - x_q) \prod_{j \in J-q} (x_j - \ell_j) \right]_L \ge 0. \end{aligned} \tag{7.8}$$

Let us write $\prod_{j \in J-q} (x_j - \ell_j) = \prod_{j \in J-q} x_j + f(x)$, where $f(x)$ is a polynomial in x of degree no more than $t - 2$. Then, from (7.8), we have

$$(X_{(J\cup p)} - \ell_p X_J) \geq \ell_q (X_{(J+p-q)} - \ell_p X_{(J-q)}) + [\ell_p x_q f(x) - x_p x_q f(x)]_L$$

$$+ \ell_q [x_p f(x) - \ell_p f(x)]_L$$

$$(X_{(J\cup p)} - \ell_p X_J) \leq u_q (X_{(J+p-q)} - \ell_p X_{(J-q)}) + [\ell_p x_q f(x) - x_p x_q f(x)]_L$$

$$+ u_q [x_p f(x) - \ell_p f(x)]_L. \tag{7.9}$$

Let $(\cdot)|_{(\bar{x},\bar{X})}$ denote the function $(\cdot)$ being evaluated at $(\bar{x}, \bar{X})$. By the induction hypothesis, $\bar{X}_{(J+p-q)} = \ell_p \bar{X}_{(J-q)}$, $[x_p x_q f(x)]_L|_{(\bar{x},\bar{X})} = \ell_p [x_q f(x)]_L|_{(\bar{x},\bar{X})}$, and $[x_p f(x)]_L|_{(\bar{x},\bar{X})} = \ell_p [f(x)]_L|_{(\bar{x},\bar{X})}$. Hence, when we evaluate (7.5) at $(\bar{x}, \bar{X})$, the right-hand sides of both the inequalities become zero, and this gives $\bar{X}_{(J\cup p)} = \ell_p \bar{X}_J$.

The case for $\bar{x}_p = u_p$ can be similarly proven by using

$$\left[(u_p - x_p)(u_q - x_q) \prod_{j\in J-q} (x_j - \ell_j)\right]_L \geq 0$$

$$\left[(u_p - x_p)(x_q - \ell_q) \prod_{j\in J-q} (x_j - \ell_j)\right]_L \geq 0 \tag{7.10}$$

in place of (7.8), and this completes the proof. □

7.1.1. A Branch-and-Bound Algorithm

We are now ready to imbed $\mathrm{LP}(\Omega)$ in a branch-and-bound algorithm to solve $\mathrm{PP}(\Omega)$. The procedure involves the partitioning of the set Ω into sub-hyperrectangles, each of which is associated with a node of the branch-and-bound tree. Let $\Omega' \subseteq \Omega$ be any such partition. Then, $\mathrm{LP}(\Omega')$ gives a lower bound for the node subproblem $\mathrm{PP}(\Omega')$. In particular, if $(\bar{x}, \bar{X})$ solves $\mathrm{LP}(\Omega')$ and satisfies (7.3) for all $J \in \cup_{r=0}^{R} \cup_{t \in T_r} \{J_{rt}\}$, then by Lemma 7.1, $\bar{x}$ solves $\mathrm{PP}(\Omega')$, and being feasible to $\mathrm{PP}(\Omega')$, the value $\nu[\mathrm{PP}(\Omega')] \equiv \nu[\mathrm{LP}(\Omega')]$ provides an upper bound for the problem $\mathrm{PP}(\Omega)$. Hence, we have a candidate for possibly updating the incumbent solution x^* and its value ν^* for $\mathrm{PP}(\Omega)$. In any case, if $\nu[\mathrm{PP}(\Omega')] \geq \nu^*$, we can fathom the node associated with Ω'.

Hence, at any stage s of the branch-and-bound algorithm, we have a set of non-fathomed or *active nodes* induced by $q \in Q_s$, say, each associated with a corresponding partition $\Omega^q \subseteq \Omega$. For each such node, we will have computed a lower bound LB_q via the solution of the linear program $\mathrm{LB}(\Omega^q)$. As a result, the lower bound on the overall problem $\mathrm{PP}(\Omega)$ at stage s is given by $\mathrm{LB}(s) = \text{minimum}\{\mathrm{LB}_q : q \in Q_s\}$. Whenever the lower bounding solution for any node subproblem turns out to be feasible to $\mathrm{PP}(\Omega)$, we update the incumbent solution x^* and its objective value ν^* as mentioned above. Additionally, if $\mathrm{LB}_q \geq \nu^*$, we fathom node q. Hence, the active nodes all satisfy $\mathrm{LB}_q < \nu^* \;\; \forall\, q \in Q_s$, for each stage s. We now select an active node $q(s)$ that yields the *least lower bound* among the nodes $q \in Q_s$, i.e., for which $\mathrm{LB}_{q(s)} \equiv \mathrm{LB}(s)$, and proceed by decomposing the corresponding hyperrectangle $\Omega^{q(s)}$ into two subhyperrectangles, based on a *branching variable* x_p selected according to the following rule.

BRANCHING RULE:

$$p \in \arg\max_{j \in N}\{\theta_j\},$$

where

$$\theta_j = \underset{t=1,\ldots,\delta-1}{\text{maximum}} \; \underset{\substack{J \subseteq \overline{N} \\ |J|=t}}{\text{maximum}}\{|\hat{X}_{(J \cup j)} - \hat{x}_j \hat{X}_J|\} \text{ for each } j \in N, \tag{7.11}$$

and where $(\hat{x}, \hat{X})$ denotes the optimal solution obtained for the corresponding node subproblem.

Note that the partitioning rule (7.11) is based on identifying the variable that yields the largest discrepancy between a new RLT variable and an associated nonlinear product that this variable represents. The idea is to drive all such discrepancies to zero. A formal statement of a procedure that accomplishes this is given below.

Step 0: Initialization. Initialize the incumbent solution $x^* = \varnothing$, and let the incumbent objective value be $v^* = \infty$. (If a feasible solution is known, the corresponding values of x^* and v^* can be used instead.) Set $s = 1$, $Q_s = \{1\}$, $q(s) = 1$, and $\Omega^1 \equiv \Omega$. Solve LP(Ω) and let $(\hat{x}, \hat{X})$ be the solution obtained of objective value $\text{LB}_1 = v[\text{LP}(\Omega^1)]$. If $\hat{x}$ is feasible to PP(Ω), update x^* and v^*, if necessary, and if $\text{LB}_1 = v^*$, then stop; x^* solves PP(Ω). Otherwise, determine a branching variable x_p by using (7.11), and note by Lemma 7.1 that we must now have $\theta_p > 0$. Proceed to Step 1.

Step 1: Partitioning Step. Partition the selected active node $\Omega^{q(s)}$ into two sub-hyperrectangles by splitting the current bounding interval for x_p at the value $\hat{x}_p$. (Note that by Lemma 7.3, since $\theta_p > 0$, we have that $\hat{x}_p$ lies in the interior of this interval.) Replace $q(s)$ by these two new node indices to revise Q_s.

Step 2: Bounding Step. Solve the RLT linear programming relaxation for each of the two new nodes generated. Update the incumbent solution if possible, and determine a corresponding branching variable index using (7.11) for each of these nodes, as done for node 1 in the Initialization step.

Step 3: Fathoming Step. Fathom any nonimproving nodes by setting $Q_{s+1} = Q_s - \{q \in Q_s\text{: } \mathrm{LB}_q + \varepsilon \geq v^*\}$, where $\varepsilon \geq 0$ is some selected optimality tolerance ($\varepsilon \equiv 0$ if an exact optimum is desired). If $Q_{s+1} = \varnothing$, then stop. Otherwise, increment s by one and proceed to Step 4.

Step 4: Node Selection Step. Select an active node $q(s) \in \arg\min\{\mathrm{LB}_q :$ $q \in Q_s\}$, and return to Step 1.

Theorem 7.1. *(Convergence Result). The above algorithm (run with $\varepsilon \equiv 0$) either terminates finitely with the incumbent solution being optimal to* $\mathrm{PP}(\Omega)$, *or else an infinite sequence of stages is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the x-variable part of the sequence of linear programming relaxation solutions generated for the node subproblems solves* $\mathrm{PP}(\Omega)$.

Proof. The case of finite termination is clear. Hence, suppose that an infinite sequence of stages is generated. Consider any infinite branch of the branch-and-bound tree, and denote the associated nested sequence of partitions as $\{\Omega^{q(s)}\}$, corresponding to a set of stages s in some index set S. Hence,

$$\mathrm{LB}(s) = \mathrm{LB}_{q(s)} \equiv \nu[\mathrm{LP}(\Omega^{q(s)})] \quad \forall\ s \in S. \tag{7.12}$$

For each node $q(s)$, $s \in S$, let $(x^{q(s)}, X^{q(s)})$ denote the optimum solution obtained for $\mathrm{LP}(\Omega^{q(s)})$. Furthermore, let $\ell^{q(s)}$ and $u^{q(s)}$ be the associated vectors of lower and upper bounds on the variables that define the hyperrectangle $\Omega^{q(s)}$. By taking any convergent subsequence if necessary, using the boundedness of the sequences generated, assume without loss of generality that

$$\{x^{q(s)}, X^{q(s)}, \ell^{q(s)}, u^{q(s)}\}_S \to (x^*, X^*, \ell^*, u^*). \tag{7.13}$$

We must show that x^* solves $\mathrm{PP}(\Omega)$.

First of all, note that since $\mathrm{LB}_{q(s)}$ is the least lower bound at stage s, we have,

$$\nu[\mathrm{PP}(\Omega)] \geq \mathrm{LB}_{q(s)} = \phi_L(x^{q(s)}, X^{q(s)}) \quad \forall\ s \in S, \tag{7.14}$$

where $\phi_L(x, X)$ is the linearized objective function for $\mathrm{LP}(\Omega)$.

Next, observe that over the infinite sequence of nodes $\Omega^{q(s)}$, $s \in S$, there exists a variable x_p that is branched on infinitely often via the choice (7.11). Associated with x_p, there must be some index set $J_\infty \subseteq \overline{N}$, $1 \leq |J_\infty| \leq \delta - 1$, which occurs along with

p infinitely often in determining θ_p. Let $S_1 \subseteq S$ be the subsequence of stages over which this set J_∞ determines θ_p in (7.11). Then, by (7.11) for each $s \in S_1$, we have

$$\left|X^{q(s)}_{(J_\infty \cup p)} - x^{q(s)}_p X^{q(s)}_{J_\infty}\right| \geq \left|X^{q(s)}_{(J \cup j)} - x^{q(s)}_j X^{q(s)}_J\right|$$

$$\forall\ J \subseteq \overline{N},\quad |J| = 1,\ldots,\delta - 1,\quad j = 1,\ldots,n. \tag{7.15}$$

Now, by (7.13), we have that

$$\{\Omega^{q(s)}\} \to \Omega^* \equiv \{x:\ \ell^* \leq x \leq u^*\} \tag{7.16}$$

and that (x^*, X^*) is feasible to LP(Ω^*). Moreover, by virtue of the partitioning scheme, we know that for each $s \in S_1$, $x^{q(s)}_p \notin (\ell^{q(s')}_p, u^{q(s')}_p)$ for all $s' \in S_1$, $s' > s$, while $x^*_p \in [\ell^*_p, u^*_p]$. Hence, we must have that $x^*_p = \ell^*_p$ or $x^*_p = u^*_p$.

By Lemma 7.3, we get

$$X^*_{(J_\infty \cup p)} = x^*_p X^*_{J_\infty}. \tag{7.17}$$

But this means from (7.15) that as $s \to \infty$, $s \in S_1$, we have

$$X^*_{(J \cup j)} = x^*_j X^*_J \quad J \subseteq \overline{N},\ |J| = 1,\ldots,\delta - 1, \text{ and } j = 1,\ldots,n. \tag{7.18}$$

Hence, the definitions (7.3) hold true for (x^*, X^*). Therefore, x^* is feasible to PP(Ω), and moreover,

$$\phi_0(x^*) = \phi_L(x^*, X^*) \geq v[\mathrm{PP}(\Omega)]. \tag{7.19}$$

Noting that (7.14) implies upon taking limits as $s \to \infty$, $s \in S_1$, that $v[\mathrm{PP}(\Omega)] \geq \phi_L(x^*, X^*)$, we deduce using (7.19) that $v[\mathrm{PP}(\Omega)] = \phi_0(x^*)$, and so x^* is optimal to $\mathrm{PP}(\Omega)$. This completes the proof. □

Remark 7.2. *Special Cases.* Note that in the spirit of the foregoing algorithmic scheme, we can retain the flexibility of exploiting certain inherent special structures in designing admissible convergent variants of this procedure. For example, consider a trilinear programming problem (see Zikan, 1990, for an application in the context of tracking trajectories) in which $\delta = 3$, with any third-order cross-product term being of the type $x_i x_j x_k$ for $1 \leq i \leq n_1$, $n_1 + 1 \leq j \leq n_2$, and $n_2 + 1 \leq k \leq n$, and similarly, any second-order cross-product term being of the form $x_i x_j$ for i and j lying in two different index sets from among $\{1, \ldots, n_1\}$, $\{n_1 + 1, \ldots, n_2\}$, and $\{n_2 + 1, \ldots, n\}$. For this problem, we would need to generate in (7.2) only those bound-factor products of order 3 that involve one variable from each index set. Then, Lemma 7.1 holds as stated, and in Lemma 7.2, the corresponding second-order bound-factor product constraints generated by indices from two different sets are also implied. Consequently, Lemma 7.3 holds with $(J \cup p)$ having at most one index per index set. Accordingly, in (7.11), we only need to consider those $(J \cup j)$ that have at most one index from each index set, and the convergence of the resulting algorithm continues to hold by Theorem 7.1.

Remark 7.3. *Alternate Branching Variable Selection Rule.* In light of Lemma 7.1 and the proof of Theorem 7.1, observe that we could have restricted in (7.11) the evaluation of only those quantities $|\hat{X}_{(J \cup j)} - \hat{x}_j \hat{X}_J|$ for which the product $\prod_{i \in (J \cup j)} x_i$ appears in some term, or as a subset of some term, in the problem. Then, by the argument associated with (7.15) in the proof of Theorem 7.1 and Lemma 7.1, the convergence of the algorithm would continue to hold true.

7.1.2. An Illustrative Example

To illustrate the branch-and-bound algorithm of the previous section, we will solve the following nonconvex polynomial program of order $\delta = 3$.

$$\textbf{PP}(\Omega)\textbf{:}\quad \text{Minimize } \phi_0(x) = x_1x_2x_3 + x_1^2 - 2x_1x_2 - 3x_1x_3 + 5x_2x_3 - x_3^2 + 5x_2 + x_3$$

$$\text{subject to } 4x_1 + 3x_2 + x_3 \le 20 \tag{7.20}$$

$$x_1 + 2x_2 + x_3 \ge 1$$

$$2 \le x_1 \le 5,\ \ 0 \le x_2 \le 10,\ \ 4 \le x_3 \le 8.$$

At stage $s = 1$, $Q_1 = \{1\}$, $v^* = \infty$, and $\Omega^1 \equiv \Omega = \{x\colon\ 2 \le x_1 \le 5,\ 0 \le x_2 \le 10,\ 4 \le x_3 \le 8\}$. The corresponding linear program LP(Ω) has 56 constraints of type (7.2), linearized by using the substitution (7.3). Two of these constraints are given below as an example.

(i) $J_1 = \{1,2,3\}, J_2 = \varnothing$: $[(x_1 - 2)(x_2)(x_3 - 4)]_L \ge 0$

yielding $X_{123} - 4X_{12} - 2X_{23} + 8x_2 \ge 0$,

(ii) $J_1 = \{1,3\}, J_2 = \{2\}$: $[(x_1 - 2)(x_3 - 4)(10 - x_2)]_L \ge 0$

yielding $X_{123} - 4X_{12} - 10X_{13} - 2X_{23} + 40x_1 + 8x_2 + 20x_3 \le 80$.

Aside from such newly generated constraints, LP(Ω^1) contains the original functional constraints of PP(Ω) linearized via (7.3), along with Ω^1 itself, in its constraint set, and has the objective function

$$\phi_L(x, X) = X_{123} + X_{11} - 2X_{12} - 3X_{13} + 5X_{23} - X_{33} + 5x_2 + x_3.$$

Note that the entire set of variables (x, X) in the problem is given by

$$(x, X) = x_1, x_2, x_3, X_{11}, X_{12}, X_{13}, X_{22}, X_{23}, X_{33}.$$

$$X_{111}, X_{112}, X_{113}, X_{122}, X_{123}, X_{133}, X_{222}, X_{223}, X_{233}, X_{333}).$$

Upon solving LP(Ω^1), we obtain,

$$(x^1, X^1) = (3, 0, 8, 8, 0, 24, 0, 0, 64, 20, 0, 64, 0, 0, 192, 0, 0, 0, 512)$$

$$v[\mathrm{LP}(\Omega^1)] = -120.$$

Note that since the constraints of (7.20) are linear, x^1 is feasible to (7.20), so that $\phi_0(x^1) = -119$ is an upper bound on the optimum to (7.20). Hence, the current incumbent solution is $x^* = (3, 0, 8)$ and $v^* = -119$. Using (7.11), we have, $\theta_1 = | X^1_{113} - x^1_1 X^1_{13} | = 8$, and $\theta_2 = \theta_3 = 0$. (If we use Remark 7.3 given at the end of the previous section, then $\theta_1 = | X^1_{11} - x^1_1 x^1_1 | = 1$.) With x_1 as the branching variable $(p = 1)$, we partition Ω^1 into the sub-hyperrectangles

$$\Omega^2 = \{x: \ 2 \le x_1 \le 3, 0 \le x_2 \le 10, 4 \le x_3 \le 8\}$$

$$\Omega^3 = \{x: \ 3 \le x_1 \le 5, 0 \le x_2 \le 10, 4 \le x_3 \le 8\},$$

and set $Q_1 = \{2, 3\}$ at Step 1. Then at Step 2, for node (2), $\mathrm{LP}(\Omega^2)$ gives

$$(x^2, X^2) = (3, 0, 8, 9, 0, 24, 0, 0, 64, 27, 0, 72, 0, 0, 192, 0, 0, 0, 512),$$

$$v[\mathrm{LP}(\Omega^2)] = -119,$$

and for node 3, $\mathrm{LP}(\Omega^3)$ gives the same solution as for $\mathrm{LP}(\Omega^2)$. By using this common solution in (7.11), we get $\theta_1 = \theta_2 = \theta_3 = 0$. Hence, this solution is feasible to $\mathrm{PP}(\Omega)$, but it does not improve the incumbent value. At the fathoming step (Step 3), we fathom the nodes 2 and 3, and since the list of active nodes is now empty, the solution $x^* = (3, 0, 8)$ solves the given problem (7.20).

Finally, let us illustrate the comment given in Remark 7.1. Suppose that in addition to the bound-factor constraints generated above, we also generated the following constraints for the relaxation at node 1:

$$[(u_i - x_i)(u_j - x_j)(20 - 4x_1 - 3x_2 - x_3)]_L \geq 0 \quad \text{for} \quad 1 \leq i \leq j \leq 3.$$

Note that this is a particular restricted set of additional constraints generated in the spirit of Remark 7.1. Then, it so happens that the augmented linear program $\mathrm{LP}(\Omega^1)$ itself yields the optimal solution to (7.20), with no branching required in this instance.

7.2. Polynomial Programs Having Rational Exponents

In the foregoing section, we considered polynomial programming problems in which the variables have integral exponents for all nonlinear terms. However, there are several applications in which polynomial programs arise wherein the variable exponents are in general rational, but non-integral. Examples of such instances include water distribution network design problems (Sherali and Smith, 1995), location-allocation problems using more accurate, empirically determined, ℓ_p distance measures (Brimberg and Love, 1991), several engineering design applications such as the design of heat exchangers and pressure vessels as described in Floudas and Pardalos (1990, Chapters 4, 7, and 11), as well as other design, equilibrium, and economics problems that are typically modeled as geometric programming problems (see Kortanek *et al.*, 1995, and the references cited therein). Furthermore, as noted by Hansen and Jaumard (1992), polynomial programs can be used to approximate various problems that include trigonometric and transcendental functions via Taylor or MacLaurin series expansions. Hence, our methodology can be potentially extended to handle such problems as well.

Aside from algorithms developed for geometric programs that are based on logarithmic/exponential transformations under the assumption that the variables are restricted to be positive valued (see Kortanek *et al.*, 1995 and Maranas and Floudas, 1994, for example), other possible approaches for solving such problems are based on the use of interval arithmetic (see Hansen *et al.,* 1993) or homotopy methods (see Watson *et al.*, 1987, and Kostreva and Kinard, 1991). However, these latter approaches require the determination of all solutions to the Fritz John necessary optimality conditions in order to recover a global optimum, which can be an arduous task.

In this section, we extend the RLT approach of the previous section to handle rational, instead of simply integer, exponents. Similar to Section 7.1, a branch-and-bound algorithm is developed that solves a sequence of relaxations over partitioned subsets of Ω in order to find a global optimal solution. However, to generate the relaxation for each node subproblem, an additional initial step is introduced that constructs an approximating polynomial program having integer exponents to which RLT is subsequently applied. Furthermore, in order to ensure convergence to a global optimum, special partitioning procedures are proposed to coordinate the two levels of relaxations that are involved in this scheme. This gives the basic structure of the algorithm to which several expedients and reduction or bound tightening strategies can be applied as discussed later in Chapters 8 and 9 in order to enhance the solution procedure.

To present this proposed modification, let us restate the polynomial program PP(Ω) defined in (7.1) as follows, to account for possible rational exponents.

$$\textbf{PP}(\boldsymbol{\Omega})\textbf{:}\ \text{Minimize} \quad \phi_0(x) \tag{7.21a}$$

$$\text{subject to} \quad \phi_i(x) \le \beta_i \quad \text{for } i = 1,\dots,m \tag{7.21b}$$

$$x \in \Omega \equiv \{x : \quad 0 \le \ell_j \le x_j \le u_j < \infty \quad \forall\ j \in N \equiv \{1,\dots,n\}\}, \tag{7.21c}$$

where,

$$\phi_i(x) = \sum_{t \in T_i} \alpha_{it} \prod_{j \in J_{it}} x_j^{\gamma_{itj}} \quad \text{for all } i = 0,1,\dots,m. \tag{7.21d}$$

Here, the objective function as well as each constraint is represented by a polynomial $\phi_i(x)$ that is comprised of terms indexed by a set T_i, where each term $t \in T_i$ has some real coefficient α_{it} that may be of either sign, and is composed of products of (generalized) monomials $x_j^{\gamma_{itj}}$ for j belonging to some subset J_{it} of N. The indices in J_{it} are assumed to be distinct, each γ_{itj} is assumed to be positive and rational, and $\ell_j < u_j \ \forall \ j = 1, \ldots, n$. For convenience in exposition, any equality constraint is assumed to be treated here as an equivalent pair of oppositely restricted inequalities.

The principal construct in the development of a solution procedure for solving Problem PP(Ω) is the construction of a tight linear programming relaxation for obtaining lower bounds for this problem, as well as for its partitioned subproblems. For convenience in exposition, let us assume for now that Ω represents either the initial bounds on the variables of the problem, or modified bounds as defined for some partitioned subproblem in a branch-and-bound scheme. The proposed strategy for generating this linear programming relaxation is to apply RLT as in Section 7.1, but with an additional initial step that generates an approximating polynomial program having integer variable exponents.

Toward this end, let us define the fractional part of each exponent γ_{itj} by f_{itj}, where,

$$\gamma_{itj} = \left\lfloor \gamma_{itj} \right\rfloor + f_{itj}, \quad 0 \leq f_{itj} < 1 \quad \forall \ i = 0, 1, \ldots, m, \ t \in T_i, \text{ and } j \in J_{it}. \tag{7.22}$$

Accordingly, for each term, let us denote the set of variables having fractional exponents by

$$J_{it}^{+} = \{j \in J_{it} : \quad f_{itj} > 0\} \qquad \forall \ i, t. \tag{7.23}$$

Now, let $C_{\Omega}(\cdot)$ denote the convex envelope of $(\cdot)$ over the hyperrectangle Ω (see Horst and Tuy, 1993, for example). In particular, for each $j \in J^{+}_{it}$, let $C_{\Omega}(x_j^{f_{itj}})$ represent the (affine) convex envelope of $x_j^{f_{itj}}$ over the interval $\ell_j \le x_j \le u_j$. This function is given by (see Figure 7.1)

$$C_{\Omega}(x_j^{f_{itj}}) = \ell_j^{f_{itj}} + \frac{(x_j - \ell_j)}{(u_j - \ell_j)}\left[u_j^{f_{itj}} - \ell_j^{f_{itj}}\right]. \tag{7.24}$$

Additionally, for each $j \in J^{+}_{it}$, let $\bar{x}_{itj}$ be some point in the (open) interval (ℓ_j, u_j). Then, by the concavity of $x_j^{f_{itj}}$, we have (see Figure 7.1),

$$C_{\Omega}(x_j^{f_{itj}}) \le x_j^{f_{itj}} \le \bar{x}_j^{f_{itj}} + \left[f_{itj}\bar{x}_{itj}^{(f_{itj}-1)}\right](x_j - \bar{x}_{itj}). \tag{7.25}$$

Consequently, from (7.22) and (7.25), we have,

$$g_{itj}^{\Omega_j}(x_j) \le x_j^{\gamma_{itj}} \le h_{itj}^{\Omega_j}(x_j) \quad \text{for all } j \in J^{+}_{it}, \quad \forall\ i, t \tag{7.26a}$$

where $\Omega_j \equiv \{x_j : \ell_j \le x_j \le u_j\}\ \forall\ j$, and where for each (i, t),

$$g_{itj}^{\Omega_j}(x_j) \equiv \begin{cases} x_j^{\lfloor \gamma_{itj} \rfloor} C_{\Omega}(x_j^{f_{itj}}) & \forall\ j \in J^{+}_{it} \\ x_j^{\gamma_{itj}} & \forall\ j \in J_{it} - J^{+}_{it} \end{cases} \tag{7.26b}$$

and

$$h_{itj}^{\Omega_j}(x_j) \equiv \begin{cases} x_j^{\lfloor \gamma_{itj} \rfloor}\left\{\bar{x}_{itj}^{f_{itj}} + \left[f_{itj}\bar{x}_{itj}^{(f_{itj}-1)}\right](x_j - \bar{x}_{itj})\right\} & \forall\ j \in J^{+}_{it} \\ x_j^{\gamma_{itj}} & \forall\ j \in J_{it} - J^{+}_{it}\,. \end{cases} \tag{7.26c}$$

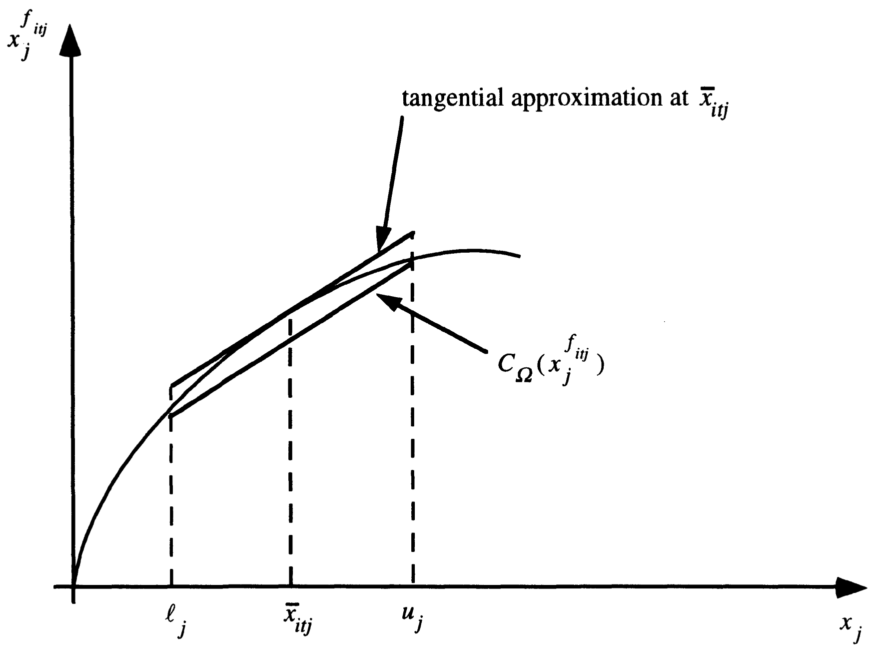


Figure 7.1. Bounding affine approximations for $x_j^{f_{itj}}$.

In order to balance the lower and upper approximating functionals in (7.26a) (see Figure 7.1), we prescribe $\bar{x}_{itj}$ to be selected as the point for which the corresponding first-order tangential support yields the same discrepancy at both the end points ℓ_j and u_j with respect to the function $x_j^{f_{itj}}$. This point is easily computed to be

$$\bar{x}_{itj} = \left[\frac{f_{itj}(u_j - \ell_j)}{(u_j^{f_{itj}} - \ell_j^{f_{itj}})}\right]^{1/(1-f_{itj})} \quad \forall\, i, t, \text{ and } j \in J_{it}^{+}. \tag{7.27}$$

We also comment here that while only one point of tangential support has been suggested in our derivation above, some special applications might permit the use of more than one tangential approximating constraint. (In general, such a strategy could result in a combinatorial population of constraints.) For example, in the water distribution network design model described in Sherali and Smith (1995), each polynomial constraint involves only a single variable having a rational exponent for which the tangential supporting approximation is necessary, and in this instance, additional supports can be generated to obtain tighter relaxations, while maintaining a manageable number of constraints.

Furthermore, in the spirit of (7.26) and (7.28)-(7.30), various bounding functions can be devised for the monomials $x_j^{\gamma_{itj}}$, $j \in J_{it}^+ \ \forall\ (i,t)$, and can be suitably coordinated with a partitioning strategy as introduced in the sequel, in order to induce convergence. For example, denoting $C^{\Omega}(\cdot)$ as the concave envelope, over Ω, we can use

$$h_{itj}^{\Omega_j}(x_j) = x_j^{\lfloor \gamma_{itj} - 1 \rfloor} C^{\Omega}\left(x_j^{\{\gamma_{itj} - \lfloor \gamma_{itj} - 1 \rfloor\}} \right) \quad \forall\ j \in J_{it}^+ \ni \gamma_{itj} > 1,\ \forall\ (i,t),$$

where C^{Ω} is an affine function in this case. This type of an approximation was used in lieu of (7.26c) in the application described by Sherali and Smith (1995), and yielded comparatively favorable results. In general, one might expect this to be the case when the polynomial expressions in the objective or constraint functions have terms that involve dissimilar variables.

Now, putting (7.26) and (7.21d) together, we obtain that

$$\alpha_{it} \prod_{j \in J_{it}} x_j^{\gamma_{itj}} \geq \phi_{it}^{R(\Omega)}(x) \equiv \begin{cases} \alpha_{it} \prod_{j \in J_{it}} g_{itj}^{\Omega_j}(x_j) & \text{if } \alpha_{it} > 0 \\ \alpha_{it} \prod_{j \in J_{it}} h_{itj}^{\Omega_j}(x_j) & \text{if } \alpha_{it} < 0 \end{cases} \qquad \forall i, t. \tag{7.28}$$

Hence, summing (7.28) over all the terms $t \in T_i$ for each $i = 0, 1, \ldots, m$, and denoting the resulting right-hand side $\sum_{t \in T_i} \phi_{it}^{R(\Omega)}(x)$ in this sum as $\phi_i^{R(\Omega)}(x)$, we have,

$$\phi_i(x) \geq \phi_i^{R(\Omega)}(x) \equiv \sum_{t \in T_i} \phi_{it}^{R(\Omega)}(x) \qquad \forall\ x \in \Omega, \qquad i = 0, 1, \ldots, m. \tag{7.29}$$

Accordingly, we construct the corresponding approximating polynomial programming relaxation **PPR(Ω)** as follows. Note from (7.26) - (7.29) that PPR has integral exponents.

$$\textbf{PPR}(\Omega)\textbf{:} \quad \text{Minimize } \left\{\phi_0^{R(\Omega)}(x) : \ \phi_i^{R(\Omega)}(x) \leq \beta_i \quad \forall\ i = 1, \ldots, m,\ x \in \Omega\right\}. \tag{7.30}$$

The fact that PPR(Ω) provides a lower bounding relaxation for Problem PP(Ω) follows directly from (7.29). Moreover, from (7.26) and (7.28), we note that the degree δ of this problem, defined as the highest degree of any polynomial term in this problem, is given by

$\delta \equiv$ degree of the polynomial program that would be obtained by *rounding up* all fractional exponents in (7.21d). (7.31)

We now present the proposed RLT scheme for generating linear programming relaxations for Problem PP(Ω). As in Section 7.1, this scheme operates in the following two phases.

Reformulation Phase

Step I: Given PP(Ω), generate the relaxed polynomial program PPR(Ω) given by (7.30) as described above. Let δ denote the degree of PPR(Ω) (see Equation (7.31)).

Step II: Let $\overline{N} = \{N, \ldots, N\}$ denote δ replicates of N. Compose all possible distinct constraints of the type (7.2) as in Section 7.1, obtained by taking products of the *bound-factors* $(x_j - \ell_j) \geq 0$ and $(u_j - x_j) \geq 0$ δ at a time, including possible repetitions. Augment Problem PPR(Ω) by adding these constraints (7.2). (In addition, *optionally*, other implied polynomial constraints of degree less than or equal to δ can be generated by taking suitable inter-products of the *constraint-factors* $\beta_i - \phi_i^R(x) \geq 0$, $i = 1, \ldots, m$, and the aforementioned bound-factors, and these can be added to PPR(Ω) in order to further tighten the relaxation obtained via this overall RLT process.)

Linearization Phase

Linearize the resulting polynomial program obtained at the end of the Reformulation Phase by substituting (7.3). Denote the resulting linear program thus produced by LP(Ω).

7.2.1. A Branch-and-Bound Algorithm

With the exception of the partitioning rule, the proposed branch-and-bound algorithm proceeds precisely as described in Section 7.1.1, using the linear programs LP(Ω^q) obtained by the above RLT process to generate a lower bound for any node subproblem defined on the hyperrectangle $\Omega^q \subseteq \Omega$. Here, in order to actively seek good quality feasible solutions, because of the two layers of approximation employed in the RLT scheme, it might be beneficial to apply some heuristic perturbations or Newton-Raphson iterations to the linear programming solutions obtained in order to possibly convert them into feasible solutions for the original polynomial program $\mathrm{PP}(\Omega)$.

The critical element of difference that guarantees convergence to a global minimum for this rational exponent case is the choice of a suitable partitioning strategy. Three such branching rules that can be used at Step 1 of the scheme of Section 7.1.1 in order to assure convergence to an optimum are stated below. The first of these (Rule (A)) is a simple, standard bisection rule. While this is sufficient to ensure convergence since it drives all the intervals to zero for the variables that are associated with the term that yields the greatest discrepancy in the employed approximation along any infinite branch of the branch-and-bound tree, it is not too cognizant of the nature and solution of the employed relaxations. Rule (B) below accordingly suggests branching at the value $\bar{x}_{itj}$ given by (7.27) whenever the approximating problem employs the tangential approximation at this point for the term under consideration, motivated in part by the fact that the tangential approximation is exact at this point. Likewise, Rule (C) is further oriented toward computational effectiveness by combining Rule (B) with a partitioning at the linear programming relaxation value whenever admissible, since as motivated by Lemma 7.3, the RLT approximation involving any variable becomes exact (in the sense stated in the Lemma) whenever this variable coincides with one of its bounds. Hence, by letting the current linear programming relaxation value become an upper interval bound on one partitioned subnode and a lower interval bound on the other, we encourage the tightening of the resulting relaxations. Further motivation is provided with the detailed statement given below.

BRANCHING RULES

Consider any node subproblem identified by the hyperrectangle $\Omega' \subseteq \Omega$, and let $(\hat{x}, \hat{X})$ represent the solution obtained to its associated linear programming relaxation LP(Ω'). Determine the term (r, τ) in the polynomial program for which the discrepancy between its value computed at $\hat{x}$ and the value of its lower bounding function $\phi_{r\tau}^{R(\Omega')}(x)$ given by (7.28) and linearized under (7.3) in LP(Ω'), computed at $(\hat{x}, \hat{X})$, is a maximum. Letting $\phi_{r\tau}^{L(\Omega')}(\hat{x}, \hat{X})$ denote the latter linearized value, we have that

$$\alpha_{r\tau} \prod_{j \in J_{r\tau}} \hat{x}_j^{\gamma_{r\tau j}} - \phi_{r\tau}^{L(\Omega')}(\hat{x}, \hat{X}) = \underset{i=0,1,\dots,m,\, t \in T_i}{\text{maximum}} \{\alpha_{it} \prod_{j \in J_{it}} \hat{x}_j^{\gamma_{itj}} - \phi_{it}^{L(\Omega')}(\hat{x}, \hat{X})\}. \quad (7.32)$$

The selection of the branching variable x_p and the partitioning of Ω' is then done using one of the following rules, where $\Omega' \equiv \{x \colon \ell'_j \le x_j \le u'_j \ \forall\ j \in N\}$.

Rule (A) Let $p = \arg\max \{u'_j - \ell'_j \colon j \in J_{r\tau}\}$, and partition Ω' by bisecting the interval $[\ell'_p, u'_p]$ into the subintervals $[\ell'_p, \frac{\ell'_p + u'_p}{2}]$ and $[\frac{\ell'_p + u'_p}{2}, u'_p]$.

Rule (B) Let $p = \arg\max \{u'_j - \ell'_j \colon j \in J_{r\tau}\}$. Motivated by the choice of the point $\bar{x}_{r\tau p}$, partition Ω' by subdividing the interval $[\ell'_p, u'_p]$ into $[\ell'_p, \bar{x}_{r\tau p}]$ and $[\bar{x}_{r\tau p}, u'_p]$ if $f_{r\tau p} > 0$, where $\bar{x}_{r\tau p}$ is given by (7.27) for the current bounds on x_p, and by bisecting $[\ell'_p, u'_p]$ if $f_{r\tau p} = 0$. (Alternatively, motivated by (7.28), this rule can be modified so that it is applied only when $\alpha_{r\tau} < 0$, with the bisection rule being used if $\alpha_{r\tau} > 0$.)

Rule (C) For each $j \in J_{r\tau}$ compute

$$\theta_j = \min\{\hat{x}_j - \ell'_j, u'_j - \hat{x}_j\} \text{ and let } \tilde{x}_j = \hat{x}_j, \text{ if } \alpha_{r\tau} > 0 \text{ or if } \alpha_{r\tau} < 0$$

$$\text{and } f_{r\tau j} = 0, \tag{7.33a}$$

and compute

$$\theta_j = \max\{|\hat{x}_j - \bar{x}_{r\tau j}|, \min\{\hat{x}_j - \ell'_j, u'_j - \hat{x}_j\}\} \text{ and let } \tilde{x}_j = \bar{x}_{r\tau j},$$

$$\text{otherwise.} \tag{7.33b}$$

Note that in the case of (7.33a), we have used the left inequality in (7.26) for the approximation (7.28), and this inequality holds as an equality in case $\hat{x}_j$ equals ℓ'_j or u'_j, and moreover, by Lemma 7.3, $\hat{x}_j \hat{X}_J \equiv \hat{X}_{J \cup j}$ $\forall$ J in this case. On the other hand, for the case of (7.33b) where $\alpha_{r\tau} < 0$ and $f_{r\tau j} > 0$, we have used the right inequality in (7.26) for the approximation (7.28), and we would like $\hat{x}_j$ to coincide with $\bar{x}_{r\tau j}$ to make this hold as an equality, but as in Lemma 7.3, we would also like $\hat{x}_j$ to be close to ℓ'_j or u'_j. Hence, with this motivation of θ_j, in conjunction with our desire to reduce the bounding interval lengths, we select $p = \arg\max\ \{(u'_j - \ell'_j)\theta_j : j \in J_{r\tau}\}$, and we partition Ω' by subdividing the interval $[\ell'_p, u'_p]$ into $[\ell'_p, \tilde{x}_p]$ and $[\tilde{x}_p, u'_p]$.

Adopting any of the foregoing partitioning rules (A), (B), or (C), it can be shown similar to the proof of Theorem 7.1 that either PP(Ω) is solved finitely by this algorithm, or else, an infinite sequence of stages is generated such that along any infinite branch of the branch-and-bound tree, any accumulation point of the x-variable part of the linear programming relaxation solutions generated for the node subproblems solves PP(Ω).

As before, the key of the proof is to exhibit that along any infinite branch of the branch-and-bound tree, having identified a convergent subsequence of relaxation solutions corresponding to nodes that yield the least lower bounds over a sequence of stages, the sequence of approximation errors for the most discrepant term approaches zero. Details of this analysis require technical arguments, and can be found in Sherali (1996). We now conclude our discussion in this chapter with an illustrative example.

7.2.2. An Illustrative Example

Consider the following example adapted from Al-Khayyal and Falk (1983).

$$\text{Minimize } \{-x_1 + x_1 x_2^{0.5} - x_2 : \ -6x_1 + 8x_2 \le 3,\ 3x_1 - x_2 \le 3,$$
$$(0,0) \le (x_1, x_2) \le (1.5, 1.5)\}.$$

Denoting any bounding intervals on the variables x_1 and x_2 by $[\ell_i, u_i]$, for $i = 1, 2$, respectively, we have from (7.24) and (7.26) that

$$x_2^{0.5} \ge \ell_2^{0.5} + (x_2 - \ell_2)(u_2^{0.5} - \ell_2^{0.5}) / (u_2 - \ell_2).$$

Hence, defining Ω as in (1c), we have from (7.28) and (7.29) that the approximating polynomial program PPR(Ω) defined in (7.30) is given as follows.

PPR(Ω): Minimize $\{(\lambda - 1)x_1 + \mu x_1 x_2 - x_2 \colon -6x_1 + 8x_2 \le 3,$

$$3x_1 - x_2 \le 3,\ x \in \Omega\},$$

where $\lambda \equiv [u_2 \ell_2^{0.5} - \ell_2 u_2^{0.5}] / (u_2 - \ell_2)$ and $\mu = [u_2^{0.5} - \ell_2^{0.5}]/(u_2 - \ell_2)$. The degree of this polynomial program is $\delta = 2$. Hence, at Step II of the Reformulation Phase of RLT, we generate all pairwise products (21 in number) of the constraint and bound-factors $(3 + 6x_1 - 8x_2) \ge 0$, $(3 - 3x_1 + x_2) \ge 0$, $(x_1 - \ell_1) \ge 0$, $(u_1 - x_1) \ge 0$, $(x_2 - \ell_2) \ge 0$, and $(u_2 - x_2) \ge 0$, including self-products. These product constraints are linearized by substituting

$X_{11} = x_1^2$, $X_{22} = x_2^2$, and $X_{12} = x_1 x_2$.

For example, the product constraint $(3 + 6x_1 - 8x_2)\ (3 - 3x_1 + x_2) \ge 0$ yields

$9 + 9x_1 - 21x_2 - 18X_{11} - 8X_{22} + 30X_{12} \ge 0.$

The RLT lower bounding linear program LP(Ω) is then to minimize $\{(\lambda - 1)x_1 + \mu X_{12} - x_2\}$ subject to the 21 RLT linearized product constraints, where the original as well as the bounding restrictions on all the variables are easily verified to be implied by these constraints.

Using Branching Rule C, the problem is solved to optimality after enumerating 7 nodes as shown in Figure 7.2. Note that in this example, the most (actually, only) discrepant term via (7.32) is the nonlinear term that appears in the objective function, and the interval partitioning is performed at the linear programming solution value as determined via (7.33a).

As a point of interest, using either the bisection Rule (A) or the Branching Rule (B), it may be verified that the branch-and-bound algorithm enumerates 9 nodes in order to solve the problem.

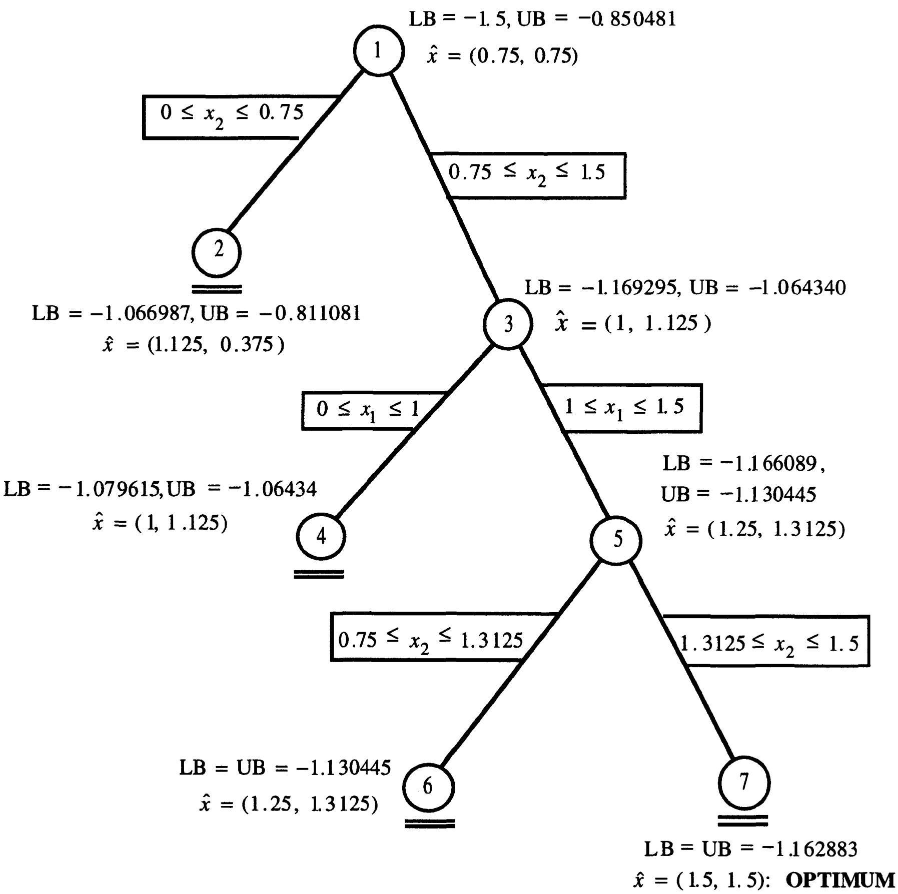


Figure 7.2. Branch-and-bound tree for Branching Rule (C).

8

REFORMULATION-CONVEXIFICATION TECHNIQUE FOR QUADRATIC PROGRAMS AND SOME CONVEX ENVELOPE CHARACTERIZATIONS

In the previous chapter, we have presented general concepts and theory for using RLT to solve polynomial problems. In the present chapter, we discuss certain specialized RLT results and implementation strategies, focusing on the global optimization of nonconvex quadratic programming problems. These problems are of the form

QP: Minimize $cx + x^t Qx$

subject to $Ax \le b$

$$0 \le \ell_k \le x_k \le u_k < +\infty \quad k = 1, \ldots, n$$

where $c \in R^n$, A is an $m \times n$ real matrix, $b \in R^m$, Q is an $n \times n$ indefinite matrix, assumed to be symmetric for convenience, and where the decision variables are $x \in R^n$. Our proposed RLT methodology is equally applicable to the case where Q is a negative semidefinite matrix; however, for this pure concave case, we do not take advantage of the extremality property of the global optimum (see Horst and Tuy, 1993), and treat it just as another nonconvex instance of the problem. Notice that we have assumed the existence of finite lower and upper bounds on all the variables. These might be prespecified bounds, or they might be derived from the other constraints of QP and imposed herein for certain algorithmic purposes. As we proceed, we shall point out modifications to be made in our algorithmic constructs whenever such bounds are known to be implied by the other constraints defining the problem, but are not explicitly specified.

Pardalos and Vavasis (1991) have shown that quadratic programs are NP-hard, even when Q has just one negative eigenvalue. These problems, which arise variously in applications such as in modeling economies of scale in a cost structure, in location-allocation problems, VLSI design problems, some production planning and risk management problems, and in various other mathematical models such as the maximum clique problem and the jointly constrained bilinear programming problem, have therefore been both interesting and challenging problems to solve. Pardalos and Rosen (1987) and Pardalos (1991) present useful surveys, and Vavasis (1992) and Floudas and Visweswaran (1990, 1993) present interesting solution approaches. Our approach here mainly follows Sherali and Tuncbilek (1995) and Sherali and Alameddine (1992).

Accordingly, in this chapter, we investigate various specialized RLT designs for generating both linear (RLT-LP), and convex, nonlinear (RLT-NLP) lower bounding problems for QP that can be suitably embedded within a provably convergent branch-and-bound procedure. In general, we call this process of generating relaxations a *Reformulation-Convexification* approach, since our relaxations are, in general, convex programs that are not necessarily linear. The first scheme we propose involves the generation of quadratic constraints through a construction of pairwise constraint products.

These constraints are subsequently linearized to yield a lower bounding linear program. Section 8.1 presents these constructs, and Section 8.2 provides an illustrative example. Next, in Section 8.3, we develop various general properties and results related to the first level RLT procedure, and in Section 8.4 we demonstrate its tendency to produce convex hull representations in special cases. (Some general tricks to construct such representations are also revealed in this section.) In particular, one of the properties we establish is that this first level relaxation is invariant under affine transformations applied to the original problem constraints. However, if eigen-transformation is used as a particular linear transformation, this enables us in Section 8.5 to introduce additional nonlinear convex constraints that can further strengthen the linear programming relaxation produced by RLT. Such nonlinear constraints can be suitably handled within a Lagrangian dual procedure, without hampering the efficiency of the solution procedure as compared with that for solving the linear programming bounding problem. We also present a rule for reducing the number of new second-order constraints generated for the bounding problem, without compromising much on the quality of the resulting lower bound. This is discussed in Section 8.6, and some comparative results on the strengths of the various proposed relaxations are presented in Section 8.7.

Following this development, in Section 8.8, we discuss implementation issues and strategies for a hybrid best-first and depth-first branch-and-bound algorithm for solving QP. Branching is performed based on the partitioning of the box constraints defined by the bounds on each variable. Lower and upper bounds on QP are then derived using the designed RLT relaxations. Several other algorithmic strategies are gainfully employed in solving the problem. In particular, the Lagrangian dual formulation is enhanced using a layering strategy, and various simple strategies are devised for range-restricting variables as well as for tightening the lower bounds in order to enhance the fathoming efficiency of the algorithm. Guidelines for other implementation details, along with some computational experience on test problems from the literature are finally provided in Section 8.9.

8.1. Reformulation by Generating Quadratic Constraints (RLT-LP)

In this section, we present our fundamental RLT construct that involves generating quadratic implied constraints, and subsequently, linearizing them to obtain a level-one RLT relaxation. To generate these quadratic constraints, we consider pairwise products of *bound-factors* and *constraint-factors* as introduced in Chapters 2, 3, and 7, defined by individual variable bounds and structural constraints that define QP. For brevity of presentation, let us combine bound and constraint-factors in a single set as follows:

$$\begin{bmatrix} (b_i - a_i x) \geq 0, & i = 1,\ldots,m \\ (u_k - x_k) \geq 0, & k = 1,\ldots,n \\ (x_k - \ell_k) \geq 0, & k = 1,\ldots,n \end{bmatrix} \equiv \begin{bmatrix} (g_i - G_i x) \geq 0 \\ i = 1,\ldots,m+2n \end{bmatrix} \tag{8.1}$$

where $a_i x \leq b_i$ is the i^{th} (structural) constraint from $Ax \leq b$, for $i = 1,\ldots,m$. In the **reformulation step**, we take all possible pairwise products of the factors in (8.1), including self-products, to generate the following nonlinear implied constraints that are included in the original problem QP:

$$(g_i - G_i x)(g_j - G_j x) \geq 0 \qquad \forall\ 1 \leq i \leq j \leq m + 2n. \tag{8.2}$$

We then **linearize** the resulting augmented problem by substituting

$$w_{k\ell} = x_k x_\ell \quad \forall\ 1 \leq k \leq \ell \leq n. \tag{8.3}$$

This substitution associates a separate new variable with each distinct nonlinear term in the problem. We often refer to these variables as *RLT variables.* The resulting problem is called the *first-level* or *first-order RLT*, since it employs first-order (linear) factors to generate new constraints. Denoting $[(g_i - G_i x)(g_j - G_j x)]_L \geq 0$, $\forall\ 1 \leq i \leq j \leq m + 2n$ as the resulting linearized constraints, we obtain the following lower bounding first-level RLT linear program, where $q_{k\ell}(= q_{\ell k})$ is the $(k, \ell)^{th}$ element of the symmetric matrix Q.

RLT-LP: Minimize $\displaystyle\sum_{k=1}^{n} c_k x_k + \sum_{k=1}^{n} q_{kk} w_{kk} + 2\sum_{k=1}^{n-1} \sum_{\ell=k+1}^{n} q_{k\ell} w_{k\ell}$ (8.4a)

subject to $[(g_i - G_i x)(g_j - G_j x)]_L \geq 0$

$\forall\ 1 \leq i \leq j \leq m + 2n.$ (8.4b)

Notice that the original constraints of QP are not included in RLT-LP, since, as we shall show in Proposition 8.1 later, these constraints are implied by the RLT constraints

(8.4b), even if the feasible region is not assumed to be bounded, provided that at least one variable has a bounded range over the feasible region.

For any feasible solution to problem QP, there exists a feasible solution to RLT-LP having the same objective function value through the definitions (8.3). However, the converse is not necessarily true. Therefore, RLT-LP is a relaxation of QP that yields a lower bound on the global minimum of QP. Moreover, if $(\bar{x}, \bar{w})$ solves RLT-LP, then since $\bar{x}$ is feasible to QP, it provides an upper bound on this problem. In particular, if this solution also satisfies the definitions (8.3) for all the nonlinear terms appearing in QP, then $\bar{x}$ solves QP.

Remark 8.1 (Equality Constraints). If there is some equality constraint $G_e x = g_e$ in QP, then we only need to consider the product of the constraint-factor $(g_e - G_e x) = 0$ with each variable x_k, $k = 1, \dots, n$, as in Section 2.5 for the discrete case, since all the other RLT constraints generated via this constraint can be obtained by suitably surrogating the constraints $[x_k(g_e - G_e x)]_L = 0,\ k = 1, \dots, n.$

8.2. An Illustrative Example: Higher Order Constraints

To provide some insights into the RLT process and its effects, consider the following illustrative concave quadratic program:

Example 8.1.

Minimize $\{z = -(x_1 - 12)^2 - x_2^2 \colon\ -6x_1 + 8x_2 \leq 48,\ 3x_1 + 8x_2 \leq 120,$

$0 \leq x_1 \leq 24,\ x_2 \geq 0\}.$

The optimal solution to this problem is $(x_1^*, x_2^*) = (24, 6)$, or alternatively, (0, 6), and the optimal objective function value is $z^* = -180$. The feasible region happens to be bounded, and given by the convex hull of the vertices (0, 0), (0, 6), (8, 12), (24, 6), and (24, 0). Before generating the first-level RLT for this problem, let us identify the bound-factors as $(24 - x_1) \geq 0$, $x_1 \geq 0$, $x_2 \geq 0$, and denote the constraint-factors as $s_1 = (48 + 6x_1 - 8x_2) \geq 0$, and $s_2 = (120 - 3x_1 - 8x_2) \geq 0$. The first-level RLT problem is then generated by constructing the 15 pairwise products (including self-products) of these factors, and then linearizing the resulting quadratically constrained QP via the variable redefinitions $w_{11} = x_1^2$, $w_{12} = x_1x_2$, and $w_{22} = x_2^2$.

The solution to this first-level RLT problem is given by $(\bar{x}_1, \bar{x}_2, \bar{w}_{11}, \bar{w}_{12}, \bar{w}_{22}) = (8, 6, 192, 48, 72)$ and has an objective function value of –216. Hence, –216 provides a lower bound on QP. Notice that $\bar{x}_1\bar{x}_2 = \bar{w}_{12}$; however, the same is not true for $\bar{w}_{11} = 192 \neq \bar{x}_1^2 = 64$, and $\bar{w}_{22} = 72 \neq \bar{x}_2^2 = 36$. Notice also that $\bar{x} = (8, 6)$ is feasible to QP and has an objective function value of –52. This provides an upper bound on QP.

Now, let us partition the problem via the dichotomy $x_1 \leq 8$ or $x_1 \geq 8$. When we solve a first-level RLT problem separately for each partitioned subproblem, the optimal value turns out to be –180 in both cases. That is, the original problem is solved after a single branching on x_1.

To show what RLT is trying to achieve, let us introduce cubic RLT constraints. In the following problem, we consider certain selected second-order RLT constraints along with some cubic bound/constraint-factor products to yield additional RLT constraints. (This selection is done for the purpose of illustration, and is motivated by the magnitude of the optimal dual variable values of the first-level RLT problem.) The resulting linear program given below is a (partial) second-level RLT problem, where the additional RLT variables represent the cubic terms $w_{111} = x_1^3$, $w_{112} = x_1^2 x_2$, and $w_{122} = x_1 x_2^2$.

Minimize $-w_{11} - w_{22} + 24x_1 - 144$

subject to

$$[(24 - x_1)s_1]_L = 1152 + 96x_1 - 192x_2 - 6w_{11} + 8w_{12} \geq 0$$

$$[x_1 s_2]_L = 48x_1 + 6w_{11} - 8w_{12} \geq 0$$

$$[(24 - x_1)x_1]_L = 24x_1 - w_{11} \geq 0$$

$$[(24 - x_1)x_1 s_1]_L = 1152x_1 + 96w_{11} - 192w_{12} - 6w_{111} + 8w_{112} \geq 0$$

$$[(24 - x_1)x_2 s_1]_L = 1152x_2 + 96w_{12} - 192w_{22} - 6w_{112} + 8w_{122} \geq 0$$

$$[(24 - x_1)x_1 s_2]_L = 2880x_1 - 192w_{11} - 192w_{12} + 3w_{111} + 8w_{112} \geq 0$$

$$[x_1 x_2 s_2]_L = 120w_{12} - 3w_{112} - 8w_{122} \geq 0.$$

The solution to this revised problem is given by $(\hat{x}_1, \hat{x}_2) = (0, 6)$, $\hat{w}_{22} = 36$, $\hat{w}_{11} = \hat{w}_{12} = \hat{w}_{111} = \hat{w}_{112} = \hat{w}_{122} = 0$, having the objective function value –180. Hence, the above linear bounding problem composed of selected second and third-order

RLT constraints solves the original problem. In fact, we can demonstrate in this instance that these constraints effectively describe the convex envelope of the objective function over the problem constraints for the original concave quadratic program.

To see this, let us define a particular surrogate of the above third-order RLT constraints as follows.

$$surr[cubic]_L \equiv 1/512[(24 - x_1)x_1s_1]_L + 1/192[(24 - x_1)x_2s_1]_L +$$

$$1/256[(24 - x_1)x_1s_2]_L + 1/192[x_1x_2s_2]_L \geq 0.$$

When we further surrogate this constraint with the objective function, and the second-order RLT constraints using two different sets of weights in turn, we obtain the following two constraints that yield the desired convex envelope representation.

$$\{z + w_{11} + w_{22} - 24x_1 + 144 = 0\} + 7/16\{[(24 - x_1)x_1]_L \geq 0\} +$$

$$\{surr[cubic]_L \geq 0\} \Rightarrow z \geq -6x_2 - 144$$

$$\{z + w_{11} + w_{22} - 24x_1 + 144 = 0\} + 7/144\{[(24 - x_1)s_1]_L \geq 0\} +$$

$$7/144\{[x_1s_2]_L \geq 0\} + \{surr[cubic]_L \geq 0\} \Rightarrow z \geq (10/3)x_2 - 200.$$

Hence, we were able to solve the given QP as the above single linear program. In fact, as this example exhibits, the RLT scheme attempts to approximate the convex envelope of the objective function over the feasible region in deriving a lower bounding linear program. If this approximation is composed properly, a tight representation of the problem can be derived.

8.3. Fundamental Insights and Results for the Level-One RLT Relaxation

In this section, we present some general results that provide insights into the level-one (first level) RLT relaxation. Some implications of these results will find direct use in designing a suitable branch-and-bound algorithm. In concert with definition (8.1), we refer to the constraint set of QP, including the simple bound restrictions on the variables, as $Gx \leq g$.

The first result below shows that the original constraints $Gx \leq g$ need not be included in the first-level RLT problem, even under a relaxed boundedness assumption on QP.

Proposition 8.1. *Suppose that the feasible region of QP is not necessarily bounded (possibly including unrestricted variables) but that for some variable $k \in \{1, \ldots, n\}$, we have,*

$$U_k \equiv max\{x_k: \ G_i x \leq g_i, \ i = 1, \ldots, m + 2n\} < \infty \tag{8.5a}$$

$$L_k \equiv min\{x_k: \ -G_i x \geq -g_i, \ i = 1, \ldots, m + 2n\} > -\infty \tag{8.5b}$$

where $U_k > L_k$. Then, the original constraints $Gx \leq g$ are implied by the RLT constraints $[(g_i - G_i x)(g_j - G_j x)]_L \geq 0 \ \forall \ 1 \leq i \leq j \leq m + 2n$.

Proof. Let $\mu^u \geq 0$ and $\mu^\ell \geq 0$ be the optimal dual multiplier vectors associated with the constraints in (8.5a) and (8.5b), respectively. Then, we have $G^t \mu^u = -G^t \mu^\ell = e_k$, where $e_k \in R^n$ is the unit vector with entry 1 at the k^{th} position, and also, $g^t \mu^u = U_k$, and $-g^t \mu^\ell = L_k$.

Now, for any $j \in \{1, \dots, m + 2n\}$, consider the surrogate of all RLT constraints involving the constraint-factor $(g_j - G_j x \geq 0)$, obtained by using the weights μ^u.

$$0 \leq \sum_{i=1}^{m+2n} \mu_i^u [(g_i - G_i x)(g_j - G_j x)]_L = \sum_{i=1}^{m+2n} \mu_i^u (g_i g_j - g_i G_j x - g_j G_i x +$$

$$[(G_i x)(G_j x)]_L) = -g_j \sum_{i=1}^{m+2n} \mu_i^u G_i x + (g_j - G_j x) \sum_{i=1}^{m+2n} \mu_i^u g_i + \sum_{i=1}^{m+2n} \mu_i^u G_i [xx^t]_L G_j^t$$

$$= -g_j x_k + (g_j - G_j x) U_k + e_k^t [xx^t]_L G_j^t.$$

After rearranging the terms in the final expression, and noting that $e_k^t [xx^t]_L G_j^t = [x_k (G_j x)]_L$, we obtain,

$$U_k (g_j - G_j x) - [x_k (g_j - G_j x)]_L = [(U_k - x_k)(g_j - G_j x)]_L \geq 0. \tag{8.6a}$$

Replacing μ^u by μ^ℓ, and then following the same steps as above, we obtain,

$$-L_k (g_j - G_j x) + [x_k (g_j - G_j x)]_L = [(x_k - L_k)(g_j - G_j x)]_L \geq 0. \tag{8.6b}$$

The surrogate of (8.6a) and (8.6b) gives $(U_k - L_k)(g_j - G_j x) \geq 0$, which implies that $G_j x \leq g_j$, since $U_k > L_k$. Since this is true for any $j \in \{1, \dots, m + 2n\}$, including the bounding constraints, the proof is complete. □

Consider the constraints $Gx \leq g$, and suppose that some of these constraints are implied by the other constraints. In particular, let us assume that the constraints $G_i x \leq g_i$, $i \in \{1, \dots, m'\}$ imply the other constraints within $Gx \leq g$, and represent, say a minimal set of non-implied constraints. The second proposition below exhibits that we do not need to use any implied constraint in generating RLT constraints, because such constraints would be implied by the RLT constraints that are generated via the products of the non-implied constraints. Hence, any constraints within $Gx \leq g$ that are known to be implied can be discarded without loss of any tightness in the resulting RLT problem, and moreover, it is futile to use any implied constraint, such as implied bounds on variables, to generate additional RLT constraints. For convenience, let us refer to the RLT constraints generated via the non-implied set of original constraints as

$$\Gamma \equiv \{[(g_i - G_i x)(g_j - G_j x)]_L \geq 0 \quad \forall\ 1 \leq i \leq j \leq m'\}. \tag{8.7}$$

Proposition 8.2. *Suppose that the assumption of Proposition 8.1 holds true, and let $\alpha x \leq \beta$ be implied by the constraints $G_i x \leq g_i$, $i = 1, \ldots, m'$. Then, the RLT constraints $[(\beta - \alpha x)(g_i - G_i x)]_L \geq 0$, $i = 1, \ldots, m'$, and $[(\beta - \alpha x)^2]_L \geq 0$ are also implied by the RLT constraints $[(g_i - G_i x)(g_j - G_j x)]_L \geq 0$, $1 \leq i \leq j \leq m'$. In particular, the constraints of the first-level RLT problem are all implied by the RLT constraints contained within Γ.*

Proof. Since $\beta \geq \text{maximum}\{\alpha x\colon\ G_i x \leq g_i,\ i = 1, \ldots, m'\}$, there exist dual multipliers $\mu_i \geq 0$, $i = 1, \ldots, m'$ such that

$$\sum_{i=1}^{m'} \mu_i G_i = \alpha \text{ and } \sum_{i=1}^{m'} \mu_i g_i \leq \beta. \tag{8.8}$$

Now, for any $j \in \{1, \ldots, m'\}$, consider the surrogate of the following RLT constraints from Γ involving the constraint-factor $(g_j - G_j x) \geq 0$, obtained by using the weights μ.

$$0 \leq \sum_{i=1}^{m'} \mu_i [(g_i - G_i x)(g_j - G_j x)]_L = \sum_{i=1}^{m'} \mu_i [(g_i g_j - g_i G_j x - g_j G_i x +$$

$$[(G_i x)(G_j x)]_L) = -g_j \sum_{i=1}^{m'} \mu_i G_i x + (g_j - G_j x) \sum_{i=1}^{m'} \mu_i g_i + \sum_{i=1}^{m'} \mu_i G_i [xx^t]_L G_j^t.$$

Since by Proposition 8.1, $(g_j - G_j x) \geq 0$ is implied by Γ, we get upon using (8.8) that $0 \leq -g_j \alpha x + \beta(g_j - G_j x) + \alpha[xx^t]_L G_j^t = [(\beta - \alpha x)(g_j - G_j x)]_L$.

Hence, $[(\beta - \alpha x)(g_j - G_j x)]_L \geq 0$ is implied by Γ for all $j \in 1, \ldots, m'$. Furthermore, by following the same algebraic steps as above, and using the foregoing assertion, we get $0 \leq \sum_{j=1}^{m'} \mu_j [(g_j - G_j x)(\beta - \alpha x)]_L \leq [(\beta - \alpha x)(\beta - \alpha x)]_L$. Hence, the self-product constraint $[(\beta - \alpha x)^2]_L \geq 0$ is also implied by Γ. The final assertion of the proposition now follows by inductively taking the implied constraints of $Gx \leq g$ one at a time, and establishing as above that the RLT constraints generated via this constraint, including the self-product constraint, are implied by the RLT constraints generated via the other remaining constraints. Discarding such implied constraints sequentially, we deduce that the constraint set Γ implies the other RLT constraints, and this completes the proof. $\square$

Knowing that the original constraints are implied by the RLT constraints, if an original constraint from $Gx \leq g$ is binding at some feasible solution $(\bar{x}, \bar{w})$ to RLT-LP, then one would expect that certain RLT constraints would also be binding at $(\bar{x}, \bar{w})$. The next result shows that the RLT constraints generated using a constraint-factor that turns out to be binding, are themselves binding. However, for this result, we need the original assumption that the feasible region of QP is bounded.

Proposition 8.3. *Assume that the feasible region of QP is bounded, and suppose that at a given point $(\bar{x}, \bar{w})$, we have $G_i\bar{x} = g_i$, for some $i \in \{1, \dots, m + 2n\}$. Then, the linearized product of this constraint factor with any original variable x_k, $k \in \{1, \dots, n\}$ at $(\bar{x}, \bar{w})$ is zero. That is, denoting $[\cdot]_L$ evaluated at $(\bar{x}, \bar{w})$ by $[\cdot]_L|_{(\bar{x},\bar{w})}$, we have,*

$$[x_k(g_i - G_i x)]_L|_{(\bar{x},\bar{w})} \equiv g_i\bar{x}_k - \sum_{\ell=1}^{k} G_{i\ell}\bar{w}_{\ell k} - \sum_{\ell=k+1}^{n} G_{i\ell}\bar{w}_{k\ell} = 0,$$

$$\forall\ k = 1, \dots, n. \tag{8.9}$$

In particular, the RLT constraint generated by multiplying this constraint with any other constraint is also binding.

Proof. Under the boundedness assumption of the feasible region, consider the constraints (8.6a) and (8.6b), which are implied by or which already exist in Γ by Proposition 8.2, in a combined form as follows:

$$L_k(g_i - G_i x) \leq [x_k(g_i - G_i x)]_L \leq U_k(g_i - G_i x) \quad \forall\ k = 1, \dots, n. \tag{8.10}$$

Evaluating (8.10) at $(\bar{x}, \bar{w})$, since $g_i - G_i\bar{x} = 0$, we obtain (8.9) holding true. For any $j \in \{1, \dots, m + 2n\}$, when we evaluate the following RLT constraint at $(\bar{x}, \bar{w})$,

$$[(g_i - G_i x)(g_j - G_j x)]_L = g_j(g_i - G_i x) - \sum_{k=1}^{n} G_{jk}[x_k(g_i - G_i x)]_L \geq 0, \qquad (8.11)$$

we get by (8.9) and $(g_i - G_i\bar{x}) = 0$, that (8.11) is satisfied as an equality. This completes the proof. □

Next, we address the question whether an application of RLT after using some affine transformation can possibly produce a different relaxation. This might be of interest, in particular, if one wished to investigate the effect of applying RLT to different nonbasic space representations of the linear constraints defining QP, or the effect of employing an eigen-transformation on QP before applying RLT. More specifically, given the quadratic programming problem

QP: $\text{Minimize}\{cx + x^t Qx\colon\ Gx \leq g\} < \infty,$

define a nonsingular affine transformation $s = Bx + p$ to represent QP in s-space as follows, using the substitution $x \equiv B^{-1}s - B^{-1}p$, and where $B^{-t} \equiv (B^{-1})^t$.

QP′: $-cB^{-1}p + p^t B^{-t} Q B^{-1} p + \text{minimize } (cB^{-1} - 2p^t B^{-t} Q B^{-1})s +$

$$s^t B^{-t} Q B^{-1} s$$

$$\text{subject to } \ GB^{-1}s \leq g + GB^{-1}p.$$

Let **RLT-LP** and **RLT-LP′** be the linear programs obtained by applying the first level RLT to QP and QP′, respectively, using all possible pairwise constraint-factor products. These problems can be stated as follows, where G_i is the i^{th} row of G, for $i \in \{1, \ldots, m + 2n\}$.

RLT-LP:

Minimize $cx + [x^tQx]_L$

subject to $g_ig_j - (g_iG_j + g_jG_i)x + [(G_ix)(G_jx)]_L \geq 0$

$\forall\ (i,j) \in M_R \equiv \{(i,j):\ 1 \leq i \leq j \leq m + 2n\}.$

RLT-LP′:

$-cB^{-1}p + p^tB^{-t}QB^{-1}p$

+ minimize $(cB^{-1} - 2p^tB^{-t}QB^{-1})s + [s^tB^{-t}QB^{-1}s]_L$

subject to $(g_i + G_iB^{-1}p)(g_j + G_jB^{-1}p) - [(g_i + G_iB^{-1}p)(G_jB^{-1}) +$

$(g_j + G_jB^{-1}p)(G_iB^{-1})]s + [(G_iB^{-1}s)(G_jB^{-1}s)]_L \geq 0$

$\forall\ (i,j) \in M_R.$

Also, in accordance with our foregoing discussion, let us define the RLT variables for each quadratic term as $w_{k\ell} \equiv [x_kx_\ell]_L$ and $y_{k\ell} \equiv [s_ks_\ell]_L$, $1 \leq k \leq \ell \leq n$. Henceforth, we will let $\nu[\cdot]$ denote the value at optimality of the corresponding problem $[\cdot]$. The next proposition shows that the level-one RLT is invariant under affine transformations.

Proposition 8.4. $\nu[\text{RLT-LP}] = \nu[\text{RLT-LP}']$. *In particular, if* (x^*, w^*) *solves RLT-LP, and if* u^* *is the corresponding optimal dual solution, then*

$$(s^*, [y^*]) \equiv (s^*, [ss^t]_L^*) = (Bx^* + p,\ B[xx^t]_L^*B^t + p(x^*)^tB^t + Bx^*p^t + pp^t) \tag{8.12}$$

solves RLT−LP′, *with the corresponding optimal dual solution being* u^*, *where* $[y^*]$ *denotes an* $n \times n$ *matrix representation of* y^*, *such that for* $1 \le k \le \ell \le n$, $[y^*]_{k\ell} = [y^*]_{\ell k} = y^*_{k\ell}$, *and where* $[\cdot]_{k\ell}$ *is the* $(k, \ell)^{th}$ *entry of the* $n \times n$ *symmetric matrix* $[\cdot]$.

Proof. From the KKT conditions of the linear program RLT-LP, we have

$$-c - \sum_{(i,j) \in M_R} u^*_{ij}(g_j G_i + g_i G_j) = 0 \tag{8.13a}$$

and

$$-2Q + \sum_{(i,j) \in M_R} u^*_{ij}(G_j^t G_i + G_i^t G_j) = 0. \tag{8.13b}$$

(Note that in the dual feasibility conditions (8.13b), there are $n(n-1)/2$ more equations than the number of w-variables, but since the left-hand side is symmetric, (8.13b) is valid, having $n(n-1)/2$ duplicated equations.) Dual feasibility of u^* for RLT−LP′ then follows directly from (8.13a) and (8.13b), since we have,

$$0 = (\mathbf{8.13a})B^{-1} + p^t B^{-t}(\mathbf{8.13b})B^{-1}$$

$$\equiv -cB^{-1} + 2p^t B^{-t} Q B^{-1} - \sum_{(i,j) \in M_R} u^*_{ij}[(g_i + G_i B^{-1} p)(G_j B^{-1}) +$$

$$(g_j + G_j B^{-1} p)(G_i B^{-1})] \tag{8.14a}$$

and

$$0 = B^{-t}(\mathbf{8.13a})B^{-1} \equiv -2B^{-t} Q B^{-1} +$$

$$\sum_{(i,j)\in M_R} u^*_{ij} B^{-t}(G^t_j G_i + G^t_i G_j)B^{-1}. \tag{8.14b}$$

For verifying the primal feasibility and complementary slackness conditions with respect to (8.12), define

$$s^* = Bx^* + p, \text{ so that } x^* = B^{-1}(s^* - p) \tag{8.15a}$$

and

$$[y^*] = B[w^*]B^t + p(x^*)^t B^t + Bx^* p^t + pp^t =$$

$$B[w^*]B^t - (pp^t - p(s^*)^t - s^* p^t) \tag{8.15b}$$

where $[w^*]$ is the matrix representation of w^*, defined similar to $[y^*]$. Then, consider any constraint of RLT−LP′ evaluated at (s^*, y^*). Recognizing that ss^t in this constraint is replaced by $[y]$, we have, using (8.15a) and (8.15b),

$$(g_i + G_i B^{-1}p)(g_j + G_j B^{-1}p) - [(g_i + G_i B^{-1}p)(G_j B^{-1}) +$$

$$(g_j + G_j B^{-1}p)(G_i B^{-1})]s^* + G_i B^{-1}[y^*]B^{-t}G^t_j$$

$$= g_i g_j - (g_j G_i + g_i G_j)B^{-1}(s^* - p) + G_i B^{-1}([y^*] + pp^t - p(s^*)^t - s^* p^t)B^{-t}G^t_j$$

$$= g_i g_j - (g_j G_i + g_i G_j)x^* + G_i[w^*]G^t_j \geq 0$$

since (x^*, w^*) is primal feasible for RLT-LP. Moreover, this also exhibits that the slack values of constraints in RLT-LP and RLT−LP′ are the same when they are evaluated at (x^*, w^*) and (s^*, y^*), respectively. Hence, the complementary slackness conditions for RLT−LP′ follow directly from the KKT conditions for RLT-LP. To complete the proof, using (8.13a) and (8.13b), we establish the equivalence of the optimal objective function values as follows.

$v[\text{RLT−LP}'] =$

$$-cB^{-1}p + p^tB^{-t}QB^{-1}p - \sum_{(i,j)\in M_R} u_{ij}^*[(g_i + G_iB^{-1}p)(g_j + G_jB^{-1}p)]$$

$$= -\sum_{(i,j)\in M_R} u_{ij}^* g_i g_j + \left[-c - \sum_{(i,j)\in M_R} u_{ij}^*(g_iG_j + g_jG_i)\right]B^{-1}p$$

$$+p^tB^{-t}\left[Q - \frac{1}{2}\sum_{(i,j)\in M_R} u_{ij}^*(G_i^tG_j + G_j^tG_i)\right]B^{-1}p$$

$$= -\sum_{(i,j)\in M_R} u_{ij}^* g_i g_j = v[\text{RLT−LP}]$$

This completes the proof. □

8.4. Construction of Convex Hulls and Convex Envelopes: General Results and Some Special Cases

In this section, we provide some motivation for the use of RLT, and its observed computational success, by demonstrating that a particular application of this technique to certain bilinear functions defined over triangular and quadrilateral D-polytopes in R^2 (see definition below), followed by a projection onto the original variable space, produces the exact convex hull representation for these cases. We also present some generalized results that can be useful in constructing convex hulls of nonconvex sets, or convex envelopes of nonconvex functions, using RLT concepts.

To begin with, let us consider some definitions.

Definition 8.1 [see Horst and Tuy, 1993]. The convex envelope of a function f taken over a nonempty convex subset S of its domain is that function f_S for which:

(a) f_S is a convex function defined over S,

(b) $f_S(x) \leq f(x)$ for all $x \in S$, and

(c) if f': $S \to R$ is a convex function that satisfies $f'(x) \leq f(x)$ for all $x \in S$, then $f'(x) \leq f_S(x) \quad \forall\ x \in S$.

Equivalently, the convex envelope of f over the convex set S is given by the pointwise supremum of all convex, or even simply affine, underestimating functions for f over the set S. Notice that the form of the convex envelope f_S as a function f depends upon the subset S of the domain of f over which the convex envelope is defined.

Although it is easily seen that, in general, the convex envelope of a sum of functions *is not* produced by the sum of their respective convex envelopes, this result is true if the various functions are separably defined. The following result states this formally.

Proposition 8.5. *Let* $S = \prod_{i=1}^{r} S_i$ *be the Cartesian product of* r *compact* n_i-*dimensional convex sets* S_i, $i = 1,\ldots,r$, *such that* $\sum_{i=1}^{r} n_i = n$. *Suppose that f:*

$S \to R$ can be decomposed into the form $f(x) = \sum_{i=1}^{r} f_i(x^i)$, where f_i: $S_i \to R$ is lower semicontinuous on S_i, $i = 1,\ldots,r$. Then, the convex envelope f_S of f over the set S is equal to the sum of the convex envelopes f_{S_i} of f_i over S_i, for $i = 1,\ldots,r$, i.e., $f_S(x) = \sum_{i=1}^{r} f_{S_i}(x^i)$. (Note that for convenience in notation, we will denote "$f_{iS_i}(x^i)$" by simply $f_{S_i}(x^i)$.)

Proof. Horst and Tuy (1993, Theorem IV.8) prove this result for S and S_i, $i = 1,\ldots,r$ being hyperrectangles; however, the proof holds true even when the sets are arbitrary compact, convex sets. □

Now, to provide a facility for constructing the convex envelope of a nonconvex function f over a given convex set S, let us make use of the following concept. Consider a lower semicontinuous function f: $S \to R$, where $S \subseteq R^n$ is a nonempty, compact, convex set. Recall that (see Bazaraa *et al.*, 1993 for related definitions) the epigraph of f over the set S, denoted $E_S(f)$, is defined by

$$E_S(f) = \{(x,z): \ x \in S, z \geq f(x)\}. \tag{8.16}$$

Let us denote the convex hull of this set by *conv* $E_S(f)$. Then, we have the following result.

Proposition 8.6. *Let f: $S \to R$ be a lower semicontinuous function, where $S \subseteq R^n$ is convex and compact, and let $E_S(f)$ be as defined by (8.16). Then*

$$conv\ E_S(f) = \{(x,z): \ x \in S, z \geq f'(x)\} \Leftrightarrow f'(x) = f_S(x) \quad \forall\ x \in S. \tag{8.17}$$

Proof. The proof follows by Horst and Tuy (1993, Lemma IV.1), and Sherali and Alameddine (1992, Theorem 4), by noting that *conv* $E_S(f)$ is a closed set since f is lower semicontinuous and S is a convex, compact set. □

Note that Proposition 8.6 is quite intuitive in that the convex hull of $E_S(f)$ is the smallest convex set that contains $E_S(f)$, and the convex envelope $f_S(x)$ of f over the set S is by definition the pointwise supremum of all convex (or even affine) underestimating functions of f over the set S.

Remark 8.2. Note that if the set S in Proposition 8.6 is nonempty, closed and convex, but not necessarily bounded, then the convex hull operator in (8.17) should be replaced by the *closure of the convex hull* for the result to hold true. For example, if $f(x) = 1 - e^{-x}$ and $S = \{x: \ x \geq 0\}$, then $f_S(x) = 0 \quad \forall \ x \in S$. However, *conv* $E_S(f)$ is not a closed set, but its closure is given by (8.17) as $\{(x, z): \quad x \in S,$ $z \geq f_S(x) \equiv 0\}$. Also, of related interest, if we replace $z \geq f(x)$ by $z = f(x)$ in (8.16), then we get analogous to (8.17) that

$$\begin{aligned} conv\ \{(x,z):\ \ x \in S, z = f(x)\} \equiv \{(x,z)&: \\ & x \in S, z \geq f_S(x), z \leq f^S(x)\} \end{aligned} \tag{8.18}$$

where $f^S(x)$ is the *concave envelope* of f over S, defined similar to its convex envelope as in Definition 8.1. □

The construct embodied by Proposition 8.6 provides a useful technique for computing the convex envelope of f over a convex set S. In particular, if the epigraph $E_S(f)$ defined in (8.16) can be represented using binary variables, for example, in a manner that facilitates the construction of its convex hull via the techniques described in Chapters 2 and 3, then Proposition 8.6 can be gainfully employed to derive the convex envelope of f. Denizel, Erenguc, and Sherali (1996) and Sherali (1996) describe several such instances. Below, we present one example for the sake of illustration.

Example 8.2. Let f: $H \to R$, where $H = \{x \in R^n: \ 0 \le x \le u \le \infty\}$ is a hypercube, and where

$$f(x) = \begin{cases} c \cdot x + k & \text{if } \sum_{j=1}^{n} x_j > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{8.19}$$

Then, let us show that

$$f_H(x) = c \cdot x + k \max_{j \in I} \left\{ \frac{x_j}{u_j} \right\} \quad \forall \ x \in H. \tag{8.20}$$

Observe that by defining an additional binary variable y, we can write

$$\begin{aligned} conv\, E_H(f) &\equiv conv\{(x, z): \ z \ge f(x), \ 0 \le x \le u\} \\ &= conv\{(x, z): \ z \ge c \cdot x + ky, \ 0 \le x \le uy, y \text{ binary}\}. \end{aligned} \tag{8.21}$$

But the convex hull of the set defined in (8.21) is obtained simply by relaxing y binary to $0 \leq y \leq 1$ since the extreme points of the resulting set all have 0-1 values of y. This follows by noting that the minimization of any linear objective function in (x, z, y) over this set which yields an optimum (that is, for which the objective coefficient of z is nonnegative), automatically yields an optimum at which y is binary valued. (This is readily verified by noting that in this problem, we can set $z = c \cdot x + ky$ at optimality, then solve for each x_j in terms of y by setting $x_j = u_j y$ if the resulting objective coefficient of x_j is negative, and zero otherwise, and finally, minimize the resulting linear objective term in y over $0 \leq y \leq 1$, yielding a binary value of y.) Hence, from (8.21), we have,

$$\begin{aligned} convE_H(f) &= \{(x, z): \; z \geq c \cdot x + ky, \; 0 \leq x \leq uy, \; 0 \leq y \leq 1\} \\ &= \left\{(x, z): \; z \geq c \cdot x + k \max_{j \in I} \left\{ \frac{x_j}{u_j} \right\}, \; 0 \leq x \leq u \right\}, \end{aligned} \tag{8.22}$$

where at the last step, we have eliminated y from the problem by projecting the region onto the (x, z) space, noting that we must have $1 \geq y \geq x_j / u_j \quad \forall \; j \in I$. Consequently, by Proposition 8.6 and Equation (8.22), we obtain the stated convex envelope (8.20). □

We now proceed to develop two results that make use of Proposition 8.6 to show that a particular application of RLT produces the closure of the convex hull of the set $\{(x, y, z): \; z \geq xy, \; (x, y) \in Z\}$, when Z is a triangular or quadrilateral D-polytope in

R^2. As in Sherali and Alameddine (1992), a *D-polytope* is one that has no finite, positively sloped edge. In other words, all the edges of Z are "downward" sloping, including possibly horizontally or vertically oriented edges. (Sherali and Alameddine, 1992, treat non-D-polytopes; this case is far more complex to analyze.) As we shall see, over D-polytopes, the epigraph of the convex envelope of bilinear functions is polyhedral. The results below demonstrate that a suitable RLT process recovers this characterization for the stated cases.

Henceforth, notationally, RLT-LP shall denote an application of the RLT in which the original defining constraints of the polyhedron are multiplied by each other pairwise as in (8.4b). Also, let us define Z_2 and Z_{L2} to be respectively regions defined by the new nonlinear constraints generated by applying RLT-LP, and by their linearized counterpart following the substitution of type (8.3).

Triangular D-Polytopes

To begin, consider the triangular D-polytope Z shown in Figure 8.1. The system of defining constraints associated with this polytope are

$$-\left[\frac{1}{r}\right]x - \left[\frac{1}{s}\right]y + 1 \geq 0 \tag{8.23a}$$

$$\left[\frac{s-q}{sp}\right]x + \left[\frac{1}{s}\right]y - 1 \geq 0 \tag{8.23b}$$

$$\left[\frac{1}{r}\right]x + \left[\frac{r-p}{rq}\right]y - 1 \geq 0. \tag{8.23c}$$

Note that by Proposition 8.4, there is no loss of generality in conveniently defining the origin relative to the polytope as shown. That is, given a D-polytope, if the origin is translated and then RLT-LP is applied to the resulting defining inequalities, the representation obtained can be verified to be equivalent to that obtained by applying RLT-LP to the original defining inequalities themselves.

For this triangular polytope, we have the following result which states that RLT-LP, with projection, constructs the convex envelope of the bilinear function $f(x, y) = xy$ over the region Z, and hence produces *cl.conv* $\{(x, y, z): \ z \geq xy, \ (x, y) \in Z\}$, where *cl.conv* denotes closure of the convex hull.

Proposition 8.7. *Consider the triangular D-polytope $Z \subset R^2$ shown in Figure 8.1. Using RLT-LP, generate the 3 constraints of Z_{L2} obtained by linearizing the constraint products $\{(8.23a) \cdot (8.23b), (8.23a) \cdot (8.23c),$ and $(8.23b) \cdot (8.23c)\}$ using the substitution $w = xy$, $X = x^2$, and $Y = y^2$. Define*

$$S = \{(x, y, z): \ z \geq xy, \ (x, y) \in Z\},$$

and denote

$$Z^p = \{(x, y, z): \ z \geq w, \ (x, y) \in Z, \ (x, y, w, X, Y) \in Z_{L2}\}$$

to be the projection onto the (x, y, z)-space of the set obtained by applying RLT-LP. Then $Z^p = cl.conv(S)$.

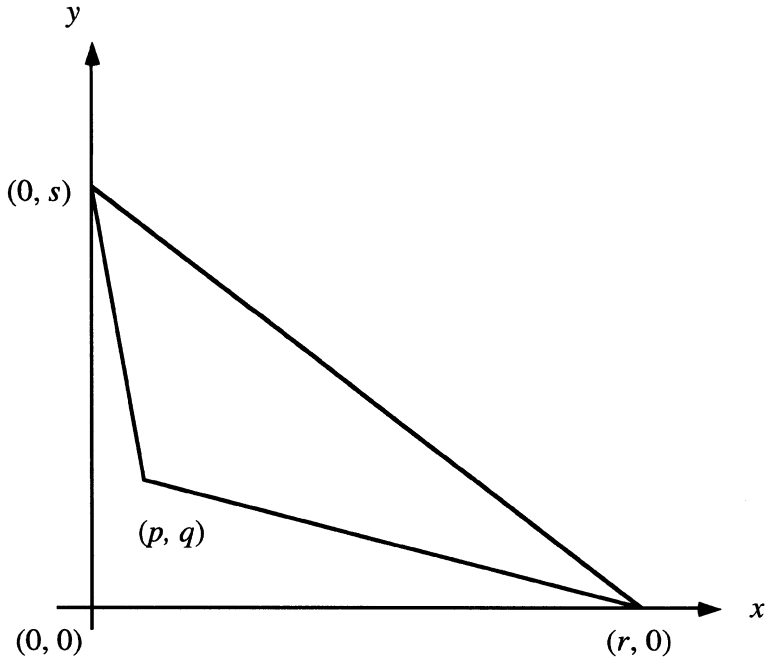


Figure 8.1. A triangular D-polytope in R^2.

Proof. Given any $(x, y, z) \in S$, by defining $w = xy$, $X = x^2$, and $Y = y^2$, it follows by construction that $(x, y, z) \in Z^p$. Hence, $S \subseteq Z^p$, and since Z^p is polyhedral, this means that $cl.\,conv(S) \subseteq Z^p$. To complete the proof, we need to show that $Z^p \subseteq cl.\,conv(S)$. By Proposition 8.6 and Remark 8.2, denoting $f(x, y) = xy$, we have that

$$cl.\,conv(S) = \{(x, y, z):\ z \geq f_Z(x, y),\ (x, y) \in Z\}.$$

Hence, given any $(x, y, z) \in Z^P$, we need to show that $z \geq f_Z(x, y)$. Noting the constraints of Z^P, it is sufficient to show that the constraints of Z_{L2} imply that $w \geq f_Z(x, y)$. This is the task undertaken below in order to prove the theorem.

First of all, note that from Sherali and Alameddine (1992), it is readily verified that

$$f_Z(x, y) = \left[\frac{-(pqs)x - (pqr)y + rspq}{(rs - sp - rq)}\right]. \tag{8.24}$$

Now, the constraints of Z_{L2} are given as follows.

$$-\left[\frac{s-q}{rsp}\right]X - \left[\frac{1}{s^2}\right]Y \geq -\left[\frac{rs + sp - rq}{rsp}\right]x - \left[\frac{2}{s}\right]y + \left[\frac{rs + sp - rq}{rs^2 p}\right]w + 1 \tag{8.25a}$$

$$-\left[\frac{1}{r^2}\right]X + \left[\frac{p-r}{rsq}\right]Y \geq -\left[\frac{2}{r}\right]x - \left[\frac{rs + rq - sp}{rsq}\right]y + \left[\frac{rs + rq - sp}{rs^2 q}\right]w + 1 \tag{8.25b}$$

$$\left[\frac{s-q}{rsp}\right]X + \left[\frac{r-p}{rsq}\right]Y \geq \left[\frac{rs + sp - rq}{rsp}\right]x + \left[\frac{rs + rq - sp}{rsq}\right]y - \left[\frac{rs - sp - rq + 2pq}{rspq}\right]w - 1. \tag{8.25c}$$

Denote the above set of inequalities (8.25) as $P\alpha \geq Q\beta + b$, where $\alpha = (X, Y)^t$, and $\beta = (x, y, w)^t$. Then, by linear programming duality, there exists an α that satisfies $P\alpha \geq Q\beta + b$ for a given β if and only if

$$(Q\beta + b)^t \pi \leq 0 \text{ for any } \pi \in \Lambda = \{\pi: \ P^t\pi = 0, \ \ e^t\pi = 1, \ \pi \geq 0\}. \tag{8.26}$$

The set Λ is explicitly given below.

$$-\left[\frac{s-q}{rsp}\right]\pi_1 - \left[\frac{1}{r^2}\right]\pi_2 + \left[\frac{s-q}{rsp}\right]\pi_3 = 0 \tag{8.27a}$$

$$-\left[\frac{1}{s^2}\right]\pi_1 - \left[\frac{r-p}{rsq}\right]\pi_2 + \left[\frac{r-p}{rsq}\right]\pi_3 = 0 \tag{8.27b}$$

$$\pi_1 + \pi_2 + \pi_3 = 1 \tag{8.27c}$$

$$\pi_1, \pi_2, \pi_3 \geq 0. \tag{8.27d}$$

Note that by (8.26),

$$\beta \equiv (x, y, w) \in Z_{L2} \text{ if and only if } (\bar{\pi})^t (Q\beta + b) \leq 0$$

$$\forall \ \bar{\pi} \in vert(\Lambda), \tag{8.28}$$

where $vert(\Lambda)$ denotes the vertices of Λ. Furthermore, note from (8.25) that

$$(Q\beta + b)^t \bar{\pi} = c_1 x + c_2 y + c_{12} w + (\bar{\pi}_1 + \bar{\pi}_2 - \bar{\pi}_3) \tag{8.29a}$$

where c_1, c_2, and c_{12} are given as follows:

$$c_1 = -\left[\frac{rs + sp - rq}{rsp}\right]\bar{\pi}_1 - \left[\frac{2}{r}\right]\bar{\pi}_2 + \left[\frac{rs + sp - rq}{rsp}\right]\bar{\pi}_3 \tag{8.29b}$$

$$c_2 = -\left[\frac{2}{s}\right]\bar{\pi}_1 - \left[\frac{rs + rq - sp}{rsq}\right]\bar{\pi}_2 + \left[\frac{rs + rq - sp}{rsq}\right]\bar{\pi}_3 \tag{8.29c}$$

$$c_{12} = \left[\frac{rs + sp - rq}{r^2 p}\right]\overline{\pi}_1 + \left[\frac{rs + rq - sp}{r^2 sq}\right]\overline{\pi}_2 + \left[\frac{rs - sp - rq + 2pq}{rspq}\right]\overline{\pi}_3. \tag{8.29d}$$

Now, it can be readily verified that Λ has a single vertex $\overline{\pi}$ given as the solution to the system (8.27a)-(8.27c). The solution to this subsystem is given below in closed-form

$$\overline{\pi}_1 = \frac{s(r - p)}{(3rs - sp - rq)} \geq 0, \ \overline{\pi}_2 = \frac{r(s - q)}{(3rs - sp - rq)} \geq 0,$$

$$\overline{\pi}_3 = \frac{rs}{(3rs - sp - rq)} \geq 0 \tag{8.30}$$

where the nonnegativity follows by noting that rs is an upper bound on the values of sp and rq. Letting $M = 3rs - sp - rq$ be a scaling parameter, and using (8.30) in (8.29), we have

$$Mc_1 = \left(\frac{-rs + sp + rq}{r}\right) \tag{8.31a}$$

$$Mc_2 = \left(\frac{-rs + sp + rq}{s}\right) \tag{8.31b}$$

$$Mc_{12} = -\left[\frac{(rs - sp - rq)^2}{rspq}\right]. \tag{8.31c}$$

Furthermore, from (8.30), we have,

$$M(\overline{\pi}_1 + \overline{\pi}_2 - \overline{\pi}_3) = rs - sp - rq. \tag{8.32}$$

Hence, putting (8.28) and (8.29a) together, we have that

$(x, y, w) \in Z_{L2}$ if and only if $M[c_1 x + c_2 y + c_{12} w + (\overline{\pi}_1 + \overline{\pi}_2 - \overline{\pi}_3)] \leq 0.$

Noting that $c_{12} < 0$, and writing the foregoing condition as

$$w \geq -\left(\frac{c_1}{c_{12}}\right)x - \left(\frac{c_2}{c_{12}}\right)y - \left[\frac{(\bar{\pi}_1 + \bar{\pi}_2 - \bar{\pi}_3)}{c_{12}}\right].$$

we see from (8.24), (8.31), and (8.32) that this is precisely $w \geq f_Z(x, y)$. This completes the proof. □

Quadrilateral D-Polytopes

Next, consider the quadrilateral D-polytope Z shown in Figure 8.2. The system of defining constraints associated with this polytope Z is

$$-\left[\frac{1}{r}\right]x - \left[\frac{1}{s}\right]y + 1 \geq 0 \tag{8.33a}$$

$$\left[\frac{s-q}{sp}\right]x + \left[\frac{1}{s}\right]y - 1 \geq 0 \tag{8.33b}$$

$$\left[\frac{q-u}{tq-pu}\right]x + \left[\frac{t-p}{tq-pu}\right]y - 1 \geq 0 \tag{8.33c}$$

$$\left[\frac{1}{r}\right]x + \left[\frac{r-t}{ru}\right]y - 1 \geq 0. \tag{8.33d}$$

Similar to Proposition 8.7, for the quadrilateral polytope Z shown in Figure 8.2, we have the following result. (This result also holds for quadrilateral D-polytopes having shapes other than that in Figure 8.2, including rectangles — see Corollary 8.1 below.)

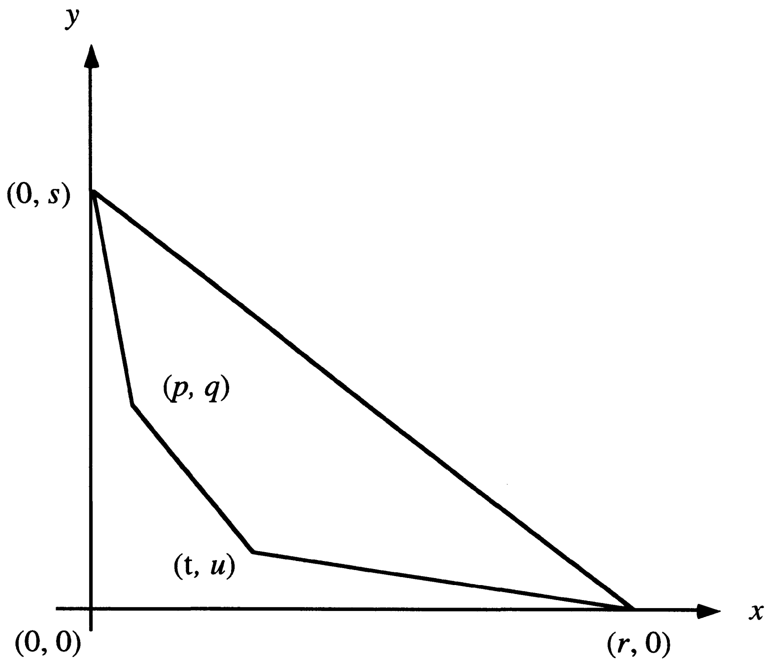


Figure 8.2. A quadrilateral D-polytope in R^2.

Proposition 8.8. *Consider the quadrilateral D-polytope $Z \subset R^2$ shown in Figure 8.2. Using RLT-LP, generate the 6 constraints of Z_{L2} by linearizing the constraint products $\{(8.33a) \cdot (8.33b), (8.33a) \cdot (8.33c), (8.33a) \cdot (8.33d), (8.33b) \cdot (8.33c), (8.33b) \cdot (8.33d)$ and $(8.33c) \cdot (8.33d)\}$, by using the substitution $w = xy$, $X = x^2$, $Y = y^2$. Define $S = \{(x, y, z)\colon\ z \geq xy,\ \ (x, y) \in Z\}$, and denote $Z^p = \{(x, y, z)\colon\ z \geq w,\ (x, y) \in Z,\ (x, y, w, X, Y) \in Z_{L2}\}$ to be the projection onto the (x, y, z)-space of the set obtained by applying RLT-LP. Then $Z^p = cl.conv(S)$.*

Proof. Following the proof of Proposition 8.7, we identically have that $cl.conv(S) \subseteq Z^P$, and in order to show that $Z^P \subseteq cl.conv(S)$, we need to show that the constraints of Z_{L2} imply that $w \geq f_Z(x, y)$.

Toward this end, let $P\alpha \geq Q\beta + b$ denote the inequalities of Z_{L2} generated by the pairwise products given in the theorem, *in that order,* where $\alpha = (X, Y)^t$ and $\beta = (x, y, w)^t$. Then the projection Z^P of this constraint set onto the space of the variables $\beta = (x, y, w)^t$ is given by

$$Z^P \equiv \{(x, y, w): \ (x, y) \in Z, \ (x, y, w)Q^t\pi^k + b^t\pi^k \leq 0, \ \forall \ k = 1, \ldots, K\},$$

where π^k, $k = 1, \ldots, K$, are the vertices of the set $\Lambda = \{\pi: \ P^t\pi = 0, \ e^t\pi = 1, \pi \geq 0\}$, and where $e^t = (1, \ldots, 1)$. Note that the vector π equals $(\pi_1, \ldots, \pi_6)^t$, and that Λ has three equality constraints that can be verified to be linearly independent. Hence, extreme points of Λ have three basic variables. Now, consider two cases.

Case 1. Suppose that the ratio

$$\left[\frac{tu(sp + rq - rs}{pq(st + ru - rs)}\right] > 1. \tag{8.34}$$

Consider the following two choices of bases for Λ, where the three designated nonzero components of each solution π^q, $q = 1, 2$, are considered as basic.

$$\pi^1 = (\bar{\pi}_1, \bar{\pi}_2, 0, 0, \bar{\pi}_5, 0)^t \tag{8.35a}$$

$$\pi^2 = (0, \bar{\pi}_2, 0, 0, \bar{\pi}_5, \bar{\pi}_6)^t. \tag{8.35b}$$

It can be verified, albeit through tedious algebraic manipulations, that these solutions are indeed vertices of Λ, and that the two constraints $(x, y, w)Q^t\pi^k + b^t\pi^k \le 0$, $k = 1, 2$, define the convex envelope of $f(x, y) = xy$ over Z. In fact, these constraints represent the convex envelopes of f over the triangular regions ABD and BCD, respectively, shown in Figure 8.3a, which are described by the following explicitly stated supports:

$$w \ge \left[\frac{-(spq)x - (rpq)y + rspq}{(rs - rq - sp)}\right] \tag{8.36a}$$

$$w \ge \left[\frac{u(t - \beta_4^*)}{(t - r)}\right]x + \left[\frac{pq(t - r) + tu(r - p)}{q(t - r) + u(r - p)}\right]y - \left[\frac{ru(t - \beta_4^*)}{(t - r)}\right], \tag{8.36b}$$

where

$$\beta_4^* = \left[\frac{pq(t - r) + tu(r - p)}{q(t - r) + u(r - p)}\right].$$

Hence, in Case 1, the constraints of Z_{L2} imply that $w \ge f_Z(x, y)$.

Case 2. On the other hand, suppose that the ratio in (8.34) satisfies

$$0 \le \left[\frac{tu(sp + rq - rs)}{pq(st + ru - rs)}\right] \le 1. \tag{8.37}$$

Consider the following two choices of bases for Λ, where the three designated nonzero components of each solution π^q, $q = 1, 2$ are considered as basic.

$$\pi^3 = (0, \bar{\pi}_2, \bar{\pi}_3, 0, \bar{\pi}_5, 0)^t \tag{8.38a}$$

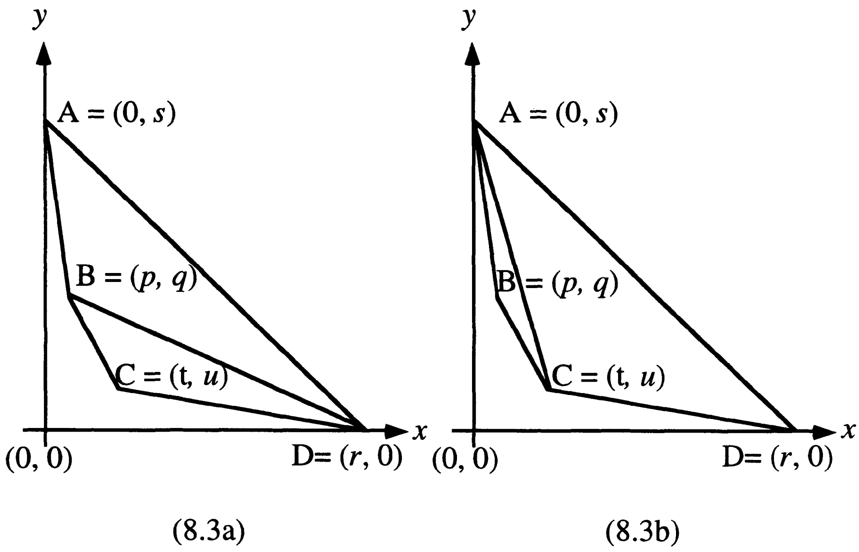


Figure 8.3. Triangular decomposition for Proposition 8.8.

$$\pi^4 = (0, \bar{\pi}_2, 0, \bar{\pi}_4, \bar{\pi}_5, 0)^t. \tag{8.38b}$$

Again, it can be shown that these are indeed vertices of Λ, and that the two constraints $(x, y, w)Q^t\pi^k + b^t\pi^k \le 0$, $k = 1, 2$, define the convex envelope of $f(x, y) = xy$ over Z. In fact, these constraints represent the convex envelopes of f over the triangular regions ACD and ABC, respectively, shown in Figure 8.3b, which are described by the following explicitly stated supports:

$$w \ge \left[\frac{-(stu)x - (rtu)y + rstu}{(rs - ru - st)}\right] \tag{8.39a}$$

$$w \ge \left[\frac{q[t(s-q) + p(u-s)] + t(u-q)(s-q)}{t(s-q) + p(u-s)}\right]x$$

$$+\left[\frac{pt(u-q)}{t(s-q)+p(u-s)}\right]y-\left[\frac{spt(u-q)}{t(s-q)+p(u-s)}\right]. \tag{8.39b}$$

Hence, we again have in Case 2, that the constraints of Z_{L2} imply that $w \geq f_Z(x, y)$, and this completes the proof. □

From Figures (8.3a) and (8.3b), note that the diagonals BD and AC each partition the area of quadrilateral ABCD into two triangular regions. Proposition 8.8 gives the condition under which either diagonal is used in the triangular partitioning process, and states that it is dependent on the ratio:

$$\left[\frac{tu(sp+rq-rs)}{pq(st+ru-rs)}\right].$$

Geometrically, this ratio represents the relative distances of the points (p, q) and (t, u) from the longest side AD.

Corollary 8.1 (Result of Al-Khayyal and Falk, 1983). *Let Ω be a hyperrectangle $\Omega = \{(x, y)\colon\ 0 \leq \ell \leq x \leq u < \infty,\ 0 \leq L \leq y \leq U < \infty\} \subseteq R^2$, and consider the weighted bilinear function $f(x, y) = cxy$, where W might be of either sign. Then,*

$$f_\Omega(x, y) = c \cdot max\{Lx + \ell y - \ell L,\ Ux + uy - uU\} \text{ if } c \geq 0, \tag{8.40a}$$

and

$$f_\Omega(x, y) = c \cdot min\{Ux + \ell y - U\ell,\ Lx + uy - uL\} \text{ if } c < 0\ . \tag{8.40b}$$

Proof. By Proposition 8.6 and 8.8, we have

$$\{(x, y, z): \ (x, y) \in \Omega, \ z \geq [xy]_{\Omega}\} \equiv conv\{(x, y, z): \ (x, y) \in \Omega, \ z \geq xy\}$$

$$= \{(x, y, z): \ (x, y) \in \Omega, \ z \geq w, \text{and } (x, y, w, X, Y) \in Z_{L2}\} \tag{8.41}$$

where Z_{L2} is produced by the linearization of the pairwise products of the bounding constraints. This yields the following set of constraints

$$[(x - \ell)(y - L)]_L \geq 0, \text{ yielding } w \geq Lx + \ell y - \ell L \tag{8.42a}$$

$$[(u - x)(U - y)]_L \geq 0, \text{ yielding } w \geq Ux + uy - uU \tag{8.42b}$$

$$[(x - \ell)(U - y)]_L \geq 0, \text{ yielding } w \leq Ux + \ell y - U\ell \tag{8.42c}$$

$$[(u - x)(y - L)]_L \geq 0, \text{ yielding } w \leq Lx + uy - uL \tag{8.42d}$$

$$[(u - x)(x - \ell)]_L \geq 0, \text{ yielding } X \leq (u + \ell)x - u\ell \tag{8.42e}$$

$$[(U - y)(y - L)]_L \geq 0, \text{ yielding } Y \leq (U + L)y - UL \ . \tag{8.42f}$$

Note that the first four inequalities in (8.42) yield the consistent relationship

$$\max\{Lx + \ell y - \ell L, \ Ux + uy - uU\} \leq w \leq \min\{Ux + \ell y - U\ell, \\ Lx + uy - uL\} \tag{8.43}$$

and (8.42e) and (8.42f) are inconsequential to determining w. Hence, in (8.41), z is determined via $z \geq w$ using the left-hand inequality in (8.43), thereby yielding (8.40a) when $c \geq 0$.

On the other hand, when $c < 0$, since $cxy \equiv |c|(-xy)$, we are interested in determining the convex envelope of $-xy$ over Ω. Putting $x' = -x$, we seek the convex envelope of $x'y$ over $\Omega' = \{(x', y)\colon\ -u \le x' \le -\ell,\ L \le y \le U\}$. By (8.40a), we obtain

$$[cxy]_{\Omega} = [|c|(-xy)]_{\Omega} = |c|(-x'y)_{\Omega'}$$

$$= |c| \max\{Lx' - uy + uL, Ux' - \ell y + U\ell\}$$

$$= -|c| \min\{Lx + uy - uL,\ Ux + \ell y - U\ell\}$$

which is (8.40b), noting that $c < 0$. (Observe that this is tantamount to having used $z \ge -|c|\,xy = -|c|\,w$ along with the right-hand inequality in (8.43).) This completes the proof. □

In the light of Corollary 8.1, it is instructive to note that the concave envelope of the function xy over Z is not available via the RLT scheme in the above cases. This follows because the concave envelope is given by the negative of the convex envelope of $-xy$ over Z. Substituting $x' = -x$, this reduces to determining the convex envelope of $x'y$ over a corresponding set Z', as in the proof of Corollary 8.1. However, Z' need not be a D-polytope in general, and in fact, the convex envelope of $x'y$ over Z' need not even be polyhedral.

Furthermore, it might be tempting to conjecture that Propositions 8.7 and 8.8 can be extended to higher order D-polytopes. Unfortunately, for the bilinear function xy defined over a pentagonal D-polytope, RLT-LP fails to produce the convex hull representation. For example, for the pentagon shown in Figure 8.4, one can verify that the convex hull

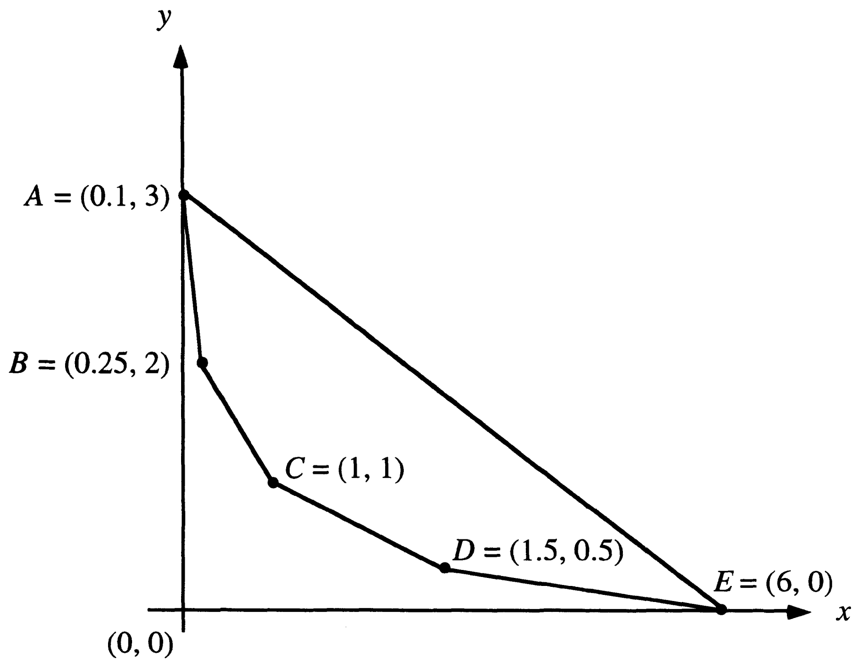


Figure 8.4. Pentagonal D-polytope in R^2.

representation is not produced by RLT-LP. However, this representation is indeed produced by the RLT application that also considers products of constraints taken *three at a time*. This therefore motivates a possible consideration for higher order products, as also suggested by Example 8.1 of Section 8.2 above.

8.5. Enhancements of the RLT Relaxation: RLT-NLP

We now propose two enhancements for the foregoing RLT relaxation for Problem QP. First, we employ an eigen-transformation on QP to generate an equivalent problem for which the first-level RLT representation can be constructed. However, the separable structure of the revised objective function now enables us to identify the RLT constraints that play an important role in providing a tight representation and the ones that do not. Translating this information for RLT-LP, permits us to reduce the size of the latter problem, without compromising on its tightness. Second, motivated by the same constructs, we include certain separable convex quadratic constraints within RLT-LP to derive a revised relaxation RLT-NLP. The structure of these constraints is such that they pose no additional burden in the context of a Lagrangian Relaxation optimization scheme, while they contribute additional strength to the resulting RLT relaxation. Each of these enhancements is presented in separate subsections below.

8.5.1. Eigen-Transformation (RLT-NLPE) and Identification of Selected Constraints (SC)

Let us begin by using a particular linear transformation based on the eigenstructure of the quadratic objective function. Let $Q = PDP^t$, where D is a diagonal matrix whose diagonal elements correspond to the eigenvalues λ_i of Q, $i = 1,\ldots,n$, and P is a matrix whose columns correspond to orthonormal eigenvectors of Q (see Golub and Van Loan, 1989, for example). Define $x = Pz$, so that $z = P^t x$. The resulting eigen-transformed quadratic program is then obtained as follows.

$$\underset{z \in R^n}{\text{Minimize}}\{cPz + z^t Dz: \quad APz \le b,\ \ell \le Pz \le u\}. \tag{8.44}$$

We next construct the first-level RLT for this quadratic problem (8.44) as before, defined in terms of z and the RLT-variables $y_{k\ell}$, $1 \le k \le \ell \le n$, but now, we further tighten the relaxation by including the nonlinear convex constraints $z_k^2 \le y_{kk} \equiv [z_k^2]_L$, $\forall\ k = 1,\dots,n$, in this model. Let us refer to the resulting problem as **RLT-NLPE.**

Let us now attempt to reduce the size of the problem RLT-NLPE by eliminating those constraints that might not contribute significantly to determining an optimal solution. Toward this end, denote the index set of "*concave*" variables by $N_v \equiv \{k:$ the eigen value $\lambda_k < 0,\ k = 1,\dots,n\}$. Renaming the constraints in (8.44) collectively as $F_i z \le f_i,\ i = 1,\dots,m+2n$, let us re-organize each second-order RLT constraint $[(f_i - F_i z)(f_j - F_j z)]_L \ge 0$ from RLT-NLPE as $\sum_{k \in N_v} -F_{ik}F_{jk}y_{kk} \le$ [the rest of the constraint]. On the left-hand side of the re-organized RLT constraint, if the positive coefficients have at least as much weight as the negative coefficients, then it is more likely that this constraint produces relatively strong upper bounds on the y_{kk}-variables that have positive coefficients, *i.e.*, for which $-F_{ik}F_{jk} > 0,\ k \in N_v$. This is useful since the objective coefficients λ_k of these y_{kk} variables are negative. Therefore, assuming that the original problem has been scaled so that all variables are roughly commensurate with each other, we suggest generating only those RLT constraints that satisfy the condition $(\sum_{k \in N_v} -F_{ik}F_{jk}) \ge 0$, and suppressing the rest of them. *Accordingly, we also retain in RLT-LP the constraints that correspond to those retained within RLT-NLPE, and we discard the remaining constraints in RLT-LP.* Observe that we can no longer guarantee that the original constraints are implied by the selected RLT constraints, and so, we include the original constraints $Gx \le g$ in the reduced RLT-LP problem. Let us call this reduced relaxation having only *selected constraints* as **RLT-LP(SC)**.

8.5.2. Reformulation-Convexification Approach: Inclusion of Suitable Nonlinear Constraints in RLT-LP to Derive RLT-NLP

Observe that in RLT-LP, the following linear RLT constraints

$$[(x_k - \ell_k)^2]_L \geq 0,\ [(u_k - x_k)^2]_L \geq 0,\ [(x_k - \ell_k)(u_k - x_k]_L \geq 0,$$
$$k = 1, \dots, n \tag{8.45}$$

approximate the relationship $w_{kk} = x_k^2$ over the interval $\ell_k \leq x_k \leq u_k$, for $k = 1, \dots, n$. Motivated by RLT-NLPE, we propose to replace (8.45) by the nonlinear constraints

$$x_k^2 \leq w_{kk} \leq (u_k + \ell_k)x_k - u_k\ell_k,\ \ell_k \leq x_k \leq u_k,\ k = 1, \dots, n \tag{8.46}$$

and call the resulting nonlinear problem as **RLT-NLP**, and its corresponding reduced version as derived in Section 8.5.1 as **RLT-NLP(SC)**. Notice that the upper bounding linear function in (8.46) is precisely the last constraint in (8.45). The improvement via (8.46), which happens to imply (8.45), appears in the case when the problem RLT-LP tries to reduce the value of w_{kk} for some $k \in \{1, \dots, n\}$ in the relaxed solution. Note that the first two constraints in (8.45) merely *approximate* the function $w_{kk} = x_k^2$ from below via tangential supports at the points ℓ_k and u_k. On the other hand, since (8.46) produces the exact lower envelope, it is equivalent to having an additional tangential support at the optimal point included within RLT-LP. Therefore, the enhancement (8.46) corresponds to a tighter bounding nonlinear problem than the linear program RLT-LP.

Note that RLT-NLP and RLT-NLPE are no longer equivalent relaxations. Although RLT-NLPE usually turns out to yield a tighter representation because the additional nonlinear constraints provide a better support for the "*convex*" variables (those associated with positive eigenvalues, *i.e.*, $\lambda_k > 0$), the loss in structure due to the increased density in the bounding constraints inhibits the development of an efficient solution scheme. Hence, we recommend the use of RLT-NLP. (See Section 8.7 for some related computational results.)

Remark 8.3. To further strengthen the bounding problem, we can derive two additional classes of linear constraints based on projecting cubic bound-factor products onto the quadratic space, and on squaring *differences* of bound (or constraint)-factors. However, although these constraints serve to tighten the relaxation somewhat, they increase the problem size considerably. Hence, due to the ensuing computational burden, they will not be used in the overall branch-and-bound algorithm. (We refer the interested reader to Tuncbilek, 1994, for the generation of such constraints; also, see Section 8.7 for some related computational results.)

8.6. A Lagrangian Dual Approach for Solving RLT Relaxations

Given a problem QP that has n variables and (up to) $m + 2n$ constraints, the corresponding (non-reduced) first-level RLT problem has $n(n + 1)/2$ additional variables, and a total of $(m + 2n)(m + 2n + 1)/2$ constraints. It is clear that the size of RLT-LP, or even RLT-NLP, gets quite large as the size of QP increases. If we can obtain a tight lower bound on RLT-NLP with relative ease, then we can trade-off between the quality of the bound and the effort necessary to obtain it. For this purpose, we propose to use a Lagrangian relaxation of RLT-NLP (see Fisher, 1981, for example), and solve *—not necessarily exactly* — the Lagrangian Dual Problem **LD-RLT-NLP**.

To define the proposed Lagrangian dual for RLT-NLP, we dualize all but the constraints $[(x_k - \ell_k)(x_\ell - \ell_\ell)]_L \geq 0,$ $[(u_k - x_k)(x_\ell - \ell_\ell)]_L \geq 0,$ $[(u_k - x_k)(u_\ell - x_\ell)]_L \geq 0$ and $[(x_k - \ell_k)(u_\ell - x_\ell)]_L \geq 0,$ $\forall\ 1 \leq k < \ell \leq n,$ from the set (8.4b), and we also do not dualize the constraints (8.46) which have now replaced (8.45). These constraints, along with the bounds $\ell_k \leq x_k \leq u_k,$ $k = 1, \ldots, n,$ on the x-variables, comprise the Lagrangian subproblem constraints (see Fisher, 1981). Note that the foregoing linearized bound-factor product constraints in the subproblem yield lower and upper bounding linear functions for the linearized cross product terms $w_{k\ell}$, for all $1 \leq k < \ell \leq n$. These constraints can be expressed in open form as follows:

$$\ell_\ell x_k + \ell_k x_\ell - \ell_k \ell_\ell \leq w_{k\ell} \leq \ell_\ell x_k + u_k x_\ell - u_k \ell_\ell \quad \forall\ 1 \leq k < \ell \leq n \tag{8.47a}$$

$$u_\ell x_k + u_k x_\ell - u_k u_\ell \le w_{k\ell} \le u_\ell x_k + \ell_k x_\ell - \ell_k u_\ell \quad \forall\ 1 \le k < \ell \le n. \tag{8.47b}$$

In addition, following the layering strategy proposed by Guignard and Kim (1987), in order to facilitate the solution of the Lagrangian subproblems, we replace $w_{k\ell}$ by $w'_{k\ell}$ $\forall\ 1 \le k < \ell \le n$ in (8.47b), and we include in RLT-NLP the constraints

$$w'_{k\ell} = w_{k\ell} \quad \forall\ 1 \le k < \ell \le n. \tag{8.48}$$

These new constraints (8.48) are also dualized using some defined Lagrange multipliers. Hence, in order to solve the Lagrangian subproblem, depending on the signs of the coefficients for each $w_{k\ell}$ and $w'_{k\ell}$ variable in the Lagrangian subproblem objective function, we first replace this variable in the objective function appropriately by either its lower or upper bounding function in terms of x as defined in (8.47). The resulting subproblem objective function is then given in terms of the variables $\{x_k$, and w_{kk}, for $k = 1, \ldots, n\}$ alone, and the subproblem constraints are now defined by (8.46) along with the bounds on the x-variables. This reduced separable Lagrangian dual subproblem can be stated as follows.

$$\sum_{k=1}^{n} \text{Minimize } \{\hat{c}_k x_k + \hat{q}_{kk} w_{kk} \colon\ x_k^2 \le w_{kk} \le (u_k + \ell_k) x_k - u_k \ell_k,$$
$$\ell_k \le x_k \le u_k\}. \tag{8.49}$$

Suppose that instead of the constraints in (8.49), we use the restrictions

$$w_{kk} = x_k^2,\ \ \ell_k \le x_k \le u_k,\ \ k = 1, \ldots, n \tag{8.50}$$

to obtain the reduced Lagrangian dual subproblem in the following more favorable form.

$$\text{Minimize } \{\sum_{k=1}^{n} \hat{c}_k x_k + \sum_{k=1}^{n} \hat{q}_{kk} w_{kk} : \text{ (8.50)}\}$$

$$= \sum_{k=1}^{n} \underset{\ell_k \le x_k \le u_k}{\text{minimize}} \{\hat{c}_k x_k + \hat{q}_{kk} x_k^2\}. \tag{8.51}$$

The following proposition asserts that the simpler problem (8.51) equivalently solves (8.49).

Proposition 8.9. *Problem (8.49) is equivalently solved via Problem (8.51).*

Proof. For any $k \in \{1, \ldots, n\}$, if $\hat{q}_{kk} > 0$, then in (8.49), w_{kk} is replaced by x_k^2 in the objective function, which makes it equivalent to (8.51). If $\hat{q}_{kk} \le 0$, then in (8.49), if w_{kk} is replaced by $(u_k + \ell_k)x_k - u_k \ell_k$, the revised objective function becomes a linear function of the variable x_k. Consequently, an optimal value for x_k occurs at either of its bounds, and so, the constraints on w_{kk} in (8.49) are satisfied as equalities at optimality. Examining the same case for (8.51), we similarly obtain a concave univariate quadratic minimization problem over simple bounds, and so, an optimal value for x_k occurs at either of the bounds. Also, since $(u_k + \ell_k)x_k - u_k \ell_k = x_k^2$ at either bound, both (8.51) and (8.49) produce the same optimal solutions. This completes the proof. □

Remark 8.4. The Lagrangian dual problem is a nondifferentiable optimization problem. To solve this problem in our implementation, we adopted the subgradient deflection algorithm of Sherali and Ulular (1989), using their recommended parameter values. After running this algorithm for up to 200 iterations, another 50 iterations were performed in a reduced subspace by fixing the dual variables having relatively small magnitudes (less than half of the average magnitude of all the dual values) at their current level in the given incumbent solution, and polishing the remaining dual variable values. Finally, a dual ascent was performed by finding optimal values for the dual variables associated with the constraints (8.48) one at a time, in a Gauss-Seidel fashion, given the remaining dual variable values. This can be executed with little additional effort, and so, it provides a quick dual ascent step (see Tuncbilek, 1994, for details).

8.7. A Preliminary Computational Comparison of the Bounding Problems

To evaluate and compare the different approaches for generating and solving a level-one RLT relaxation, we performed some preliminary empirical experiments using test problems from the literature. Using three bilinear programming problems (BLP1, BLP2, BLP3) from Al-Khayyal and Falk (1983), and two concave programming problems (CQP1, CQP2) from Floudas and Pardalos (1990), we solved the first level linear RLT problem RLT-LP, the nonlinear RLT problem RLT-NLP, the eigen space first level nonlinear RLT problem RLT-NLPE, and the reduced RLT-NLP that uses only selected constraints (denoted RLT-NLP(SC)), as well as the Lagrangian dual problems corresponding to the latter three (denoted by the prefix **LD-**). For these computations, the linear and nonlinear programs were solved using GAMS along with the solver

MINOS 5.2, and a FORTRAN code was written for solving the Lagrangian dual problems as per Remark 8.4. All the runs were conducted on an IBM 3090 mainframe computer. Table 8.1 reports the resulting lower bounds obtained along with the solution times required to generate these bounds.

RLT-LP yields lower bounds very close to the actual global minimum for all the problems except for BLP1 (BLP2 and BLP3 are solved exactly). RLT-NLPE, where we have included the nonlinear constraints $z_k^2 \le y_{kk}$ only for $k \in \{1,\ldots,n\}$ such that $\lambda_k \ge 0$, solves all problems to (near) optimality. However, the computational effort is considerably increased since this lower bounding problem is a nonlinear programming problem. When we include the nonlinear constraints $z_k^2 \le y_{kk}$ for all $k \in \{1,\ldots,n\}$ in RLT-NLPE, the imposed resource limit of 1000 cpu seconds for GAMS is exceeded for BLP2. On the other hand, we were able to solve RLT-NLP, even while including all n nonlinear constraints, and it produced solutions comparable to those obtained by RLT-NLPE. Reconsidering RLT-LP, but this time, including the additional two classes of constraints mentioned in Remark 8.3 of Section 8.5, the lower bound improved for BLP1 to -1.125, and that for CQP2 to -39.47, the latter of which is greater than any of the bounds reported in Table 8.1 for this problem. However, the computational effort for CQP2 increased to 13.36 cpu seconds.

Although the Lagrangian dual problem LD-RLT-NLP should give the same lower bound as does RLT-NLP, the bounds for BLP3 and CQP2 are somewhat worsened due to the

Table 8.1. Comparison of RLT schemes.

Problem	m	n	Known v[RLT-LP]	v[RLT-LP]	cpu secs.
BLP1	2	2	−1.083	−1.50	0.042
BLP2	10	10	−45.38	−45.38	3.75
BLP3	13	10	−794.86	−794.86	22.74
CQP1	11	10	−267.95	−268.02	9.72
CQP2	5	10	−39.00	−39.83	3.23

Problem	v[RLT-NLPE]	cpu secs.	v[LD-RLT-NLPE]	cpu secs.
BLP1	−1.083	0.086	−1.097	0.09
BLP2	−45.38	31.23	−69.17	0.81
BLP3	−794.76	48.55	−11586.88	0.97
CQP1	−268.02	12.16	−269.87	0.85
CQP2	−39.83	4.69	−43.38	0.65

Problem	v[RLT-NLP]	cpu secs.	v[LD-RLT-NLP]	cpu secs.
BLP1	−1.089	0.105	−1.089	0.02
BLP2	−45.38	9.79	−46.10	0.43
BLP3	−794.86	78.41	−829.52	0.57
CQP1	−268.02	13.30	−269.83	0.48
CQP2	−39.83	4.69	−43.93	0.25

Problem	v[RLT-NLP(SC)]	cpu secs.	$\frac{\text{used}}{\text{total}}$ constrs.	v[LD-RLT-NLP(SC)]
BLP1	−1.089	0.091	15 / 21	−1.089
BLP2	−45.38	6.07	339 / 465	−45.62
BLP3	−806.53	32.84	387 / 564	−841.60
CQP1	−268.02	13.61	352 / 497	−269.68
CQP2	−40.10	2.54	261 / 320	−42.99

Legend: (m, n) = size of the problem QP, $v[\cdot]$ = optimal solution of problem $(\cdot)$, cpu secs. = cpu seconds to solve the problem on an IBM 3090 computer.

inherent difficulty in solving nondifferentiable optimization problems. Nevertheless, the attractive computational times, especially when the problem size increases, makes this method our first choice to be used in the branch-and-bound algorithm proposed in Section 8.8 below.

In adopting the Lagrangian dual approach of Section 8.6 for RLT-NLPE, the $2n(n-1)$ RLT constraints that are generated using the constraints $\ell \le Pz \le u$ were dualized. Implied simple bounds on the z-variables were computed by minimizing and then maximizing each row of $z = P^t x$ over $\ell \le x \le u$ to obtain counterparts of both the constraints (8.47) and the subproblem (8.51) in z-space. Ideally, we would like to use LD-RLT-NLPE as the bounding problem for the overall algorithm. However, due to its dense structure, LD-RLT-NLPE tends to perform poorly as for problems BLP2 and BLP3, although the bounds for CQP1 and CQP2 are slightly improved over those obtained via LD-RLT-NLP. Upon using the tightest simple bounds on the z-variables that contain the feasible region, the lower bound for BLP3 improved considerably to -1507.64, but it is still 81% lower than that obtained via LD-RLT-NLP.

Applying the constraint selection strategy of Section 8.5.1 on RLT-NLP, we solved the problem RLT-NLP(SC) with reduced computational effort, although at an expense of a 1.5% decrease in the lower bound for BLP3. (Roughly 18-30% of the constraints are deleted by this strategy.) For the same problem, v[LD-RLT-NLP(SC)] has also worsened by 1.5% compared to v[LD-RLT-NLP]. However, for the rest of the problems, LD-RLT-NLP(SC) actually improved the lower bounds. We have also observed that there is only a negligible increase in the required computational effort compared to that consumed by LD-RLT-NLP. Section 8.9 reports on computational experiments using RLT-NLP, LD-RLT-NLP, and LD-RLT-NLP(SC) within the branch-and-bound algorithm described next.

8.8. Design of a Branch-and-Bound Algorithm

In this section, we first discuss design issues for various features or elements of the proposed branch-and-bound algorithm, and then present a statement of this procedure.

8.8.1. Scaling

Our experiments with the subgradient deflection algorithm have indicated that scaling plays an important role in the performance of this algorithm. Among several scaling methods we tried, including a sophisticated iterative method used in the package MINOS (see Murtagh and Saunders, 1987), a simple one seemed to work well. This method scales the variables such that the lower and upper bounds are mapped onto the hypersquare $[0, 1]^n$. In addition, a row scaling is employed that divides each constraint $a_i x \leq b_i$, $i \in \{1, \ldots, m\}$, by the ℓ_∞-norm of (a_i, b_i).

8.8.2. Branch-and-Bound Search Strategy

The branch-and-bound search strategy employed uses a hybrid of the depth-first search and the best-first search strategies as suggested by Sherali and Myers (1985/86), where at most a fixed (*MAXACT*) number of nodes are kept active in the branch-and-bound tree.

We used the largest value of *MAXACT* as permissible by storage limitations, depending on the size of the problem. In this approach, branching is performed by splitting the bounding interval of a variable as stated in Section 8.8.5 below.

8.8.3. Optimality Criterion

To avoid undue excessive computations involved in sifting through alternative optimal solutions or close to global optimal solutions, we adopted the fathoming criterion

$$LB \geq UB - \varepsilon \max\{1, |UB|\} \tag{8.52}$$

where $0 < \varepsilon < 1$, and where LB is a valid lower bound at the current branch-and-bound node, and UB is the current best (incumbent) solution value for QP. Hence, when the algorithm stops, we can claim that the global minimum is within $100\varepsilon\%$ (or within ε) of the current best solution.

8.8.4. Range Reductions

The tightness of the lower and upper bounds that define the *box constraints* on the variables play a major role in the performance of the Lagrangian dual bounding scheme. Fast and effective procedures to improve simple bounds, known as "logical tests," have acquired a good reputation in (mixed) integer 0-1 programming. In addition to showing that these types of tests have their counterparts in global optimization, Hansen, Jaumard and Lu (1991) have also developed new tests using analytical and numerical methods. In the same spirit, we also propose some suitable strategies for improving lower and upper bounds on the variables at each node of the branch-and-bound tree. The first procedure is based on knapsack problems defined by individual linear functional constraints along with the box constraints. The second procedure is motivated by the number of constraints that have to be binding at optimality. The third and the fourth procedures are applied to cutting planes based on, respectively, the Lagrangian dual objective function, and the eigen-transformed separable objective function. The latter three strategies are based on some appropriate optimality conditions, by which we can further tighten the box constraints beyond feasibility considerations, by discarding regions that cannot contain an optimal solution. Since the bounds are tightened based on optimality related considerations, they are possibly not implied by the already existing constraints, and in the light of Proposition 8.2, this helps to generate tighter RLT bounding problems as well. Below, we summarize these range reduction strategies. Other strategies can be devised based on minimizing and maximizing each variable in turn in a cyclic fashion, over RLT based constraints, inclusive of a linearized objective function cutting plane. For further details, see Chapter 9 as well as Tuncbilek (1994).

Range Reduction Strategy 1: This is a strategy for tightening variable bounds by virtue of a simple feasibility check. By minimizing the left-hand side expressions of the less than or equal to type functional constraints of QP in turn over the box constraints, we can obtain a maximum slack for each constraint. The procedure then discards that portion of the interval for each variable for which the maximum slack of some constraint would become negative.

Range Reduction Strategy 2: Denoting the number of nonpositive eigenvalues of the matrix Q by q, this procedure is based on the result that at optimality, at least q out of the $m + 2n$ constraints can be required to hold as equalities (see Phillips and Rosen, 1990, and Mueller, 1970). The proposed strategy computes a measure of redundancy for each functional constraint with respect to the box constraints by maximizing the left-hand side of the less than or equal to type functional constraints. If the number of nonredundant constraints thus detected, plus the number of x-variables that have an *original* lower or upper bound restricting its interval at the current node, is less than q, we can then fathom the current node. If this condition does not hold, we can attempt to improve the bounds on the variables by using the same concept over the feasible range of each variable in turn. For any given variable, we can readily identify in closed-form the range of its interval for which the foregoing criterion would hold, if at all. This range is then discarded unless the remaining interval becomes disjoint, and the procedure is continued with another variable in a cyclic fashion until no further restrictions are affected in a complete cycle.

Range Reduction Strategy 3: For each node, given the current bound restrictions on the x-variables, consider the Lagrangian dual subproblem (8.51) corresponding to the incumbent dual solution of the parent subproblem. Since this gives a valid lower bounding problem, we examine each variable in turn and identify a subinterval of its range for which, if the variable was so restricted, the resulting Lagrangian-based bound would fathom the node. Any such range identified for a variable is discarded, unless the remaining interval becomes disjoint. In a similar fashion, after having solved the current node's problem via its Lagrangian dual, we perform this range reduction before making any branching decision. This restriction also serves as an input for the immediate descendent nodes, for which this strategy would be applied after imposing a branching decision.

Range Reduction Strategy 4: This procedure is performed on the eigen-transformed problem (8.44). We first derive lower and upper bounds on the z-variables, say $[L, U]$, by minimizing and then maximizing for each $i = 1, \ldots, n$, the definition function $z_i = (P^t_{i\cdot})x$, where $(P^t_{i\cdot})$ is the i^{th} row of P^t, over the box constraints on the x-variables corresponding to the current node restrictions. Let z^0 minimize the separable eigen-transformed objective function over $L \leq z \leq U$. Then, considering one z-variable at a time, while fixing the others at their values in z^0, we determine a subinterval to be eliminated for that z-variable by finding the range for which the objective function value would exceed the incumbent solution value. We eliminate such a subinterval, unless the remaining interval becomes disjoint, and continue this procedure in a cyclic fashion until there is no further reduction in the bounding intervals for the z-variables. Let $[L^{new}, U^{new}]$ be the final bounds thus obtained. Then, by including the constraints $L^{new}_i \leq z_i = (P^t_{i\cdot})x \leq U^{new}_i$ for those z_i, $i = 1, \ldots, n$, for which an interval reduction has resulted, we perform Range Reduction Strategy 1 again to possibly further restrict the x-variable bounds.

8.8.5. Branching Variable Selection

We describe below a branching rule that attempts to resolve the discrepancy between the values of the RLT variables and the corresponding nonlinear terms they represent. If there is no such discrepancy, that is, if (8.3) holds at an optimal solution to the bounding problem, then this solution is also optimal to QP. In particular, to identify the RLT variables that contribute toward reducing the lower bound below the true QP objective value, let us compute the quantity

$$d_1 = \underset{1\le k\le \ell\le n}{\text{minimum}}\left\{ \min\{0, \begin{bmatrix} q_{k\ell}, & \text{if } k = \ell \\ 2q_{k\ell}, & \text{if } k < \ell \end{bmatrix} (\overline{w}_{k\ell} - \overline{x}_k\overline{x}_\ell)\}\right\}. \tag{8.53}$$

The next proposition provides a sufficient condition under which the x-variable part of the solution to the RLT bounding problem optimally solves QP.

Proposition 8.10. *Suppose that $(\overline{x}, \overline{w})$ solves the RLT bounding problem of QP, and let $\overline{z}$ be the corresponding objective function value. If $d_1 = 0$ in (8.53), then $\overline{x}$ solves QP with the corresponding objective value being $\overline{z}$.*

Proof. By Proposition 8.2, the constraints of QP are implied in the RLT problem. (Alternatively, if a reduced RLT is being employed, then these constraints are directly present in the RLT problem.) Therefore, $\overline{x}$ is feasible to QP, and its objective function value $c\overline{x} + \overline{x}^t Q\overline{x}$ gives an upper bound on the actual optimum v[QP]. Due to the inner minimization operation in (8.53), $d_1 = 0$ implies that $q_{kk}\overline{w}_{kk} \ge q_{kk}\overline{x}_k^2$ $\forall\ k = 1,\dots,n$, and $q_{k\ell}\overline{w}_{k\ell} \ge q_{k\ell}\overline{x}_k\overline{x}_\ell$ $\forall\ 1 \le k < \ell \le n$. Hence, $\overline{z} \ge c\overline{x} + \overline{x}^t Q\overline{x}$.

But since $\bar{x}$ is feasible to QP, and $\bar{z}$ is a lower bound on QP, we have, $c\bar{x} + \bar{x}^t Q\bar{x} \geq \nu[\text{QP}] \geq \bar{z} \geq c\bar{x} + \bar{x}^t Q\bar{x}$, which implies that equality holds throughout. Hence, $\bar{x}$, of objective value $\bar{z}$, solves QP, and this completes the proof. □

Now, suppose that d_1 given by some indices $(\underline{k}, \underline{\ell})$ in (8.53) is negative. (We break any ties in (8.53) in favor of the maximum discrepancy $\bar{x}_k \bar{x}_\ell - \bar{w}_{k\ell}$, with further ties broken arbitrarily.) In order to choose between $x_{\underline{k}}$ and $x_{\underline{\ell}}$ for a branching variable, using the same motivation as in (8.53), we compute for $t = \underline{k}$ and $\underline{\ell}$,

$$d_2(t) = \sum_{j=1}^{t-1} \min\{0, 2q_{jt}(\bar{w}_{jt} - \bar{x}_j \bar{x}_t)\} + \sum_{j=t+1}^{n} \min\{0, 2q_{tj}(\bar{w}_{tj} - \bar{x}_t \bar{x}_j)\} + \min\{0, q_{tt}(\bar{w}_{tt} - \bar{x}_t^2)\}. \tag{8.54}$$

The branching variable index is then selected as follows:

$$br = \arg\min\{d_2(t):\ t = \underline{k}, \underline{\ell}\}. \tag{8.55}$$

Remark 8.5. The foregoing branching scheme is based on an optimal solution $(\bar{x}, \bar{w})$ to the RLT bounding problem. However, note that since we are solving the Lagrangian dual of the enhanced first level RLT problem, we do not directly obtain the optimal primal (x, w) variable values. Based on Theorem 7.1, we could solve a linear program obtained by surrogating all the constraints of RLT-LP using the optimal dual solution obtained via LD-RLT-NLP, except for the bound-factor product constraints and the original constraints of QP, and guarantee convergence of the algorithm. However, this linear program itself might be a computational bottleneck. Hence, for theoretical purposes, one could resort to solving the latter linear program only finitely often along any branch of the branch-and-bound tree, but for the most pat, resort to the following branching scheme that is motivated by the above theory.

Step 1: Consider the Lagrangian subproblem associated with the dual incumbent solution to LD-RLT-NLP at the stage (8.51), where $[\ell, u]$ are the x-variable bound intervals associated with the *current node.* Letting $f_k(x_k) = \hat{c}_k x_k + \hat{q}_{kk} x_k^2$, select a branching variable index as follows:

$$br = \underset{\substack{k \in \{1,\dots,n\} \\ \ni \ell_k < \bar{x}_k < u_k}}{\arg\max} [\min\{|f_k(u_k) - f_k(\bar{x}_k)|, |f_k(\ell_k) - f_k(\bar{x}_k)|\}] \tag{8.56}$$

where $\bar{x}$ solves (8.51), breaking ties in favor of the variable that has the largest feasible interval at the current node. If all variables are at their bounds in $\bar{x}$, proceed to Step 2.

Step 2: Using the incumbent dual solution to LD-RLT-NLP as the starting solution, continue the conjugate subgradient procedure for 50 more iterations omitting the resetting strategy, and accumulate $(\bar{x}, \bar{w})$ as the average of the Lagrangian subproblem solutions. (See Sherali and Choi (1996) for a discussion on the theoretical convergence to an optimal primal solution to RLT-NLP using such an approach. To aid this process, we also attempt to project $\bar{x}$, if infeasible, onto the feasible region of QP by taking a single step toward this region along a direction defined by the violated constraints.) Select a branching variable using (8.53)-(8.55) provided that $d_1 < 0$ in (8.53), and otherwise, if $d_1 = 0$, then proceed to Step 3.

Step 3: Select the x-variable that has the largest feasible interval in the current node problem as the branching variable. That is, let

$$br = \underset{k \in \{1,\ldots,n\}}{\arg\max} \{(u_k - \ell_k)\} \tag{8.57}$$

and exit this procedure.

Partitioning Phase: If the branching variable x_{br} is selected using Step 1 or Step 2 of the above procedure, then split its current interval at the value $\bar{x}_{br}$, creating the partitions $[\ell_{br}, \bar{x}_{br}]$ and $[\bar{x}_{br}, u_{br}]$, provided that the length of each resulting partition is at least 5% of the length of the current interval $[\ell_{br}, u_{br}]$. Otherwise, partition the current interval of x_{br} by simply bisecting it.

8.8.6. Finding Good Quality Feasible Solutions

In the branch-and-bound algorithm, besides a tight lower bound, we should also actively seek good quality solutions early in the algorithm. Although RLT-(N)LP yields a feasible solution by construction, this may not be true for the case of LD-RLT-NLP. Therefore, we developed and tested the following three heuristic procedures, and composed them in the manner described below.

In the first procedure, we formulated an ℓ_1-norm penalty function for the problem QP that incorporates absolute violations in the functional constraints into the penalty term, and then approximately minimized this penalty function using the conjugate subgradient algorithm described in Remark 8.4. As the starting solution, we used the average of the x-variable part of the subproblem solutions obtained at improving iterations of the conjugate subgradient algorithm while solving the Lagrangian dual problem at the current branch-and-bound node. During this procedure, we attempted to project promising near feasible points onto the original feasible region of QP by taking a single step toward this region along a direction defined by the violated constraints. We considered a feasibility tolerance of 10^{-6} in Euclidean distance to be compatible with the default setting for MINOS (see Murtagh and Saunders, 1987).

In the second procedure, we simply applied MINOS to the original problem QP, using the resulting point of the foregoing penalty approach as the starting solution.

In the third procedure, at each node, having solved the Lagrangian dual problem, we formulated (8.49) corresponding to the incumbent dual solution, but now, we also included the functional constraints of QP in this subproblem. The resulting convex program was then solved using MINOS. Notice that this procedure also happens to be a dual ascent step for the Lagrangian dual problem.

In the overall branch-and-bound algorithm, we implemented the following heuristic scheme. For the first 10 nodes of the branch-and-bound tree, we employed the second procedure, using the solution of the first procedure as a starting solution. At all subsequent nodes, we employed the first and the third procedures, except that whenever the incumbent solution improved, we executed the second procedure using this new incumbent point as the starting solution, to possibly further improve this incumbent solution.

8.8.7. Summary of the Algorithm

Step 0. Initialization: Apply the heuristic procedure of Section 8.8.6 using $x^0 = Pz^0$ as the starting solution, where z^0 is defined as in Strategy 4 of Section 8.8.4, to obtain an initial incumbent solution. Initialize the branch-and-bound tree as node 0, and let the present set of bounds on the x-variables be as given in Problem QP. Flag node 0 and proceed to Step 1.

Step 1. Range Reductions: Designate the most recently flagged node as the current active node. For the given set of bounds, apply the Range Reduction Strategies 1-4. If this indicates a fathoming of the current node, then go to Step 3. Otherwise, proceed to Step 2.

Step 2. Bounding and Branching Step: Scale the node subproblem using the scheme of Section 8.8.1, and solve the problem LD-RLT-NLP to obtain a lower bound on the node subproblem. (If LD-RLT-NLP(SC) is used, then prior to the constraint selection process, scale the problem (8.44) as described in Section 8.8.1 using the bounds $[L^{new}, U^{new}]$ on the z-variables, which are readily available as a byproduct of the Range Reduction Strategy 4.) During the process of solving the Lagrangian dual problem, whenever the incumbent dual solution improves during the conjugate subgradient optimization iterations, check if the fathoming condition (8.52) is satisfied, and proceed to Step 3 if this is the case. Apply the heuristic procedure of Section 8.8.6, to possibly improve the incumbent solution. Again, if the fathoming rule (8.52) is satisfied, then proceed to Step 3. Otherwise, apply the Range Reduction Strategy 3 using the current incumbent dual solution, and select a branching variable according to the branching rule of Section 8.8.5. Accordingly, partition the current node subproblem by creating two nonactive descendent nodes corresponding to the resulting two sets of (revised) bounds on the branching variable x_{br}, and go to Step 4.

Step 3. Fathoming Step: Fathom the current node. If the sibling of the fathomed node is not active (see Section 8.8.2) then flag that node. Otherwise, flag the nonactive sibling of the highest level node on the path from the current node to the root node, and return to Step 1. If there is no such node, then either stop if there exist no active end nodes, or else, proceed to Step 4.

Step 4. Node Selection Step: If the incumbent solution has improved since the last time Step 4 has been visited, then fathom any active node that satisfies the criterion (8.52). If the number of active end nodes equals $MAXACT$, then select an active end node that has the least lower bound, and flag one of its descendent nodes. On the other hand, if the number of active end nodes is less than $MAXACT$, then along the branch of each such end node, find the lowest level node (closest to the root) that has at least one nonactive descendent node, and among these nodes, flag the nonactive descendant node of the one that has the least lower bound. Return to Step 1.

The convergence of the above algorithm follows that for the more general procedure of Chapter 7 (see Theorem 7.1). Hence, any accumulation point of the sequence of solutions generated for the RLT relaxations along any infinite branch solves the Problem QP, and finite convergence to an ε-optimal solution can therefore be obtained.

8.9. Computational Results

We now evaluate the proposed algorithm using a set of test problems chosen from the literature, as well as on some randomly generated problems. In addition to the five problems used in Section 8.7, six larger sized $(m = 10, n = 20)$ standard test problems from Floudas and Pardalos (1990), and seven randomly generated problems using the generation scheme of Phillips and Rosen (1990) and Visweswaran and Floudas (1993) of size up to $(m = 20, n = 50)$ are solved.

Tables 8.2 and 8.3 present results on the standard test problems using LD-RLT-NLP(SC) and LD-RLT-NLP, respectively, as the lower bounding problem. The optimality tolerance (see Section 8.8.3) is taken as 1% for the first five problems and as 5% for the remaining ones. A particularly noteworthy feature of all the variants implemented is that very few nodes need to enumerated by the branch-and-bound algorithm due to the tightness of the lowering bounding RLT relaxation. Often, a single relaxation suffices. Sherali and Alameddine (1992) provide computations on using RLT-LP on six other standard bilinear programs from the literature, as well as on eleven separably constrained and thirteen jointly constrained random bilinear programs of sizes up to seven variables in

Table 8.2. Performance of the branch-and-bound algorithm using LD-RLT-NLP(SC).

Problem	(m, n)	Known v[QP]	v[B&B]	cpu secs.	No. of B&B nodes	Node 0 LB
BLP1	(2, 2)	–1.083	–1.083	0.71	1	–1.089
BLP2	(10, 10)	–45.38	–45.38	1.08	1	–45.81
BLP3	(13, 10)	–794.86	–794.86	3.02	5	–838.85
CQP1	(11, 10)	–267.95	–268.01	1.17	1	–270.69
CQP2	(5, 10)	–39.00	–39.00	1.72	5	–42.95
CQP3	(10, 20)	–394.75	–394.75	3.29	3	–423.32
CQP4	(10, 20)	–884.75	–884.75	2.61	1	–904.02
CQP5	(10, 20)	–8695.01	–8695.01	2.55	1	–9097.99
CQP6	(10, 20)	–754.75	–754.75	2.61	1	–787.60
CQP7	(10, 20)	–4105.28	–4150.41	15.94	11	–5126.68
IQP1	(10, 20)	49318.0	49317.97	2.73	3	44937.40

Table 8.3. Performance of the branch-and-bound algorithm using LD-RLT-NLP.

Problem	(m, n)	Known v[QP]	v[B&B]	cpu secs.	No. of B&B nodes	Node 0 LB
BLP1	(2, 2)	–1.083	–1.083	0.71	1	–1.089
BLP2	(10, 10)	–45.38	–45.38	1.37	3	–46.02
BLP3	(13, 10)	–794.86	–794.86	2.66	5	–839.02
CQP1	(11, 10)	–267.95	–268.01	1.12	1	–270.68
CQP2	(5, 10)	–39.00	–39.00	1.61	5	–42.96
CQP3	(10, 20)	–394.75	–394.75	8.13	7	–439.03
CQP4	(10, 20)	–884.75	–884.75	2.54	1	–928.92
CQP5	(10, 20)	–8695.01	–8695.01	13.26	11	–9541.67
CQP6	(10, 20)	–754.75	–754.75	5.04	5	–803.31
CQP7	(10, 20)	–4105.28	–4150.41	27.00	25	–5262.57
IQP1	(10, 20)	49318.0	49317.97	2.61	3	45776.43

Legend: Problem = problem name, (m, n) = size of the problem QP, v[QP] = known optimal (best) solution of problem QP, v[B&B] = branch-and-bound algorithm incumbent value, cpu secs. = cpu seconds to solve the problem on an IBM 3090 computer, No. of B&B nodes = number of branch-and-bound nodes generated, Node 0 LB = lower bound on QP at root node (optimality criterion is 1% for the first 5 problems, and 5% for the rest of the problems).

each set and up to twelve constraints. All the standard test problems, all the separably constrained random problems and all but two jointly constrained problems were solved at the initial node itself via a *single linear program.* In fact, the following result *theoretically* supports this computational behavior for the separably constrained problems in particular. (Note that for *separably constrained* problems that are constrained as $x \in X$ and $y \in Y$, only pairwise cross-product RLT constraints are necessary, where each product considers one constraint from each of the sets X and Y.) Hence, the proposed RLT scheme affords a powerful mechanism for recovering optimal or near optimal solutions to such nonconvex quadratic problems via *single* convex or linear relaxations.

Proposition 8.11. *Let $X \subseteq R^n$ and $Y \subseteq R^m$ be two nonempty polytopes in variables x and y, respectively, and consider the set of RLT constraints generated by taking a pairwise product of one constraint from each set. Let $z \equiv (x, y, w) \in Z$ be the resultant constraint set thus generated, where $w_{ij} = x_i y_j$ $\forall\ i, j$ has been substituted in the linearization step, similar to (8.3). Then, for any feasible solution $(\bar{x}, \bar{y}, \bar{w}) \in Z$, if either $\bar{x} \in vert(X)$ or $\bar{y} \in vert(Y)$, we necessarily have $\bar{w}_{ij} \equiv \bar{x}_i \bar{y}_j$ $\forall\ i, j$, where $vert(\cdot)$ denotes the set of vertices (extreme points) of $(\cdot)$.*

Proof. Let $(\bar{x}, \bar{y}, \bar{w}) \in Z$, and suppose that $\bar{x} \in vert(X)$. (The case of $\bar{y} \in vert(Y)$ is symmetric.) Hence, there exist a set of linearly independent hyperplanes defining X, say $Hx = h$ that are binding at $\bar{x}$, yielding $\bar{x} \equiv H^{-1}h$. Since Y is bounded, by

Proposition 8.3, the RLT constraints imply that $[(Hx - h)y_j]_L = 0 \quad \forall\ j = 1, \ldots, m$ holds true at $(\bar{x}, \bar{y}, \bar{w})$, *i.e.*, $H\bar{w}_{\cdot j} = h\bar{y}_j \quad \forall\ j = 1, \ldots m$, where $w_{\cdot j} \equiv (w_{ij},\ i = 1, \ldots, n)$. Hence, $\bar{w}_{\cdot j} = H^{-1}h\bar{y}_j = \bar{x}\ \bar{y}_j \quad \forall\ j = 1, \ldots, n$, and this completes the proof. $\square$

By Proposition 8.11, we have that for a bilinear programming problem

BLP: minimize $\{c^t x + d^t y + x^t Cy\colon\ \ x \in X,\ y \in Y\}$,

where X and Y are as defined in the proposition, if the RLT relaxation yields a solution $(\bar{x}, \bar{y}, \bar{w})$ where either $\bar{x} \in vert(X)$ or $\bar{y} \in vert(Y)$, then this LP relaxation solution would be optimal for BLP. Because of the separable constraint structure, this event is quite likely (though not necessary), and hence the observed computational results.

The foregoing comments readily generalize to the case of $r \geq 2$ polytopes, where the nonlinear terms in the objective function involve only cross-products of degree less than or equal to r among variables from different sets. Suppose that the RLT constraints are generated by constructing all possible r-products composed by selecting one constraint factor from each defining set. Then, if the resulting RLT-LP yields a solution that corresponds to an extreme point of at least $(r - 1)$ sets, the RLT variables would then match with the corresponding nonlinear products, hence solving the underlying nonconvex multilinear problem.

Using this same argument in the context of QP, we have the following result.

Proposition 8.12. *Consider Problem QP and suppose that we generate the relaxation RLT-LP. If $(\bar{x}, \bar{w})$ solves RLT-LP and we have that $\bar{x}$ is an extreme point for QP, then $\bar{x}$ solves QP.*

Proof. The proof is similar to that of Proposition 8.11. □

To summarize, we observe that particular strength can be derived via RLT when the underlying nonconvex problem has a strong propensity toward extreme point optimality.

Continuing with our computational analysis, we see that by using LD-RLT-NLP(SC), all the problems are solved under 16 cpu seconds. Compared to Table 8.3, although a more relaxed problem is being solved in Table 8.2, we observe an improvement in several of the root node lower bounds, especially for larger sized problems. This is principally due to the reduction of the dual search space, which improved the performance of the conjugate subgradient algorithm, while not significantly sacrificing the tightness of the theoretical lower bound. In Table 8.3, the run times have somewhat increased for the larger problems, where all the problems, except one, are solved under 14 cpu seconds. Problem CQP7 required by far the greatest effort, taking 27 cpu seconds to be solved. However, even when using an optimality criterion of 5%, a better solution than the best known one to CQP7 is found. (The previous best solution reported in Floudas and Pardalos (1990) has an objective value of –4105.2779.) Upon reducing the optimality criterion to 1% and then to 0.1%, CQP7 is solved in 64 cpu seconds and in 205 cpu seconds, respectively, and in both cases, the same solution of value –4150.4087 is obtained (the non-zero variables in this solution have values $x_3 = 1.0429$, $x_{11} = 1.746744$, $x_{13} = 0.4314709$, $x_{16} = 4.43305$, $x_{18} = 15.85893$, $x_{20} = 16.4869$). For both of these cases, 86% of the overall effort is spent in solving the Lagrangian dual problem, showing that this is the determining factor for the total computational effort required. Using the reduced problem LD-RLT-NLP(SC), when we set the optimality criterion to 1% for CQP7, the algorithm consumed 90 cpu seconds, while for an accuracy tolerance of 0.1%, the algorithm was prematurely terminated after enumerating the preset limit of 200 nodes in 290 cpu seconds. Therefore, if a higher degree of accuracy is required, we recommend using the non-reduced problem LD-RLT-NLP; where overall, the marginally tighter representation does play an important role.

The heuristic procedure of Section 8.8.6 performed well by identifying the incumbent solution at the root node for all the problems, except for Problem CQP2, for which the optimum was found at the second node. The range reduction strategies of Section 8.8.4 prove to be very fast and effective; for example, if these reductions are not performed for Problem CQP7, the number of branch-and-bound nodes enumerated increases from 25 to 45 (requiring 57 cpu seconds) in Table 8.3.

Although by Proposition 8.2, the tightness of the implied bounds on the variables should not affect the result of the bounding problems, this seems to play an important role in the performance of the Lagrangian dual solution procedure. As originally stated, problems CQP3-CQP7 do not include upper bounds on the variables, and for the purpose of the branch-and-bound algorithm, we used the smallest hyperrectangle that contains the

Table 8.4. Performance of the branch-and-bound algorithm using LD-RLT-NLP.

Problem (m, n)		Known v[QP]	v[B&B]	cpu secs.	No. of B&B nodes	Node 0 LB
BLP1	(2, 2)	–1.083	–1.083	1.15	1	–1.089
BLP2	(10, 10	–45.38	–45.38	12.90	1	–45.38
BLP3	(13, 10)	–794.86	–794.86	63.60	1	–794.86
CQP1	(11, 10)	–267.95	–268.01	15.25	1	–268.01
CQP2	(5, 10)	–39.00	–39.00	16.68	3	–39.82

Legend: Problem = problem name, (m, n) = size of the problem QP, v[QP] = known optimal (best) solution of problem QP, v[B&B] = branch-and-bound algorithm incumbent value, cpu secs. = cpu seconds to solve the problem on an IBM 3090 computer, No. of B&B nodes = number of branch-and-bound nodes generated, Node 0 LB = lower bound on QP at root node (optimality criterion is 1%).

feasible region found by minimizing and maximizing each variable over the feasible region. As a comparison, upon using a looser upper bound of 40 on each variable, which is trivially implied by a generalized upper bounding type of constraint present in these problems, CQP3 and CQP7 are solved in 35 and 43 cpu seconds, respectively, usingLD-RLT-NLP(SC) as the bounding problem. (See the discussion in the next chapter on bound tightening strategies.) Table 8.4 presents results for the smaller sized problems using RLT-NLP as the bounding problem, and solving this bounding relaxation by using MINOS 5.1, in lieu of the Lagrangian dual approach used in Table 8.3. We observe that although the computational time has increased for all problems, the first four problems are solved at the root node itself, while the fifth problem returns an initial lower bound of –39.82 at the root node, the optimum value being –39.00. Note that while there is some loss in the tightness of the bounds due to the inaccuracy in solving RLT-NLP via the Lagrangian dual approach, the overall gain in efficiency is quite

Table 8.5. Performance of the branch-and-bound algorithm for randomly generated problems using LD-RLT-NLP(SC).

(m, n)	v[B&B]	Node 0 LB	cpu seconds 5% optimality	Full Node 0	No. of B&B nodes	Node 0 relative gap 100(UB-LB)/|UB|
(20, 25)	365.91	365.64	1.71	6.06	1	0.076%
(20, 25)	1170.54	1169.98	1.52	5.88	1	0.048%
(20, 40)	–234.55	–234.95	3.58	12.91	1	0.172%
(20, 40)	–1264.53	–1264.93	2.30	12.44	1	0.031%
(20, 50)	–1311.48	–1326.01	3.84	17.21	1	1.109%
(20, 50)	–1259.01	–1319.69	12.77	18.50	1	4.82%
(20, 50)	–1215.62	–1241.04	4.98	18.08	1	2.091%

significant. Hence, there exists a great potential for further improvement if the lower bounding problem RLT-NLP could be solved more accurately by the Lagrangian dual scheme. Using the random problem generator of Visweswaran and Floudas (1993) and Phillips and Rosen (1990), we solved seven larger sized problems having up to 20 constraints and 50 variables using the bounding problem LD-RLT-NLP(SC), and an optimality criterion of 5%. These problems are of the form $\min\{\theta_1(0.5\sum_{i=1}^{n}\lambda_i(x_i - \overline{w}_i)^2)\colon\ Ax \le b,\ x \ge 0\}$, where $\theta_1 = -0.001$, and the number of positive and negative components of λ are roughly equal. As reported in Table 8.5, all the problems are solved at the root node with a reasonable computational effort. Note that for several of these problems, the 5% optimality tolerance was detected even before the node zero analysis was completed (see the cpu seconds columns). However, the results given in Table 8.5 correspond to a full node zero analysis. The final column in Table 8.5 shows that the proven accuracy of the solutions obtained at node zero is typically significantly better than 5%. All incumbent solutions obtained were subsequently verified to be at least within 1% of optimality, except for the sixth problem, for which a better incumbent solution of value −1281.0 was obtained when we enumerated two more nodes. This shows that the actual accuracy of the node zero lower bound for this problem is at least 3%.

To summarize, in this chapter, we have investigated Reformulation-Convexification based relaxations embedded within a branch-and-bound algorithm for solving nonconvex quadratic programming problems. Tight nonlinear programming relaxations have been constructed, and a suitable Lagrangian dual procedure has been designed to solve these relaxations efficiently. The proposed algorithm has been further enhanced by incorporating fast and effective range reduction procedures. For implementation, we recommend the use of LD-RLT-NLP when a better than 5% accuracy is desired, and the use of the reduced relaxation LD-RLT-NLP(SC) otherwise. For specially structured QPs, especially in the light of Remark 8.1 and Proposition 8.2, we strongly suggest that specialized, reduced RLT relaxations be investigated. The eigen-space based relaxation LD-RLT-NLPE is also recommended to be used whenever it can be conveniently constructed, and if there is a significant gap observed between the lower bounds generated via RLT-NLP and RLT-NLPE. For large sized problems, we recommend that the heuristic of Section 8.8.6 be used, perhaps in concert with solving the RLT based relaxation LD-RLT-NLP(SC) at a limited number of nodes.

9

REFORMULATION-CONVEXIFICATION TECHNIQUE FOR POLYNOMIAL PROGRAMS: DESIGN AND IMPLEMENTATION

In Chapter 7, we discussed the design and theory of a Reformulation-Linearization Technique for solving polynomial programming problems. We now return to this class of problems, but this time, we shall focus on specific implementation issues and describe various algorithmic strategies and additional classes of RLT constraints that can be gainfully employed in enhancing the performance of an RLT-based solution procedure. For the sake of convenience, we restate below the type of polynomial programming problems considered here, as introduced in Chapter 7.

$$\textbf{PP}: \text{ Minimize } \{\phi_0(x): \; x \in Z \cap \Omega\} \tag{9.1}$$

where $Z = \{x: \; \phi_r(x) \geq \beta_r$ for $r = 1, \ldots, R_1$, and $\phi_r(x) = \beta_r$ for $r = R_1 + 1, \ldots, R\}$, $\Omega = \{x \in R^n: \; 0 \leq \ell_j \leq x_j \leq u_j < \infty$ for $j = 1, \ldots, n\}$, and where

$$\phi_r(x) = \sum_{t \in T_r} \alpha_{rt} \left[\prod_{j \in J_{rt}} x_j \right] \text{ for } r = 0, 1, \ldots, R. \tag{9.2}$$

Note that we shall explicitly treat the case of polynomial programs having integral exponents on variables in this chapter; extensions to the case of rational exponents can follow directly based on the discussion in Chapter 7. As before, we shall denote $N = \{1, \ldots, n\}$, define δ as the maximum degree of any polynomial term that appears in Problem PP, and let $\overline{N}$ contain δ replicates of each element of N. Then, each $J_{rt} \subseteq \overline{N}$, with $1 \leq |J_{rt}| \leq \delta$ for each $t \in T_r$, $r = 0, 1, \ldots, R$.

In the sequel, we shall specifically address both univariate and multivariate versions of Problem PP. Note that in the former case, the problem is typically reduced to that of minimizing a univariate polynomial function over simply a compact interval (or a collection of such intervals). Sherali and Tuncbilek (1996) discuss some related literature, and our presentation here will be based largely on this paper. In particular, one class of methods for multivariate problems of the type PP is based on a successive-quadrification process introduced by Shor (1990), in which the problem is first equivalently transformed into a quadratic polynomial program, and then a suitable algorithm is used to solve this problem. Floudas and Visweswaran (1990, 1995) use such a process and then develop an extended version of the generalized Benders algorithm for the resulting problem, that is designed to address the inherent nonconvexity of the problem. A specialization of this technique to univariate problems is presented in Visweswaran and Floudas (1992). However, Hansen *et al.* (1993) show that this is equivalent to a particular application of interval arithmetic methods (see Hansen (1992), for example) for solving such problems. Al-Khayyal *et al.* (1994) also employ this quadrification process, further transforming the problem into an equivalent bilinearly constrained bilinear program, for which convex envelope based linear programming relaxations are generated and embedded within a branch-and-bound algorithm. In contrast, we have introduced in Chapter 7 an RLT procedure that can be applied directly to Problem PP itself in order to generate linear programming relaxations, and have designed globally convergent branch-and-bound algorithms using these relaxations.

We begin our discussion in this chapter by first showing in Section 9.1 that the process of applying RLT directly to PP generates relaxations that theoretically dominate the relaxations that would be obtained by applying RLT to the entire family of alternative, equivalent quadrified polynomial programs, even if they are *simultaneously* considered within a single problem. Moreover, the latter family of relaxations in turn strictly dominate the relaxation of Al-Khayyal *et al.* (1994). Section 9.2 then provides some related empirical evidence of this dominance. We next focus on the application of RLT to Problem PP, and present a significant enhancement of this procedure via the generation of new classes of constraints and via various implementation strategies such as a constraint filtering technique, a range-reduction process, and a new branching variable selection procedure. Since the new classes of constraints introduced include certain convex variable bounding types of restrictions, we sometimes refer to this as a *Reformulation-Convexification Approach.* However, aside from these simple convex constraints, since the relaxation is otherwise predominantly linear, we will continue to use "RLT" to describe this process of generating relaxations. For the univariate case, we present in Section 9.3 such convex bounding constraints, as well as new squared grid-factor based and squared Lagrangian interpolation polynomial based (linear) RLT constraints. We show empirically in Section 9.4 that these relaxations yield very tight lower bounds, recovering the global optimum for 4 out of 7 test problems from the literature via the initial relaxation itself, the maximum deviation of this lower bound from the global optimum being at most 3% for the other instances (barring zero optimal values). In contrast, the exponential transformation approach typically used in geometric programming produces lower bounds as far as 896, 726, 900% from optimality! Next, in Section 9.5, we present relaxations for multivariate problems, along with the aforementioned new classes of constraints and related algorithmic strategies. A branch-and-bound algorithm is proposed, which is numerically tested in Section 9.6 on a set of standard chemical process, pooling, and engineering design test problems from the literature. The chapter concludes with suggestions for further extensions and enhancements.

9.1. Application of RLT to a Class of Quadrified Versus the Original Polynomial Program

In the previous chapter, we discussed specialized algorithmic strategies for solving nonconvex quadratic programs. Although we assumed that the constraints are linear, the proposed methodology largely extends to solving *quadratic polynomial programs*, i.e., *quadratically constrained quadratic programs* as well. We begin this section by showing how a polynomial program PP of the type (9.1) can be converted to an equivalent quadratic polynomial program. As we shall see, this transformation is not unique. However, we shall prove that by applying RLT directly to the original polynomial program, we can derive a bound that *dominates* (in the sense of being at least as tight as) the value obtained when RLT is applied to an *all encompassing* quadratic problem that is constructed by *simultaneously* or *jointly* using all possible quadrified representations of the original problem. Some related numerical comparisons will then be presented in Section 9.2 to show that this dominance can be strict, and sometimes quite significant.

9.1.1. Quadrification Process and the Application of RLT to the Quadrified Polynomial Program

First, let us present the basic transformation used for *quadrifying* Problem PP, i.e., for converting it into an equivalent quadratic polynomial program. Let the highest degree of each variable x_j appearing in PP be S_j, $j = 1,\ldots,n$, and define $s_j = \lceil S_j/2 \rceil$, where $\lceil \cdot \rceil$ denotes the rounding-up operation. Consider the set

$$A = \{a = (a_1,\ldots,a_n) \in Z_+^n:\ 0 \le a_j \le s_j\ \forall\ j = 1,\ldots,n, \text{ and } \sum_{j=1}^{n} a_j \le \delta\} \tag{9.3}$$

where Z_+^n is the set of nonnegative integral n-vectors. For each $a \in A$, define a **variable** $R[a]$ to represent the **monomial** $x^a \equiv \prod_{j=1}^{n} x_j^{a_j}$. Our equivalent quadrified program will be defined in terms of the variables $R[a]$, which motivates (9.3) along with the definition of s_j for $j = 1,\ldots,n$.

To illustrate the quadrification process, consider the polynomial term $x_1^3 x_2^2 x_3^4$. This can be reduced to a quadratic form by including the following series of identity relations involving the $R[\cdot]$ variables,

$$x_1 \equiv R[1,0,0],\ x_2 \equiv R[0,1,0],\ x_3 \equiv R[0,0,1],$$

$$x_3^2 \equiv R[0,0,2] = R[0,0,1]R[0,0,1]$$

$$x_2 x_3^2 \equiv R[0,1,2] = R[0,0,2]R[0,1,0],$$

$$x_1 x_2 x_3^2 \equiv R[1,1,2] = R[0,1,2]R[1,0,0]$$

$$x_1^2 x_2 x_3^2 \equiv R[2,1,2] = R[1,1,2]R[1,0,0],$$

and then, replacing $x_1^3 x_2^2 x_3^4$ by the quadratic term $R[1,1,2]R[2,1,2]$ in the $R[\cdot]$ variables.

Hence, by defining suitable variables $R[a]$, $a \in A$, and by including appropriate identities involving these variables as done above, we can thereby quadrify Problem PP. Notice that this successive quadrification scheme is not uniquely defined; for example, instead of writing $x_2 x_3^2$ as $(x_2)(x_3^2)$ as essentially done above, we could have as well composed $x_2 x_3^2 = (x_3)(x_2 x_3)$ as follows:

$$x_2 x_3 \equiv R[0,1,1] = R[0,1,0]R[0,0,1],$$

$$x_2 x_3^2 \equiv R[0,1,2] = R[0,1,1]R[0,0,1].$$

In order to simultaneously capture all such transformations or quadrifying identities within an *all-encompassing* equivalent quadratic polynomial program, and then to apply RLT to this quadratic program, we adopt the following stepwise scheme.

Step 1 (Variable Definition and Quadrification of Objective and Constraint Functions). For each $a \in A$ defined in (9.3), associate a variable $R[a]$ as above, and restrict

$$\prod_{j=1}^{n} \ell_j^{a_j} \leq R[a] \leq \prod_{j=1}^{n} u_j^{a_j} \quad \forall \text{ such variables } R[a]. \tag{9.4}$$

Replace each polynomial term $\phi_r(x)$ in PP by some quadratic expression of the form

$$\phi_r(x) \leftarrow \sum_{t \in T_r} \alpha_{rt} R[a^{t_1}] R[a^{t_2}], \text{ where } \{a^{t_1}, a^{t_2}\} \subseteq A, \text{ and where}$$

$$x^{(a^{t_1}+a^{t_2})} \equiv \prod_{j \in J_{rt}} x_j \quad \forall\ t \in T_r,\ r = 0,1,\ldots,R. \tag{9.5}$$

Step 2 (Quadrification Constraints). To establish the required inter-relationships among the $R[\cdot]$ variables, include all possible quadratic identities of the following form, where $R[0,\ldots,0] \equiv 1$.

$$R[a^1]R[a^2] = R[a^3]R[a^4] \quad \forall \quad \{a^1, a^2, a^3, a^4\} \subseteq A \text{ such that}$$

$$a^1 + a^2 = a^3 + a^4 \leq S \equiv (S_1, \ldots, S_n), \text{ with}$$

$$\sum_{j=1}^{n} (a_j^1 + a_j^2) = \sum_{j=1}^{n} (a_j^3 + a_j^4) \leq \delta. \tag{9.6}$$

Step 3 (Reformulation Phase of RLT). Construct all possible pairwise products (including self-products) of the bound-factors defined in (9.4), so long as for any quadratic term $R[a^1]R[a^2]$ thus produced, we preserve the maximum degree δ, i.e., $\sum_{j=1}^{n}(a_j^1 + a_j^2) \leq \delta$.

Step 4 (Linearization Phase of RLT). Substitute a variable for each distinct product of the type $R[a^1]R[a^2]$. However, noting the relationship (9.6), we can substitute in this phase a variable $W[a^1 + a^2]$ for each distinct product of the type $R[a^1]R[a^2]$, so that if $a^1 + a^2 = a^3 + a^4$, the same linearizing variable would be substituted in place of $R[a^1]R[a^2]$ as would be for $R[a^3]R[a^4]$. This would hence render (9.6) redundant.

The net effect of this process is that at Step 1, each distinct polynomial term of the form $\pi_{j \in J} x_j$, for some $J \subseteq \overline{N}$, in the objective function and in the constraints of Problem PP is replaced by a single variable $W[a]$, where a_j is the number of times the index j appears in J. Note that by (7.3) (see 9.8 below), this variable $W[a]$ is precisely the variable X_J used in **LP(PP)**, the linear programming relaxation that is obtained by applying RLT directly to Problem PP. This latter problem is stated explicitly below for convenience.

LP(PP): Minimize $[\phi_0(x)]_L$ (9.7a)

subject to $[\phi_r(x)]_L \geq \beta_r \ \forall r = 1, \ldots, R_1$,

$[\phi_r(x)]_L = \beta_r \ \forall r = R_1 + 1, \ldots, R$ (9.7b)

$$\left[\underset{j \in J_1}{\pi} (x_j - \ell_j) \underset{j \in J_2}{\pi} (u_j - x_j) \right]_L \geq 0$$

$$\forall \, (J_1 \cup J_2) \subseteq \bar{N}, \; |J_1 \cup J_2| = \delta \tag{9.7c}$$

where $[(\cdot)]_L$ denotes the linearized form of the polynomial function $(\cdot)$ that is obtained upon substituting a single variable for each distinct polynomial term according to

$$X_J = \underset{j \in J}{\pi} x_j \quad \forall \, J \subseteq \bar{N}. \tag{9.8}$$

Hence, the linearization phase applied to the quadrified objective function and constraints of PP would result in producing precisely (9.7a) and (9.7b), respectively. In addition, using the same linearization substitution as in (9.8), the bound-factor product constraints of Step 3, along with the constraints (9.4), would produce the following restrictions.

$$[(\underset{j=1}{\overset{n}{\pi}} x_j^{a_j^1} - \underset{j=1}{\overset{n}{\pi}} \ell_j^{a_j^1})(\underset{j=1}{\overset{n}{\pi}} x_j^{a_j^2} - \underset{j=1}{\overset{n}{\pi}} \ell_j^{a_j^2})]_L \geq 0 \text{ and}$$

$$[(\underset{j=1}{\overset{n}{\pi}} u_j^{a_j^1} - \underset{j=1}{\overset{n}{\pi}} x_j^{a_j^1})(\underset{j=1}{\overset{n}{\pi}} u_j^{a_j^2} - \underset{j=1}{\overset{n}{\pi}} x_j^{a_j^2})]_L \geq 0$$

for all distinct *unordered* pairs $\{a^1, a^2\} \subseteq A$ such that $\sum_{j=1}^{n} (a_j^1 + a_j^2) \leq \delta$, (9.9a)

$$[(\underset{j=1}{\overset{n}{\pi}} x_j^{a_j^1} - \underset{j=1}{\overset{n}{\pi}} \ell_j^{a_j^1})(\underset{j=1}{\overset{n}{\pi}} u_j^{a_j^2} - \underset{j=1}{\overset{n}{\pi}} x_j^{a_j^2})]_L \geq 0$$

for all distinct *ordered* pairs $\{a^1, a^2\} \subseteq A$ such that $\sum_{j=1}^{n} (a_j^1 + a_j^2) \leq \delta$, (9.9b)

and

$$\underset{j=1}{\overset{n}{\pi}} \ell_j^{a_j} \leq \left[\underset{j=1}{\overset{n}{\pi}} x_j^{a_j} \right]_L \leq \underset{j=1}{\overset{n}{\pi}} u_j^{a_j} \ \forall\ a \in A. \tag{9.9c}$$

Hence, the *all-encompassing* linear programming RLT relaxation that would result from simultaneously considering all possible quadrification transformations and then applying RLT to this as above, is given as follows.

LP($\overline{\textbf{QPP}}$): Minimize $\{[\phi_0(x)]_L$: Constraints (9.7b) and (9.9)$\}$. (9.10)

Remark 9.1. Note that we could have considered all possible bound-factor products composed from (9.4) at Step 3, without the restriction that $\sum_{j=1}^{n} (a_j^1 + a_j^2) \leq \delta$ at the possible expense of increasing the degree of the polynomial terms in (9.9) to $\delta' > \delta$. If this is done, then the same dominance result below holds true by correspondingly also increasing δ to δ' in (9.7c) for LP(PP). □

9.1.2. Dominance of LP(PP) over LP($\overline{\text{QPP}}$)

In the following discussion, we show that $\nu[\text{LP(PP)}] \geq \nu[\text{LP}(\overline{\text{QPP}})]$, where $\nu[\cdot]$ denotes the optimal value of the given problem $[\cdot]$. In order to prove this dominance result, we first show that the defined terms of the type $(\underset{j=1}{\overset{n}{\pi}} x_j^{a_j} - \underset{j=1}{\overset{n}{\pi}} \ell_j^{a_j})$ and $(\underset{j=1}{\overset{n}{\pi}} u_j^{a_j} - \underset{j=1}{\overset{n}{\pi}} x_j^{a_j})$ as well as their products, can each be expressed as a sum of nonnegative multiples of ordinary bound-factor and nonnegative variable products. Hence, a linearization of products of such compound factors can likewise be expressed as a sum of linearizations of the latter type of ordinary nonnegative bound-factor and variable products. Then, by showing that such latter products are themselves implied by the RLT constraints defining LP(PP), we will establish the dominance results.

Proposition 9.1. *The terms* $\pi_{j=1}^{n}(x_j^{p_j} - \ell_j^{p_j})$ *and* $\pi_{j=1}^{n}(u_j^{p_j} - x_j^{p_j})$, *where* $p \in Z_+^n$, $\sum_{j=1} p_j \leq \delta$, *as well as the term* $\pi_{j=1}^{n}(x_j^{p_j} - \ell_j^{p_j})\ \pi_{j=1}^{n}(u_j^{q_j} - x_j^{q_j})$, *where p and* $q \in Z_+^n$, $\sum_{j=1}(p_j + q_j) \leq \delta$, *can each be written as a sum of nonnegative multiples of terms of the type*

$$\pi_{j\in J_1}(u_j - x_j)\ \pi_{j\in J_2}(x_j - \ell_j)\ \pi_{j\in J_3} x_j \text{ where } (J_1 \cup J_2 \cup J_3) \subseteq \overline{N},$$

$$|J_1 \cup J_2 \cup J_3| \leq \delta. \tag{9.11}$$

Proof. By the binomial expansion, we know that,

$$(y + a)^r = \binom{r}{0} y^r a^0 + \binom{r}{1} y^{r-1} a^1 + \ldots + \binom{r}{r-1} y^1 a^{r-1} + \binom{r}{r} y^0 a^r. \tag{9.12}$$

Hence, putting $a = \ell_j$, $y = (x_j - \ell_j)$, and $r = p_j$, for $j \in N$, in (9.12), we get

$$(x_j^{p_j} - \ell_j^{p_j}) = (x_j - \ell_j)^{p_j} + p_j \ell_j (x_j - \ell_j)^{p_j - 1}$$

$$+ \frac{p_j(p_j - 1)}{2} \ell_j^2 (x_j - \ell_j)^{p_j - 2} + \ldots + p_j \ell_j^{p_j - 1}(x_j - \ell_j). \tag{9.13}$$

Similarly, putting $a = x_j$, $y = (u_j - x_j)$, $r = p_j$, for $j \in N$ in (9.12), we get

$$(u_j^{p_j} - x_j^{p_j}) = (u_j - x_j)^{p_j} + p_j x_j (u_j - x_j)^{p_j - 1}$$

$$+ \frac{p_j(p_j - 1)}{2} x_j^2 (u_j - x_j)^{p_j - 2} + \ldots + p_j x_j^{p_j - 1}(u_j - x_j). \tag{9.14}$$

The assertion of the lemma is now evident from (9.13) and (9.14), and this completes the proof. □

Proposition 9.2. *For any $p \in Z^n_+$ such that $\sum_{j=1}^{n} p_j \leq \delta$, the terms of the type $(\pi_{j=1}^{n} x_j^{p_j} - \pi_{j=1}^{n} \ell_j^{p_j})$ and $(\pi_{j=1}^{n} u_j^{p_j} - \pi_{j=1}^{n} x_j^{p_j})$ can each be expressed as a sum of nonnegative multiples of terms of the following form:*

$$\pi_{j \in J_1}(u_j - x_j) \pi_{j \in J_2}(x_j - \ell_j) \pi_{j \in J_3} x_j \text{ where } (J_1 \cup J_2 \cup J_3) \subseteq \bar{N},$$

$$|J_1 \cup J_2 \cup J_3| \leq \delta. \tag{9.15}$$

Proof. First, note that $(\pi_{j=1}^{n} x_j^{p_j} - \pi_{j=1}^{n} \ell_j^{p_j})$ can be expressed in terms of nonnegative multiples of products of factors $(x_j^{p_j} - \ell_j^{p_j})$, $j = 1, \ldots, n$, by inductively applying

$$(\pi_{j=1}^{n} x_j^{p_j} - \pi_{j=1}^{n} \ell_j^{p_j}) = (\pi_{j=1}^{n-1} x_j^{p_j} - \pi_{j=1}^{n-1} \ell_j^{p_j})(x_n^{p_n} - \ell_n^{p_n})$$

$$+ (x_n^{p_n} - \ell_n^{p_n}) \pi_{j=1}^{n-1} \ell_j^{p_j} + \ell_n^{p_n} (\pi_{j=1}^{n-1} x_j^{p_j} - \pi_{j=1}^{n-1} \ell_j^{p_j}).$$

In a similar fashion, $(\pi_{j=1}^{n} u_j^{p_j} - \pi_{j=1}^{n} x_j^{p_j})$ can be expressed in terms of nonnegative multiples of products of factors $(u_j^{p_j} - x_j^{p_j})$ and $x_j^{p_j}$, $j = 1, \ldots, n$, by inductively applying

$$(\pi_{j=1}^{n} u_j^{p_j} - \pi_{j=1}^{n} x_j^{p_j}) = u_n^{p_n} (\pi_{j=1}^{n-1} u_j^{p_j} - \pi_{j=1}^{n-1} x_j^{p_j}) + (u_n^{p_n} - x_n^{p_n}) \pi_{j=1}^{n-1} x_j^{p_j}.$$

Hence, having done this, the assertion of Proposition 9.2 now follows by Proposition 9.1. This completes the proof. □

To prove the dominance result, we need one additional intermediary step, relating the constraints of type $[(9.15)]_L \geq 0$ with the regular RLT constraints (9.7c). Let us first define the following family of sets of constraints to identify all possible constructs of type (9.15).

$$\Omega_{s,\delta'} \equiv \{[\prod_{j\in J_1}(u_j - x_j)\prod_{j\in J_2}(x_j - \ell_j)\prod_{j\in J_3} x_j]_L \geq 0 \quad \forall$$

distinct ordered triplets (J_1, J_2, J_3) where $(J_1 \cup J_2 \cup J_3) \subseteq \overline{N}$,

$$|J_1 \cup J_2 \cup J_3| = \delta', \ |J_3| = s \tag{9.16}$$

for each $0 \leq s \equiv |J_3| \leq \delta' \leq \delta$. Notice that $\Omega_{0,\delta}$ is the set of regular RLT constraints (9.7c). Let us denote the feasible region defined by the constraint set $\Omega_{s,\delta'}$ by $\overline{\Omega}_{s,\delta'}$.

Proposition 9.3. *(a) For any* $0 \leq s \leq \delta$, $\overline{\Omega}_{s,\delta'} \supseteq \overline{\Omega}_{s,\delta'+1}$ *for all* $\delta' \in \{s, \ldots, \delta - 1\}$. *(b) For any* $\delta' \in \{s, s+1, \ldots, \delta\}$, $\overline{\Omega}_{s-1,\delta'} \subseteq \overline{\Omega}_{s,\delta'}$, *for all* $1 \leq s \leq \delta$.

Proof. For any $0 \leq s = |J_3| < \delta$ and $\delta' \in \{s, \ldots, \delta - 1\}$, consider the following constraint from $\Omega_{s,\delta'}$:

$$[\prod_{j\in J_1}(u_j - x_j)\prod_{j\in J_2}(x_j - \ell_j)\prod_{j\in J_3} x_j]_L \geq 0. \tag{9.17}$$

Surrogating the following two constraints from the set $\Omega_{s,\delta'+1}$,

$$[(x_t - \ell_t)\prod_{j\in J_1}(u_j - x_j)\prod_{j\in J_2}(x_j - \ell_j)\prod_{j\in J_3} x_j]_L \geq 0$$

and $[(u_t - x_t) \underset{j\in J_1}{\pi}(u_j - x_j) \underset{j\in J_2}{\pi}(x_j - \ell_j) \underset{j\in J_3}{\pi} x_j]_L \geq 0$ (9.18)

where $t \in \{1,\ldots,n\}$, we obtain that the sum equals $(u_t - \ell_t)$ times (9.17), hence implying (9.17). Therefore, we have $\overline{\Omega}_{s,\delta'} \supseteq \overline{\Omega}_{s,\delta'+1}$. This proves part (a).

To prove part (b), for any $1 \leq s \leq \delta' \leq \delta$ and $t \in \{1,\ldots,n\}$, consider the following constraint from $\Omega_{s,\delta'}$:

$$[x_t \underset{j\in J_1}{\pi}(u_j - x_j) \underset{j\in J_2}{\pi}(x_j - \ell_j) \underset{j\in J_3}{\pi} x_j]_L \geq 0 \tag{9.19}$$

for any $|J_3| = s-1$, $J_1 \cup J_2 \cup J_3 \cup \{t\} \subseteq \overline{N}$, $|J_1 \cup J_2 \cup J_3 \cup \{t\}| = \delta'$. Notice that (9.19) is well defined, since $s = |J_3 \cup \{t\}| \geq 1$. We can obtain (9.19) by a particular surrogate of the following constraint from $\Omega_{s-1,\delta'}$,

$$[(x_t - \ell_t) \underset{j\in J_1}{\pi}(u_j - x_j) \underset{j\in J_2}{\pi}(x_j - \ell_j) \underset{j\in J_3}{\pi} x_j]_L \geq 0 \tag{9.20a}$$

with the following constraint from $\Omega_{s-1,\delta'-1}$, where the latter constraint has been shown to be implied by the constraints in $\Omega_{s-1,\delta'}$, in part (a) of this proposition,

$$[\underset{j\in J_1}{\pi}(u_j - x_j) \underset{j\in J_2}{\pi}(x_j - \ell_j) \underset{j\in J_3}{\pi} x_j]_L \geq 0. \tag{9.20b}$$

Using $\ell_t \geq 0$ as the weight for (9.20b), we have that (9.20a) + ℓ_t (9.20b) = (9.19). Hence, each constraint in $\Omega_{s,\delta'}$, is implied by those in $\Omega_{s-1,\delta'}$, and this completes the proof. □

Proposition 9.4. $v[LP(PP)] \geq v[LP(\overline{QPP})]$.

Proof. By Proposition 9.2, the compound factors $(\prod_{j=1}^{n} x_j^{p_j} - \prod_{j=1}^{n} \ell_j^{p_j})$ and $(\prod_{j=1}^{n} u_j^{p_j} - \prod_{j=1}^{n} x_j^{p_j})$ can each be expressed as a sum of nonnegative multiples of the terms of the form (9.15). Hence, so can the pairwise products of these compound factors. Consequently, constraints (9.9) can be obtained as surrogates of the constraints from the combined set $\{\Omega_{s,\delta'},\ 0 \le s \le \delta' \le \delta\}$. Since $\overline{\Omega}_{0,\delta} \subseteq \overline{\Omega}_{s,\delta'}$ for all $0 \le s \le \delta' \le \delta$ by Proposition 9.3, the RLT constraints (9.9) of Problem LP($\overline{\text{QPP}}$) are all implied by those in $\Omega_{0,\delta}$, where the latter are the RLT constraints (9.7c) of Problem LP(PP). Hence, from (9.7) and (9.10), we have $\nu[\text{LP(PP)}] \ge \nu[\text{LP}(\overline{\text{QPP}})]$. This completes the proof. $\square$

Notice that in establishing the above result, we have not assumed that the restrictions $a_j^1,\ a_j^2 \le s_j,\ j = 1,\dots,n$, are imposed when generating the bound-factor products in (9.9). The removal of this restriction from (9.9) results in the generation of additional RLT constraints to be included in LP($\overline{\text{QPP}}$) upon linearization. Therefore, the above dominance result holds even when considering this potentially tighter linear programming relaxation than the all-encompassing quadrified relaxation LP($\overline{\text{QPP}}$).

9.2. A Computational Comparison: Evaluation of the Quadrification Process

To numerically illustrate the dominance of Proposition 9.4, we compare $\nu[\text{LP(PP)}]$ and $\nu[\text{LP}(\overline{\text{QPP}})]$ empirically below, using some test problems from the literature.

Table 9.1. Comparison of $\nu[\mathrm{LP(PP)}]$ versus $\nu[\mathrm{LP}(\overline{\mathrm{QPP}})]$.

Problem	(n, δ)	Known optimal value	$\nu[\mathrm{LP(PP)}]$	$\nu[\mathrm{LP}(\overline{\mathrm{QPP}})]$
Problem 1	(1, 3)	–4.5	–4.5	-9
Problem 2	(1, 4)	0	–875	–6375.0
Problem 3	(1, 6)	7.0	–17385.0	–2,292,825.0
Problem 4	(2, 4)	-5.50796	-6.750	-6.9867
Problem 5	(2, 4)	–118.705	–8108.0	–54764.0
Problem 6	(2, 4)	–16.7389	–28.5	–29.0

Problems 1-4 are from Visweswaran and Floudas (1992), and Problems 5 and 6 are from Ryoo and Sahinidis (1995). All the problems are scaled so that the lower and upper bounds on the variables are all zeros and ones, respectively. Table 9.1 presents the results obtained. In particular, the dominance is quite significant for Problems 2, 3 and 5. Also, in light of Remark 9.1, we have used $\delta' = 4$ in generating the RLT relaxations for Problem 1. However, even when we use only the RLT constraints of order $\delta = 3$ for LP(PP), LP(PP) yields a lower bound of –6, which still dominates the lower bound obtained via $\mathrm{LP}(\overline{\mathrm{QPP}})$. (This also illustrates that the bound via LP(PP) can be enhanced by generating RLT constraints of order greater than δ.)

To examine the consequence of strengthening the relaxation $\mathrm{LP}(\overline{\mathrm{QPP}})$ even further beyond the inclusion of all possible quadrified representations as mentioned after the proof of Proposition 9.4, we removed the restrictions $a_j^1,\ a_j^2 \le s_j,\ \ j = 1,\dots,n,$ from (9.9) for Problems 2 and 5. Although the inclusion of the consequent additional constraints improved the respective lower bounds to –5241.67 and –46654 for these problems, the difference from $v[\text{LP(PP)}]$ still remains significant. We now proceed to examine various additional classes of constraints and algorithmic strategies that can be used to enhance the relaxation LP(PP). We begin by presenting certain specialized relaxations for univariate polynomial programs and then discuss multivariate problems.

9.3. Relaxations for Univariate Polynomial Programs

Consider a univariate polynomial programming problem (**UPP**) and let us assume that the constraint set has been equivalently reduced to a compact interval as mentioned earlier, and has furthermore been scaled onto the unit interval in order to obtain the following representation. (We assume that constants are preserved within the objective function in order to maintain the same objective function value, and hence the same level of accuracy in a percentage-tolerance optimality criterion, as for the original problem instance.)

UPP: Minimize $\{\phi_0(x): \quad 0 \le x \le 1\}$. (9.21)

Adopting the regular RLT constraints from the basic **reformulation phase** of Chapter 7, we augment (9.21) with the implied restrictions

$$x^k(1-x)^{\delta-k} \ge 0 \quad \text{for} \quad k = 0, 1, \ldots, \delta. \tag{9.22}$$

Then, in the subsequent **linearization phase**, we substitute a single variable X_p in place of x^p for $p = 1, \ldots, \delta$ (note that $X_1 \equiv x$), and hence transform the augmented polynomial program into a lower-bounding linear program. As usual, denoting by $[\cdot]_L$

the linearized expression obtained by using such a substitution on a polynomial $[\cdot]$, this linear program is given as follows.

$$\text{Minimize } \{[\phi_0(x)]_L : \ [x^k(1-x)^{\delta-k}]_L \geq 0 \text{ for } k = 0, 1, \ldots, \delta,$$
$$0 \leq x \leq 1\}. \tag{9.23}$$

Additionally, in order to further tighten the relaxation (9.23), we introduce the following class of *convex variable-bounding* constraints. The motivation here is to generate additional valid inequalities that would relate the new RLT linearized variables with the original variables of the problem, and hence tend to enforce their interrelationships embodied in the substitution of X_p for x^p.

Let δ' be the highest degree term in $\phi_0(x)$ that has a positive coefficient, and let P denote the set of prime numbers in the set $\{2, \ldots, \delta'\}$. Then, this class of constraints is given by

$$(X_q)^p \leq X_{p \cdot q} \quad \forall\ p \in P \text{ such that } p \cdot q \leq \delta', \text{for each } q = 1, \ldots, \left\lfloor \frac{\delta'}{2} \right\rfloor. \tag{9.24}$$

For example, if $\delta' = 6$, so that $P = \{2, 3, 5\}$, we would generate the constraints $x^2 \leq X_2$, $x^3 \leq X_3$ and $x^5 \leq X_5$ for $q = 1$, $X_2^2 \leq X_4$ and $X_2^3 \leq X_6$ for $q = 2$, and $X_3^2 \leq X_6$ for $q = 3$. Note that if p is not prime, but $p \cdot q \leq \delta'$, then the corresponding constraint $(X_q)^p \leq X_{p \cdot q}$ is implied by the collection in (9.24). For instance, in the above example, the constraint $x^6 \leq X_6$ is implied since $x^6 = (x^2)^3 \leq X_2^3 \leq X_6$. Also, since these constraints (9.24) place lower bounds on the RLT variables that appear on their right-hand sides, they are typically useful whenever the objective coefficient of this RLT variable is positive. Hence, the motivation for using δ' in lieu of δ.

Incorporating (9.24) within (9.23), we derive a basic *convex lower bounding* problem (**C-LB**) stated below.

C-LB: Minimize $\{[\phi_0(x)]_L$: constraints in (9.23) and (9.24)$\}$. (9.25)

We now proceed to develop two other new classes of constraints, each of which will yield an enhanced lower bounding version of Problem C-LB.

9.3.1. Squared Grid-Factor Constraints (SGF) and Associated Problem SGF-LB

For the unit interval, define g uniform grid points $x_1, \ldots, x_g$ such that $0 < x_1 < x_2 \ldots < x_g < 1$. Denoting $h = \lfloor \delta / 2 \rfloor$, consider the choice of all possible distinct combinations of the grid point indices $\{1, \ldots, g\}$ taken h at a time, including repetitions. Let C_I represent the collection of all such index sets J. (For example, when $\delta = 6$, so that $h = 3$, the index sets J that comprise C_I would be $\{1,1,1\}$, $\{1,1,2\}$, etc. In general, there are $\binom{g+h-1}{h}$ such combinations.) Then, the proposed set of SGF constraints are given by

$$\left[\prod_{j \in J}(x - x_j)^2\right]_L \geq 0 \quad \forall\ J \in C_I \tag{9.26}$$

if δ is even, and in case δ is odd, we multiply the nonlinear expressions in (9.26) separately by each of the bound factors $x \geq 0$ and $(1 - x) \geq 0$ to again obtain valid polynomial constraints of degree δ. The resulting lower bounding problem is accordingly given as follows.

SGF-LB: Minimize $\{[\phi_0(x)]_L$: constraints in (9.25) and (9.26)$\}$. (9.27)

9.4. Squared Lagrangian Interpolation Polynomial Constraints (SLIP) and Associated Problem SLIP-LB

Using the uniform grid points $x_1, \ldots, x_g$ as defined in Section 9.3.1, consider all possible distinct combinations of these grid points taken $(h + 1)$ at a time, with no repetitions. Let C_{II} represent the collection of such index sets J, where $|C_{II}| = \binom{g}{h+1}$. Then the class of constraints SLIP impose valid restrictions, attempting to interrelate RLT variables, by squaring Lagrange interpolation polynomials that agree with the grid point values taken $(h + 1)$ at a time. These constraints are given by

$$\left[\sum_{j \in J} \{\phi_0(x_j) \prod_{\substack{k \in J \\ k \neq j}} \frac{(x - x_k)}{(x_j - x_k)}\}\right]_L^2 \geq 0 \quad \forall\, J \in C_{II} \tag{9.28}$$

if δ is even, and in case δ is odd, then as for (9.26), we multiply the nonlinear expressions in (9.28) separately by each of the bound factors $x \geq 0$ and $(1 - x) \geq 0$. The corresponding lower bounding problem in this case is given as follows.

SLIP-LB: Minimize $\{[\phi_0(x)]_L$: constraints in (9.25) and (9.28)$\}$. (9.29)

9.5. Computational Evaluation of Relaxations for Univariate Problems

We now provide some preliminary computational results on a set of 7 univariate test problems from the literature having degree δ varying from 3 to 6 (see Visweswaran and Floudas, 1993). Our motivation here is to exhibit the strengths of the various proposed relaxations in terms of the lower bound produced relative to the known global optimum for these test problems. Toward this end, as a benchmark, we also provide lower bounds obtained via the exponential transformation method of geometric programming (abbreviated ET-LB) as used by Maranas and Floudas (1994). (For uniformity in comparison, this lower bounding problem was also mapped onto the unit interval; the resulting problem actually yielded better bounds than Maranas and Floudas' suggestion of mapping the problem onto $[0, \beta]$ for $\beta < 0.2$.)

Table 9.2 presents the results obtained, where for any Problem P, as before, $v(P)$ denotes its optimal objective value. All lower bounding problems were solved using the commercial package GAMS-MINOS 5.2 on an IBM 3090 computer. For the problems SGF-LB and SLIP-LB, we have used $g = 9$ uniform grid points. Notice that Problem SGF-LB yielded very tight bounds, with reasonable computational effort, recovering the optimal solution for 4 out of the 7 test problems. Problem SLIP-LB produced somewhat competitive results. Both these classes of constraints significantly enhance the lower bounds produced by the basic convex-RLT problem C-LB. Moreover, all these bounds substantially dominate the bounds produced by ET-LB. Hence, the proposed classes of constraints hold promise for solving the more challenging multivariate polynomial

Table 9.2 Comparison of bounding schemes for univariate polynomial programs.

Prob-lem	δ	Known Optimum	v(ET-LB)	v(C-LB)	v(SLIP-LB)	v(SGF-LB)	cpu secs for SGF-LB
1	3	-4.5	-47.63	-5.62	-4.5	-4.5	0.07
2	4	0	-67649.82	-752.78	0	0	0.13
3	4	-7.5	-65807.06	-715.28	-8.22	-7.5	0.16
4	5	-443.67	-114608.49	-1231.07	-542.27	-444.37	0.38
5	6	7	-62770870.65	-15202.35	6.94	7	0.49
6	6	0	-10549414.19	-2565.20	-0.564	-0.417	0.63
7	6	-29763.233	-34107365.19	-34598.47	-34598.47	-30415.75	0.52

programming problems. In particular, we explore the use of the regular RLT constraints, the convex variable bounding constraints, and a collection of constraint-factor based restrictions within a branch-and-bound algorithm for Problem PP over the next two sections. Similar to the motivation of this lattermost set of valid inequalities, an effective extension of the SGF and SLIP types of constraints for multivariate problems, is suggested for future study.

9.6. Relaxations and an Algorithm for Multivariate Problems

In this section, we consider the solution of constrained multivariate polynomial programming problems PP. Similar to the univariate case, we assume that the variables in the problem have been scaled onto the unit interval for the sake of convenience in deriving lower bounding problems, so that $\ell_j \equiv 0$ and $u_j \equiv 1$ $\forall\ j = 1,\ldots,n$. We now proceed to describe some specific classes of constraints for generating these lower bounding problems, and then recommend algorithmic strategies that include such problems in a branch-and-bound procedure.

9.6.1. Regular RLT and Convex Variable Bounding Constraints

As in Chapter 7, define the *bound-factors* for the scaled problem as $x_j \geq 0$ and $(1 - x_j) \geq 0$ for $j = 1, \ldots, n$. Following the RLT scheme, we shall generate certain nonlinear implied constraints in the **reformulation phase**, and subsequently, in the **linearization phase**, we shall substitute a single variable for each distinct nonlinear term. However, unlike the basic RLT process, as in Section 9.3, we shall maintain certain simple nonlinear convex variable bounding constraints in the relaxation. Accordingly, let us first generate the usual *bound-factor product constraints* that ensure convergence properties as follows.

$$\left[\prod_{j\in J_1} x_j \prod_{j\in J_2} (1 - x_j)\right]_L \geq 0 \quad \forall \quad J_1 \cup J_2 \subseteq \overline{N},\ |J_1 \cup J_2| = \delta, \tag{9.30}$$

where in the present context, $[\cdot]_L$ denotes the linearization process for the expression $[\cdot]$ in which we substitute

$$X_J = \prod_{j\in J} x_j \quad \forall \quad J \subseteq \overline{N}, \tag{9.31}$$

and where throughout, $J_1 \cup J_2$ will mean the joint collection of indices (*preserving repetitions*) that appear in J_1 and J_2. Note that the constraints (9.30) are generated by taking products of x_j and $(1 - x_j)$, $j = 1,\ldots,n$, δ at a time, including possible repetitions. For consistency in variable definitions, we assume that the indices in J for each variable X_J are sequenced in nondecreasing order, and that $X_{\{j\}} \equiv x_j \quad \forall\ j \in N$, and $X_\varnothing \equiv 1$.

Additionally, as in (9.24), we generate *convex variable-bounding* constraints separately for each variable x_i, $i = 1,\ldots,n$, as follows.

$$\left\{\left[x_i^q\right]_L\right\}^p \le \left[x_i^{p\cdot q}\right]_L \quad \forall \quad q = 1,\ldots,\lfloor \delta / 2 \rfloor \text{ and prime numbers}$$

$$p = 2,\ldots,\lfloor \delta / q \rfloor, \text{ for each } i = 1,\ldots,n. \tag{9.32}$$

9.6.2. Constraint-Factor Based Restrictions

The RLT relaxation can be further enhanced by incorporating additional restrictions that are generated by involving the structural constraints that define the set Z in Problem PP within the products constructed during the reformulation phase. Specifically, let us define nonnegative *constraint-factors* $\left[\phi_r(x) - \beta_r\right] \ge 0$, $r = 1,\ldots, R_1$, based on the inequality structural constraints in (9.1). Then, we can augment (9.30) by generating all possible distinct products of the joint collection of bound-factors and constraint-factors (including possible repetitions and involving at least one constraint-factor), so long as the degree of the resulting polynomial expression is equal to δ. (These constraints can be readily shown to imply similar constraints of degree lesser than δ, following Lemma 7.2.) As far as the equality constraints defining (9.1) are concerned, since similar products involving these constraints preserve the equality restriction, it is readily seen that the foregoing types of constraints are all implied by generating simply the following restrictions. Consider an equality constraint of degree $\delta' < \delta$. Then, we multiply this constraint by combinations (including repetitions) of the variables x_j, $j = 1, \ldots, n$, taken τ at a time, for $\tau = 1, \ldots, \delta - \delta'$. All these constraints are subsequently linearized using the substitution (9.31). We will refer to the resulting set of inequality and equality restrictions as **constraint-factor based restrictions**.

Remark 9.2. In the above discussion, we have always restricted the degree of any RLT constraint to be no more than the degree δ of the original polynomial program, merely to contain the size of the resulting relaxation. Although this is what we have implemented, it is actually possible to generate tighter relaxations by constructing RLT constraints of order exceeding δ, if the size of the resulting relaxation is manageable. (The next subsection addresses size reduction techniques that might permit the use of such a strategy.) □

Remark 9.3. In preserving the degree δ of the polynomial program in the reformulation phase of RLT, note that we are unable to use constraint-factors that themselves have a degree of δ in any further product constraints. This can be remedied by projecting out the terms of degree δ by using appropriate surrogates with the bound-factor product constraints (9.30). (For example, consider a "≥" type linearized constraint of degree δ, and suppose that we maximize the expression from this constraint that is comprised of the terms having degree δ subject to appropriately chosen linearized bound-factor constraints along with $x \in \Omega$. Then these terms of degree δ can be eliminated from the original constraint by using surrogates based on the optimal dual multipliers thus obtained.) The resulting reduced-degree implied constraints can then be used to generate additional constraint-factor based restrictions. We provide some sample results using this strategy in Section 9.6. □

9.6.3. Constraint Filtering Scheme and Lower Bounding Problem

The principal purpose of the RLT constraints is to introduce inter-relationships among the variables X_J, $J \subseteq \overline{N}$, that would tend to enforce equality in (9.31) at an optimum to the underlying RLT relaxation. The automatic constraint generation process constructs a general set of such relationships, not all of which might be actually helpful in tightening the relaxation for a specific problem instance. We can therefore attempt to predict RLT constraints that might be active at an optimum to the RLT relaxation by examining the signs on the coefficients α_{rt} in (9.2), and retaining those RLT constraints that impose restrictions on the corresponding RLT variables $X_{J_{rt}}$ that serve to oppose the direction in which the definitions (9.31) are more likely to be violated.

To present such a *Constraint Filtering Scheme*, let us view all the problem constraints (including the equalities), as well as the objective function, in a "≥" form, where the latter is represented as $z - \phi_0(x) \geq 0$, with z to be minimized. Accordingly, as in (9.2), we shall refer to the coefficients appearing with the various nonlinear terms $\prod_{j \in J_{rt}} x_j$ in this representation as $\overline{\alpha}_{rt}$. Furthermore, denote the complete set of linearized bound-factor and constraint-factor based RLT constraints as $f_i(x, X)$, for i in some index set M, say. We now use the following two-stage process to filter this set of constraints and select a subset $M_S \subseteq M$ to include within the proposed relaxation for computing lower bounds.

Stage 1. For each nonlinear term associated with the set J_{rt} for which $\overline{\alpha}_{rt} > 0$, include within M_S the RLT constraints that contain $X_{J_{rt}}$ with a negative coefficient. (The motivation here is that since $X_{J_{rt}}$ might therefore have a tendency to violate (9.31) in the increasing direction, we include RLT constraints that impose an upper bounding type of restriction on it.) Similarly, if $\overline{\alpha}_{rt} < 0$ for any nonlinear term appearing in the original problem, include within M_S the RLT constraints that contain $X_{J_{rt}}$ with a positive coefficient. Let $M_E = M - M_S$ index the set of currently *eliminated* constraints.

Stage 2. (To conserve effort, this stage is executed only if $|M| \leq 10^3$.) Consider a selected constraint $f_s(x, X) \geq 0$, $s \in M_S$, and an eliminated constraint $f_e(x, X) \geq 0$, $e \in M_E$. For any term in f_s having a nonzero coefficient, excluding linear x-variable terms and the terms appearing in the original problem (which have already been considered in Stage 1 above), if f_e has a nonzero coefficient of the opposite sign, then we say that *f_e provides a cover for f_s*. Let f_e provide a cover for f_i for all $i \in C_e \subseteq M_S$. We now select additional constraints $e \in M_E$ such that $|C_e| \geq 0.5|M_S|$, that is, constraints which provide a cover for at least 50% of the inequalities selected in Stage 1, and include these within M_S.

Remark 9.4. We can also filter the constraints (9.32) by selecting only those members for which $\left[x_i^{p \cdot q}\right]_L$ shows up with a negative coefficient in any of either the original linearized constraints or the selected RLT constraints as determined above. □

The proposed lower bounding problem is then given as follows.

$$\textbf{RLT-LB:} \text{ Minimize } \left[\phi_0(x)\right]_L \tag{9.33a}$$

$$\text{subject to } \left[\phi_r(x)\right]_L \geq \beta_r \text{ for } r = 1, \ldots, R_1 \tag{9.33b}$$

$$\left[\phi_r(x)\right]_L = \beta_r \text{ for } r = R_1 + 1, \ldots, R \tag{9.33c}$$

$$f_i(x, X) \geq 0 \text{ for } i \in M_S \tag{9.33d}$$

$$\text{Constraints (9.32) filtered as in Remark 9.4} \tag{9.33e}$$

$$0 \leq x_j \leq 1 \quad \forall \quad j = 1, \ldots, n,$$

$$\text{and } 0 \leq X_J \leq 1 \quad \forall \quad J \subseteq \overline{N}. \tag{9.33f}$$

Furthermore, to obtain a compatible basis for comparing the objective function and the constraints, particularly for the purpose of the partitioning scheme of Section 9.5.5, we assume that the objective function (9.33a) and the linearized constraints (9.33b) and (9.33d) have been *scaled* by dividing them by the Euclidean norm of their respective coefficient vectors.

9.6.4. Range-Reduction Strategies

As variously demonstrated in Hansen *et al.* (1991), Hamed and McCormick (1993), Ryoo and Sahinidis (1995), Maranas and Floudas (1994) and Sherali and Tuncbilek (1995a), a preprocessing of Problem PP in order to tighten the lower and upper bounds on the original problem variables can significantly enhance algorithmic performance. The range-reduction scheme we implemented was the one proposed by Maranas and Floudas (1994) in which they minimize and maximize each variable in turn subject to the constraints of their exponential transformation based relaxation, where the transformed objective function is also included in the constraints, being restricted to take on values lesser than or equal to a heuristically determined upper bound. This tightening of bounds on all the variables is cyclically repeated, so long as the volume of the resulting hyperrectangle continues to fall by at least a factor of 0.9. (Additionally, we can impose a limit of 10, say, on the total number of cycles executed.) Note that we can alternatively use our relaxation (9.33) in order to perform this range-reduction strategy, and as shown in Tuncbilek (1994), this usually results in a substantial tightening of the hyperrectangle bounds. However, to conserve effort due to the size of the relaxation (9.33), it is advisable to use some approximate subgradient-based scheme on a Lagrangian dual formulation of (9.33) in this case (see the Appendix). In our computations, we have used the bounds resulting from the latter strategy (see Tuncbilek, 1994, for the data).

9.6.5. Branch-and-Bound Algorithm

The branch-and-bound algorithm implemented to globally solve Problem PP is principally driven by the lower bounding relaxation RLT-LB defined in (9.33). We begin by preprocessing the problem to perform range-reductions as in Section 9.5.4. Thereafter, at each node of the branch-and-bound tree, we compute a lower bound via the transformed/scaled problem RLT-LB given in (9.33). If $(\bar{x}, \bar{X})$ solves this lower bounding problem, we use $x = \bar{x}$ to possibly update the incumbent solution, in case this is feasible to Problem PP. Additionally, up to a depth of two of the branch-and-bound tree, we applied the software MINOS to Problem PP using $\bar{x}$ as a starting solution in order to detect good quality feasible solutions early in the algorithmic process. Letting LB and UB denote the resulting lower bound and the incumbent upper bound, respectively, if (UB-LB) $\leq 0.01 \max\{0.1, |\text{UB}|\}$, then we fathom the corresponding node. Otherwise, we partition the node by splitting the interval of some variable x_p into two subintervals, cutting at the value $\bar{x}_p$. The specific choice of x_p is described below, and is an alternative to the choice described in (7.11) that turns out to be computationally superior, while preserving the arguments of the proof of convergence of Theorem 7.1. Finally, the node having the least lower bound is selected for further exploration of the branch-and-bound tree.

Branching Variable Selection Strategy. Let the objective function and the constraints be represented in the "≥" form as described in Subsection 9.5.3, having coefficients $\overline{\alpha}_{rt}$ as defined therein. Then, for each nonlinear term $\prod_{j \in J} x_j$ that appears in Problem PP, we compute a discrepancy index given by

$$D_J = \sum_{\substack{r \geq 0 \\ \ni J_{rt} = J \text{ for some } t \in T_r}} \max \left\{ 0, \quad \overline{\alpha}_{rt} \left[\overline{X}_{J_{rt}} - \prod_{j \in J_{rt}} \overline{x}_j \right] \right\}. \tag{9.34}$$

Note that if $D_J \equiv 0 \;\; \forall \;\; J$, then $\overline{x}$ is clearly an optimum for the underlying polynomial node subproblem. Hence, if this node is not fathomed, we must have $D_{\overline{J}} = \max\{D_J :\ J$ represents a monomial in PP$\} > 0$. Now, based on the same motivation as for (9.34), we compute a cumulative penalty for each candidate variable x_k, for $k \in \overline{J}$, as follows.

$$\theta_k = \sum_{r \geq 0} \sum_{\substack{t \in T_r \\ \ni k \in J_{rt}}} \max \left\{ 0, \quad \overline{\alpha}_{rt} \beta_{krt} \left[\overline{X}_{J_{rt}} - \prod_{j \in J_{rt}} \overline{x}_j \right] \right\} \tag{9.35}$$

where β_{krt} is the number of times the index k appears in J_{rt}. Note that this factor β serves to ascribe a greater penalty for variables that introduce a higher degree of nonlinearity in the discrepant terms. The branching variable x_p is then selected as the one that yields the largest value of θ_k, for $k \in \overline{J}$, with ties broken arbitrarily, and its bounding interval is split into two at the value $\overline{x}_p$.

9.7. Computational Results

In this section, we evaluate the proposed algorithm using a set of nine practical application test problems from the literature, denoted PP1-PP9. The statement and source of each of these problems is given in Tuncbilek (1994); Table 9.3 gives the number of variables (n) and the degree (δ) of each problem along with the known optimal objective function value. The nature of the structural constraints of the problems is as follows. Problem PP1 has a 4^{th} degree equality constraint. Problem PP2 has one 4^{th} degree inequality and one quadratic equality constraint. Problem PP3 is a flywheel design problem having one 3^{rd} degree and one 4^{th} degree inequality constraints. Problem PP4 is an alkylation process design problem having five linear inequalities, three 3^{rd} degree inequalities, and three quadratic inequalities. Problem PP5 has six quadratic inequalities. Problem PP6 is a heat exchanger design problem having three linear and three quadratic inequalities. Problem PP7 is a reactor network design problem having six quadratic equalities and one linear inequality constraint. Problem PP8 is a pooling problem having three linear equalities, one quadratic equality, and two quadratic inequalities. Finally, Problem PP9 is a three-stage membrane separation problem having four linear inequalities and ten quadratic inequality constraints.

Table 9.3 presents the results obtained. All runs were made on an IBM 3090 computer. The codes were written in FORTRAN and an available version of the software MINOS 5.1 was used as a subroutine to solve the bounding problems. The constraint filtering strategy of Subsection 9.5.3 eliminated 30-90% of the constraints without significantly affecting the lower bounds. For all the problems, except PP3, a near optimum was found at the initial root node itself, whose value did not improve by more than 0.5% over the remaining nodes. For Problem PP3, the final incumbent solution (which is slightly better than the optimum reported in the literature) was found after enumerating three

Table 9.3. Results for multivariate polynomial programming problems.

Problem	$(n,\ \delta)$	$\|M\|$	$\|M_s\|$ in (9.33d)	# constrs in (9.33e)
PP1	(2, 4)	35	13	3
PP2	(4, 4)	344	240	12
PP3	(3, 7)	981	140	11
PP4	(10, 3)	2723	487	9
PP5	(5, 2)	55	29	1
PP6	(8, 2)	190	15	0
PP7	(8, 2)	153	27	2
PP8	(9, 2)	153	27	2
PP9	(13, 2)	435	61	3

Problem	Known opt. value	Best soln found	Initial LB	# Nodes enum.	Total cpu secs
PP1	–16.7389	–16.7389	–16.7389	1	0.97
PP2	17.01417	17.01402	16.75834	3	15.05
PP3	-5.6848	-5.6852	-5.6847	3	20.30
PP4	–1768.807	–1768.807	–1795.572	3	21.24
PP5	-30665.539	-30665.527	-30765.071	1	1.05
PP6	7049.25	7049.24	5000.83	155	35.42
PP7	–0.3888	–0.3888	–0.4708	139	30.96
PP8	–400.00	–400.00	-500.00	3	1.99
PP9	97.588	97.588	97.588	1	3.31

nodes. Furthermore, the initial lower bounds for all the problems were tight enough to require the enumeration of only 1-3 nodes, except for Problems PP6 and PP7 which required the enumeration of 155 and 139 nodes, respectively. Evidently, these two problems can benefit by additional classes of tightening constraints and perhaps, enhanced range-reduction techniques. However, we comment here that when we employed the *looser* initial bound restrictions for Problem PP6 as specified in Maranas and Floudas (1994), this problem was solved after enumerating only 25 nodes in 6.45 cpu seconds. Apparently, this was induced by the alternative combinations of partitionings performed by the algorithm. Also, for Problem PP8, for example, when we applied the degree reduction idea of Remark 9.3 to obtain additional selected constraints via the quadratic restrictions, the lower bound at the initial node increased to –400.00, matching with the optimum value. Hence, the problem was solved at this initial node itself, while consuming a total of 0.569 cpu seconds.

To summarize, in this chapter, we have developed some tight reformulation-linearization/ convexification based relaxations for univariate and multivariate polynomial programming problems, along with effective algorithmic strategies for determining a global optimum for such problems. Our computational results on a set of test problems arising in various types of applications indicate that overall, good quality solutions can be obtained in practice for these difficult nonconvex programming problems by simply performing the analysis at the initial node, or by enumerating only a few nodes.

Further enhancements are possibly by generating additional classes of (filtered) constraints, perhaps based on an extension of the constraints developed in Section 9.3 for the univariate case to the multivariate case, or based on the ideas suggested in Remarks 9.2 and 9.3, as well as through the development of improved range-reduction strategies, and the use of Lagrangian relaxation techniques to solve the relaxations more efficiently.

PART III

SPECIAL APPLICATIONS TO DISCRETE AND CONTINUOUS NONCONVEX PROGRAMS

10

APPLICATIONS TO DISCRETE PROBLEMS

In Chapters 2 and 3 we have presented a general theoretical framework for generating a hierarchy of relaxations spanning the spectrum from the continuous linear programming relaxation to the convex hull representation. In practice, the actual relaxation adopted needs to be carefully designed using these general concepts and results as guidelines, being specialized for the particular application. Often, even a full first-level relaxation might be too large to manage computationally, although this is highly dependent on linear programming technology. For example, as mentioned in Section 2.5, first as well as second-level relaxations have been successfully implemented for the notorious quadratic assignment problem, despite their massive size; such problems have as many as 15-25 facilities and location sites.

There are several alternatives that one can explore, ranging from the generation of full or partial first-level relaxations for the entire problem, to even a convex hull representation for a subset of the problem. Inherent special structures can be exploited and conditional logic can also be used to tighten RLT constraints as discussed in Chapter 3. Another approach that can be adopted is to generate cutting planes based on solving suitable RLT relaxations that are composed by using bound-factors that involve variables which fractionate in an original LP relaxation. The cutting planes can then be generated by suitably surrogating the constraints of the relaxation using optimal dual variables. This is frequently referred to as the *lift-and-project* strategy, where the *lifting* refers to the higher-dimensional RLT relaxation that is generated and solved, and the *projection* operation involves the surrogation of the RLT constraints using optimal dual variables in a manner that eliminates the new RLT variables from the resulting inequality.

There is another aspect that needs to be heeded in an actual implementation. Because of the nature of the RLT relaxations whereby the structure of the original constraints is replicated in blocks, and whereby groups of constraints become active when variables take on binary values, these relaxations turn out to be highly degenerate and their bases are often very ill-conditioned. As a result, both simplex-based and interior point methods experience difficulty in solving these relaxations. On the other hand, Lagrangian relaxation/dual approaches can exploit such structures without being encumbered by their size or by degeneracy and ill-conditioning effects, and offer a worthwhile alternative that must be seriously considered in any application.

We will illustrate many of these features in the present chapter by discussing two particular successful applications of RLT (see Chapters 1 and 11 for a discussion on other applications). The following section discusses an application of RLT concepts to solve mixed-integer bilinear programming problems, and Section 10.2 describes an RLT methodology for solving 0-1 quadratic programming problems. We close this chapter by presenting in Section 10.3 a brief synopsis on several other applications for which RLT methods have been gainfully employed to enhance the performance of algorithms.

10.1. Mixed-Integer Bilinear Programming Problems

Consider the following class of mathematical programs referred to as *mixed-integer bilinear programming problems* (MIBLP).

MIBLP: Minimize $\{c^t x + d^t y + x^t Cy : x \in X, y \in Y, y \text{ binary}\}$ (10.1)

where X and Y are nonempty, bounded polyhedral sets given respectively as

$$X = \left\{x \in R^m : \sum_{i=1}^{m} a_{ki} x_i = b_k \text{ for } k = 1, \ldots, K, x \geq 0\right\}$$

and

$$Y = Y_1 \cap Y_2 \cap Y_3,$$

with

$$Y_1 = \left\{y \in R^n : \sum_{j=1}^{n} G_{\ell j} y_j \geq g_\ell \text{ for } \ell = 1, \ldots, L\right\},$$

$$Y_2 = \left\{y \in R^n : \sum_{j=1}^{n} H_{pj} y_j = h_p \text{ for } p = 1, \ldots, P\right\},$$

and

$$Y_3 = \left\{y \in R^n : 0 \leq y_j \leq 1 \text{ for } j = 1, \ldots, n\right\}.$$

We denote the components of c, d and C as c_i, d_j and C_{ij} respectively, $i = 1, ..., m, j = 1, ..., n$.

Problems of this type arise in various production, location-allocation, and product distribution situations. For example, the variables x may denote quantities shipped between designated origin-destination (O-D) pairs (for instance, from various plants to customers), with $x \in X$ denoting some transportation or more general network flow type of constraints. The variables y, which assume binary values, are logical variables which may signify the decisions whether or not to adopt some new operations on the products manufactured, or to incorporate some new technologies, or the decision to construct some intermediate service or processing facilities on a transshipment network. Accordingly, the objective function records the costs of such operations and decisions. This includes a cross-product term which may either subsidize total costs with negative terms as in the case of adopting new technologies or providing specialized intermediate facilities/departments for performing certain job operations more efficiently, or the cross-product term may represent positive costs for providing additional mandatory services or product improvements. These costs may in general be specific to each O-D pair. In a likewise manner, the constraints $y \in Y$ may represent managerial budgetary and product quality control types of constraints, or set covering constraints to ensure that at least one product improvement which imparts a certain property is adopted, or that at least one facility in each geographically designated region is constructed. Problem MIBLP also arises as an equivalent restatement of the linear complementarity problem LCP (see Chapter 11).

Observe that Problem MIBLP possesses the bilinearity property, namely, for a fixed y it reduces to a linear program in x, and for a fixed x, it reduces to a zero-one linear integer program in y. Furthermore, in special cases in which the vertices of Y are all binary valued, as in the discrete location-allocation problem studied by Sherali and Adams (1984), or in the two-stage assignment problem given in Vaish (1974), or for LCPs as seen in Chapter 11, Problem MIBLP may be equivalently solved as a continuous bilinear programming problem. However, our experience leads us to believe that in such cases it is far more efficient to exploit, rather than ignore, the binary restrictions on the y-variables. Adams and Sherali (1993) discuss several readily solvable special cases for Problem MIBLP. In this chapter, we focus on the general problem in the form (10.1) and illustrate how one might compose a suitable partial first-level RLT relaxation for this problem, and accordingly, design an effective solution algorithm.

10.1.1. An Equivalent Linear Reformulation

In this section, we prescribe a specialized partial first-level RLT application to transform Problem MIBLP into an equivalent linear mixed-integer program. Observing the nature of the cross-product terms and the constraints defining the problem, we introduce the following sets of RLT constraints into the problem.

(i) nK constraints obtained by multiplying the K equalities defining X by each y_j, $j = 1, \ldots, n$;

(ii) $m(L + P + n)$ constraints obtained by multiplying the constraints in Y_1, Y_2, and the constraints $y_j \leq 1$, $j = 1, \ldots, n$, in Y_3 by each x_i, $i = 1, \ldots, m$; and

(iii) mL constraints obtained by multiplying the constraints in Y_1 by each $(x_i^+ - x_i)$, $i = 1, \ldots, m$, where

$$x_i^+ = \max\{x_i : x \in X\}. \tag{10.2}$$

Observe that in this application of RLT, we have focused on the bilinear terms and have neglected to exploit the binary property of the y-variables; nowhere do we use $y_j^2 = y_j \ \forall \ j$. We could have, for example, constructed the level-one RLT representation corresponding to the set Y as in Chapter 2 to obtain a potentially strengthened relaxation. However, to maintain a reduced problem size, we have opted not to construct such additional variables and constraints in this case. Note also that (ii) and (iii) above involve products using special generalized constraint factors (as in Chapter 3) on $x \geq 0$ and on $x_i \leq x_i^+, \forall \ i$, which are obtainable as surrogates of X using optimal dual multipliers for the subproblems (10.2). Of course, if $x_i^+ = 0$ for some i, then x_i may be eliminated from the problem. Then, substituting w_{ij} for $x_i y_j$ in the resulting program for each $i = 1, \ldots, m$, $j = 1, \ldots, n$, and adding $w \geq 0$, we obtain the following linearized mixed-integer program MIP.

MIP: Minimize $\quad z = \sum_i c_i x_i + \sum_j d_j y_j + \sum_i \sum_j C_{ij} w_{ij}$

$$\text{subject to} \quad \sum_i a_{ki} w_{ij} = b_k y_j \qquad \forall\ (j, k), \tag{10.3a}$$

$$\sum_j G_{\ell j} x_i^+ y_j - g_\ell x_i^+ \geq \sum_j G_{\ell j} w_{ij} - g_\ell x_i \geq 0\ \forall\ (i, \ell) \tag{10.3b}$$

$$\sum_j H_{pj} w_{ij} = h_p x_i \qquad \forall\ (i, p) \tag{10.3c}$$

$$0 \leq w_{ij} \leq x_i \qquad \forall\ (i, j) \tag{10.3d}$$

$$x \in X,\ y \in Y_2 \cap Y_3, \quad y \text{ binary}. \tag{10.3e}$$

Noting the boundedness of X and the RLT constraints (10.3a) and (10.3d), in particular, it is easy to verify (see Adams and Sherali, 1990) that Problems MIBLP and MIP are equivalent in the following sense. For any $(\bar{x}, \bar{y})$ feasible to MIBLP, there exists a $\bar{w}$ such that $(\bar{x}, \bar{y}, \bar{w})$ is feasible to MIP with the same objective value and, conversely, for any $(\bar{x}, \bar{y}, \bar{w})$ feasible to MIP, $(\bar{x}, \bar{y})$ is feasible to MIBLP with the same objective value. In fact, in either case, we necessarily have $\bar{w}_{ij} = \bar{x}_i \bar{y}_j \ \forall\ (i, j)$.

Problem MIP contains $Kn + m(2L + P + n)$ constraints in addition to the constraints in (10.3e) and the nonnegativity restrictions on w in (10.3d), and contains $mn + m + n$ variables, where m, n, K, L, and P are as defined in Problem MIBLP. Various linearization schemes, more concise than Problem MIP, have been proposed by Peterson (1971) and Glover (1975), that can be applied to Problem MIBLP with straightforward modifications. These strategies, however, yield considerably weaker reformulations in terms of their continuous relaxations than does Problem MIP. In fact, it is demonstrated by Adams and Sherali (1990) that the constraints of these other formulations may be obtained by taking surrogates of the constraints of Problem MIP. Hence, all dual-based inequalities, including Benders' cuts (1962) and strongest surrogate constraints of Rardin and Unger (1976), available through these other formulations are also available through MIP, though not necessarily vice versa.

10.1.2. Design of an Algorithm

The algorithm described in this section for solving Problem MIP, and hence Problem MIBLP, has evolved through both theoretical and empirical considerations. We present a general schema of the algorithm to provide the reader with the concepts of a suitable algorithmic approach for handling the structure of MIP.

To begin with, standard logical preprocessing tests (see Nemhauser and Woolsey, 1989) are performed on the set Y in order to possibly fix certain y-variables at zero or one based on feasibility considerations. Next, in order to tighten the linear programming relaxation, we compute

$$v_L = \min\left\{\sum_j y_j : y \in Y\right\} \text{ and } v_U = \max\left\{\sum_j y_j : y \in Y\right\},$$

and incorporate the inequalities $\sum_j y_j \geq \lceil v_L \rceil$ and $\sum_j y_j \leq \lfloor v_U \rfloor$ in Y (or Y_1), where $\lceil v_L \rceil$ and $\lfloor v_U \rfloor$ are rounded up and down values of v_L and v_U, respectively. (While these cuts only mildly contribute to the strength of the linearization, they are inexpensive to generate and hence are included in the preprocessing stage.)

Next, a strongest surrogate Benders' constraint is generated. As in Rardin and Unger (1976), this constraint may be generated by solving the continuous relaxation $\overline{\text{MIP}}$ of

Problem MIP and then using the resulting optimal dual multipliers in the usual manner to obtain a Benders' cut in terms of the "complicating" y-variables. Of course, any dual feasible solution which is near-optimal would serve just as well to derive such a cut. This is fortunate because $\overline{\text{MIP}}$ is a large-scale linear program for reasonably-sized MIBLP problems, and obtaining optimal solutions via the simplex method or even interior point methods can be time consuming (see Subsection 10.1.3). We implemented a subgradient optimization scheme for a Lagrangian dual of the problem along with a dual ascent technique in order to obtain, with a reasonable amount of effort, an appropriate set of dual multipliers for such a cut. The details of this technique are given in Adams and Sherali (1993) and involve the addition of implied surrogate constraints in the Lagrangian subproblem (in addition to X and Y) in order to control the dual ascent scheme. Let us denote the resulting strongest surrogate Benders' constraint as

$$z \geq \delta^0 + \sum_j \gamma_j^0 y_j. \tag{10.4}$$

Once (10.4) is generated, the following linear programming problem is solved

$$v(\overline{\text{MIP}}) = \min\left\{\delta^0 + \sum_j \gamma_j^0 y_j \ : \ y \in Y\right\}. \tag{10.5}$$

The linear program (10.5) yields three useful byproducts. First, since (10.4) is generated from a (near-optimal) dual feasible solution, $v(\overline{\text{MIP}})$ provides a (tight) lower bound on the objective value of $\overline{\text{MIP}}$ and hence, a lower bound for Problem MIBLP. Second, one may attempt to round or truncate the optimal solution to (10.5) in order to possibly obtain a binary solution $\bar{y}$ that is feasible to Y. Consequently, an upper bound $(\bar{x}, \bar{y})$

with objective value $\bar{v}$ may be found for MIBLP by solving MIBLP with $y \equiv \bar{y}$ fixed. Of course, if $\bar{v}$ and $v(\overline{\text{MIP}})$ are fortuitously close enough, one may terminate the algorithm. Third, one may use the optimal dual multipliers from (10.5) to obtain a surrogate constraint

$$\sum_j r_j y_j \geq R \tag{10.6}$$

of Y (which may be an equality depending on the nonzero dual multipliers). This constraint may be used to perform logical tests and also to obtain quick, though weaker, lower bounds in an implicit enumeration framework by solving bounded-variable knapsack problems over it, rather than general linear programs over Y. In such a context, given a partial solution with some y-variables fixed, it is further advantageous to update the surrogate constraint (10.6). To conserve computational effort, this may be done by simply using the dual multipliers from (10.5) on only those constraints in Y which are not rendered redundant by the fixed y-variables. Henceforth, we will assume that this strategy has been used to obtain the appropriate constraint (10.6).

The algorithm we propose is basically an implicit enumeration strategy. The bookkeeping used is a combination of the depth-first and best-first methods. A certain maximum limit on the number of active tree end nodes is specified, and the enumeration scheme works as a best-first strategy so long as this maximum number is not attained, and then reverts to a depth-first approach. At each node selection stage, an active end node having the least lower bound is selected. If the maximum permissible number of active end nodes already exists, this node is further explored in a depth-first fashion.

Otherwise, an extra end node is created by finding that node closest to the root node on the path from the root node to the current node, for which the other successor node has not yet been examined, and then creating this successor node.

To analyze each node and to select a branching variable if necessary, the algorithm uses four ingredients. First, it uses the strongest surrogate constraint (10.4) to compute bounds. In view of this, the procedure may be thought of as a Lagrangian relaxation approach in which the dual multipliers are not revised when certain variables are fixed. Second, the algorithm generates suitable Benders' cuts whenever feasible zero-one y-variable completions are found which cannot be fathomed using the currently available information. Following Magnanti and Wong (1981), we suitably select from among the alternative optimal dual solutions for a fixed y in MIP in order to generate strong Benders' cuts. This is done by adding implied surrogate constraints that control the choice of the dual optimal solution (see Adams and Sherali, 1993, for the particular constraints incorporated and their use in this context). Let us denote the resulting Benders' cuts as

$$z \geq \delta^e + \sum_j \gamma_j^e y_j \quad \text{for } e = 1, \ldots, E, \text{ say.} \tag{10.7}$$

Third, the algorithm generates certain disjunctive cuts whenever it finds feasible completions $\hat{y}$ whose objective value is not greater than $\bar{v} + 0.1(\bar{v} - v(\overline{\text{MIP}}))$, where $\bar{v}$ is the incumbent objective value for MIP. (The value 0.1 was chosen empirically.) Such completions are likely to be in the vicinity of the optimum, and the disjunctive cut not only strengthens the linear programming relaxation in this region, but the process of generating it provides a further opportunity to improve the incumbent solution. Specifically, this cut is generated from $\hat{y}$ in the following manner. Each component of $\hat{y}$ is complemented in turn to verify that the resulting adjacent extreme point of the unit hypercube is either infeasible to the present constraint system, or does not improve upon the current incumbent. (Otherwise, a new incumbent is located and the process is repeated at this point.) When this is verified, we can invoke the disjunction that any feasible, improving y solution must have at least two components of value different from that in $\hat{y}$, and hence the following disjunctive cut may be generated:

$$\sum_{j:\hat{y}=1} (1 - y_j) + \sum_{j:\hat{y}_j=0} y_j \geq 2. \tag{10.8}$$

We do caution the reader to judiciously decide on whether or not to use (10.8). For example, when X is well structured (for instance network constraints), generating (10.8) may pay off well as it did in our computational experience. If, for example, Y is the assignment set, then (10.8) is worthless, although a conceptually equivalent disjunctive cut based on pairwise exchanges as described in Bazaraa and Sherali (1981) may be used instead. Finally, as a fourth ingredient, the algorithm performs logical tests on Y, (10.4), (10.7), and (10.8) and solves suitable linear programs to compute bounds.

More specifically, let us consider a node being explored at which J^+, J^-, and J_f, respectively denote the index sets of y-variables which are fixed at one, fixed at zero, and currently free. (If $|J_f| \leq 2$ at any stage of the procedure, we simply enumerate and explicitly test the possible completions.) Logical tests are initially performed in an attempt to fix variables and possibly fathom the partial solution. Then, the zero-completion is tested for feasibility to the current constraint-cut system. In case it is feasible, its bilinear objective is evaluated and this readily yields a Benders' cut (10.7). If the above-mentioned criterion is satisfied, a disjunctive cut (10.8) is generated. Of course, either of these steps may yield a new incumbent which is updated. Next, using the newly-generated Benders' cut, having index E in (10.7), a trivial lower bound is computed by adding to δ^E all the components γ_j^E for $j \in J^+$ and all the negative components γ_j^E for $j \in J_f$. If this does not fathom the current partial solution, a variable for branching on the value one is selected as being that free variable having the most negative coefficient γ_j^E. (Note that not all γ_j^E could be nonnegative, or else the lower bound would equal the objective value of the zero completion and would therefore be no lesser than $\bar{v}$, the objective value of the incumbent solution.)

On the other hand, if the zero completion is not feasible to the current constraint-cut system, then a lower bound LB(e) is computed using each Benders' cut $e = 0, 1, ..., E$ (i.e., including the strongest surrogate cut) in the same manner as the cut E was used above. (Note that LB(e), $e = 0, 1, ..., E$, are already available from the logical tests.) Next, an index g is chosen such that LB(g) $\geq$ LB(e), $e = 0, 1, ..., E$. Since LB(g) $< \bar{v}$ (otherwise the logical tests would have fathomed the partial solution), we strengthen this lower bound by computing

$$\text{LB1}(g) = \delta^g + \sum_{j \in J^+} \gamma_j^g + \min\{\sum_{j \in J_f} \gamma_j^g y_j : \text{(10.6) updated with respect to the free variables holds, and } 0 \le y_j \le 1 \text{ for } j \in J_f\}. \quad (10.9)$$

If this bounded-variable knapsack problem does not fathom the partial solution and its solution is infeasible to the current constraint-cut system, then a decision is made as to whether a further strengthened lower bound should be computed. If adding 0.1 of the difference between the sum of the positive γ_j^g, $j \in J_f$, and the knapsack objective value to LB1(g) exceeds $\bar{v}$, then a strengthened bound LB2(g) is calculated by replacing the knapsack problem for LB1(g) by the linear program over Y and the disjunctive cuts of the form (10.8), with y_j fixed appropriately for $j \in J^+ \cup J^-$. (Note that LB2(g) would equal LB1(g) if the solution to the knapsack problem for LB1(g) is feasible to Y and (10.8).) If no fathoming results, then a variable for branching on the value one is selected based on the optimal solution to LB2(g) if it is solved, or LB1(g) otherwise. Calling this solution $\tilde{y}$, that index $j \in J_f$ for which $\tilde{y}_j$ is a maximum, with ties broken by selecting the index corresponding to the smallest value of γ_j^g, is chosen as the branching variable index. We note additionally that we found it useful to round or truncate the optimal solutions that compute LB1(g) and LB2(g) in an attempt to improve the incumbent.

10.1.3. Computational Experience

In this section, we provide some comments on our computational experience in studying the contributions of the algorithmic strategies and the effects of problem data on the performance of the proposed algorithm. As motivated by the most common types of applications of MIBLP, for generating test problems, we selected X as the transportation constraint set and Y as having set covering constraints of density 0.25 or 0.5. Again, as motivated by applications, we generated the cross-product term coefficients C_{ij} uniformly either on the interval [–25, 0] or [0, 25]. In order to ensure difficult problems in either case, for the negative C_{ij} problems, each c_i was generated uniformly on $[-\sum_j C_{ij}, -2\sum_j C_{ij}]$ and each d_j was generated uniformly on $[-c_j^+, -c_j^-]$ where, for each j, $c_j^- = \min\{\sum_i C_{ij} x_i : x \in X\}$ and $c_j^+ = \max\{\sum_i C_{ij} x_i : x \in X\}$. For the positive C_{ij} problems, each c_i was generated uniformly on [0, 25] and each d_j was set at zero simply to permit the quadratic terms to dominate and make these problems more difficult. For the algorithm, a maximum of 300 Benders' cuts from binary $\bar{y}$-solutions were stored at a time, with additional cuts replacing earlier ones in a circular-list fashion. In a similar manner, the length of the circular-list storing the disjunctive cuts (10.8) was set at 100. Furthermore, once 100 disjunctive cuts were generated, additional cuts (10.8) were generated only when a new incumbent solution was discovered.

Adams and Sherali (1993) provide details of the results obtained. Here, we summarize the main observations and insights. First, the problems having negative C_{ij} coefficients turned out to yield a harder class of problems than those having positive C_{ij} coefficients.

Observe that in the latter case, an optimal solution happens to be a prime cover for our test problems, and this similarity with set covering problems may have made these problems relatively easier. Likewise, an increase in the density of Y makes the problems relatively harder to solve. Furthermore, for the problems having negative C_{ij} coefficients, the knapsack and linear programming bounds, LB1(g) and LB2(g) respectively, failed consistently to fathom partial solutions (see Equation (10.9)). However, for the positive C_{ij} coefficient problems, roughly 2% of LB1(g) and 8% of LB2(g) led to a fathoming of partial solutions. The index g almost always turned out to correspond to the strongest surrogate constraint whenever it was generated. (For some test runs, we did not derive the cut (10.4) to isolate its effectiveness.)

The most interesting insight was the role played by the strongest surrogate constraint. For all the problems, this constraint greatly reduced the effort required by the implicit enumeration scheme, and the payoff was quite dramatic as problem size increased. However, for the case of positive C_{ij} coefficients, because of the relative ease with which these problems were solved, the problem size had to be increased somewhat more (70 binary variables as compared with 20 binary variables) before the benefits derived from this constraint became convincingly evident. We remark here that some of our preliminary test runs were made with the strongest surrogate constraint derived from more compact linearizations of the type mentioned in Subsection 10.1.1. The effect of the strongest surrogate constraint in these cases was not at all as dramatic as with Problem MIP. Hence, the tighter linearization considerably benefits the solution procedure. For the size problems considered in our study (ranging up to 100 continuous and 70 binary variables) the average initial lower bound was about 94% of optimality.

The Benders' cuts also appeared to contribute to the efficiency of the solution process, as seen by turning this option on and off. This effect was relatively more pronounced on the negative C_{ij} coefficient problems since the Benders' cuts tend to be strong when the difference $(c_j^+ - c_j^-)$ is small as compared with $|d_j|$ $\forall j$.

The disjunctive cuts (10.8) also appeared to aid in the solution procedure (except for the problems having positive C_{ij} coefficients), by discovering new incumbent solutions. Some runs that we attempted by suppressing the generation of these cuts confirmed this observation. In addition to playing a role in logical tests and perhaps strengthening LB2(g) for some problem classes, these cuts, by discovering new incumbent solutions, helped in discounting the sensitivity of the algorithm to the selection of branching variables, and to the compromise between a depth and best-first scheme.

Finally, in order to provide some insights into the effort required by our Lagrangian relaxation procedure for solving *the continuous relaxations* $\overline{\text{MIP}}$ *of Problem* MIP as compared with that required by standard commercial packages, we attempted to solve the linear programming problems associated with three of the relatively larger (positive C_{ij} coefficient) problems using both CPLEX 1.2 and OB1. An IBM RS/6000 Model 320 computer was used, and the results are summarized in Table 10.1. (The run times do not include the time to input problem data.) For these problems, we found that the most

Table 10.1 Tests runs using CPLEX 1.2 and OB1 to solve the initial LP relaxation.

	CPLEX 1.2			OB1		
	Objective value	cpu time (secs)	# iterations	Objective value	cpu time (secs)	# iterations
Problem 1	2371.58	245.80	1920	2371.54	2081.19	32
Problem 2	2320.00	198.12	1423	2320.00	1814.11	19
Problem 3	2913.03	540.92	2962	2913.02	4153.10	33

efficient implementation of CPLEX 1.2 was to solve the dual using a steepest-edge pricing. Relative to the software OB1, we ran the test problems using the primal version with default parameter settings. We also tried running the problems using the dual version, but found this option to be less effective because of the density of the Cholesky factors. An attempt to solve the primal version while treating the dense columns separately was also found to be less efficient. In contrast with the reported efforts for solving just the initial LP relaxation, using the same computer, our approach took 43.75, 35.51, and 186.55 cpu seconds, respectively, to both solve the linear program *and* progress through the enumeration strategy and obtain an optimal (mixed) integer solution for these three problems. In all cases it took significantly less time to *optimally* solve Problem MIP using the proposed algorithm than to just solve the *linear program* $\overline{\text{MIP}}$ using either CPLEX 1.2 or OB1. Of course, since both these standard codes solve the linear programs to optimality, as opposed to generating a good-quality dual solution, the lower bounds provided by them are at least as strong as those found by our Lagrangian dual scheme. In fact, for Problem 2 of Table 10.1, the linear programming value proved to be the integer optimal solution value. In summary, the results indicate that the reformulated Problem MIP is a strong representation of MIBLP, and also, that it possesses a special structure which should be exploited in designing effective solution approaches for this class of problems. Further enhancements in generating tighter representations might be possible as discussed in Subsection 10.1.1.

10.2. Zero-One Quadratic Programming Problems

In this section we consider linearly-constrained zero-one quadratic programming problems (QP). Problems of this structure arise in numerous economic, facility location, and strategic planning situations.

QP: Minimize $\sum_{i=1}^{m} c_i x_i + \sum_{i=1}^{m-1} \sum_{j=1}^{m} D_{ij} x_i x_j$

subject to $x \in X$, x binary

where $X \equiv X_1 \cap X_2 \cap X_3$ is a nonempty polyhedron with

$$X_1 = \{x \in R^m : \sum_{i=1}^{m} a_{ki} x_i = b_k \text{ for } k = 1, \ldots, K\}$$

$$X_2 = \{x \in R^m : \sum_{i=1}^{m} G_{\ell i} x_i \geq g_\ell \text{ for } \ell = 1, \ldots, L\}, \text{ and}$$

$$X_3 = \{x \in R^m : 0 \leq x_i \leq 1 \text{ for } i = 1, \ldots, m\}.$$

We will focus in this section on using RLT constraints to develop a strong linearized reformulation of QP, for which an algorithm similar to that described in Subsection

10.1.2 for mixed-integer bilinear programs can be applied. Alternatively, Fortet (1959, 1960) (see also Watters (1967) and Zangwill (1965)) has proposed a linearization technique for zero-one polynomial programming problems which replaces each polynomial term with a single additional zero-one variable and two additional constraints. Other concise zero-one linear programming formulations of zero-one polynomial programming problems are found in Glover and Woolsey (1973). These formulations, as well as the ones by Fortet, require the adoption of additional zero-one variables. On the other hand, some proposed linearization techniques do not introduce additional zero-one variables, but introduce only additional continuous variables and additional constraints. The linearizations due to Glover (1975), Glover and Woolsey (1974), and Rhys (1970) are examples of such techniques. A similar technique for the special case of quadratic assignment problems has been presented by Bazaraa and Sherali (1980). The experience reported by Bazaraa and Sherali (1980), and a comment by McBride and Yormark (1980) seem to indicate that the linearization techniques do not yield very effective solution procedures for Problem QP.

This section summarizes the work in Adams and Sherali (1986), which demonstrates that the performance of algorithms based on equivalent linearizations can be dramatically improved by generating tighter model representations. We begin our discussion by presenting an RLT-based linearization for Problem QP that theoretically dominates the linearizations found in the literature. In Subsection 10.2.2, we propose an effective algorithm that exploits the strength of the derived linearization, and in Subsection 10.2.3, we summarize our computational experience. In particular, the strength of the linearization is verified by exhibiting the dramatic savings in computational effort that is achieved in comparison with the implementation of the same algorithmic strategies on an alternative existing linearization of Problem QP.

10.2.1. An Equivalent Linear Reformulation

The linearization that we discuss in this section is simply the first-level representation of Chapter 2. This involves multiplying the equality constraints defining X_1 by each x_j, $j = 1, \ldots, m$, the *inequality* constraints defining X_2 by each x_j and $(1 - x_j)$, and by including the inequalities of the form $f_2(J_1, J_2) \geq 0\ \forall\ (J_1, J_2)$ of order two (see Section 2.1). Linearizing the resulting formulation by substituting $w_{ij} = x_i x_j$ $\forall\ i < j$, and replacing x_i^2 by x_i $\forall\ i$, results in the following equivalent linear zero-one mixed-integer program (henceforth, MIP refers to this problem — not to be confused with MIP in Subsection 10.1.1).

MIP: Minimize $z = \sum_i c_i x_i + \sum_{i<j}\sum D_{ij} w_{ij}$

subject to $(a_{kj} - b_k)x_j + \sum_{i<j} a_{ki} w_{ij} + \sum_{i>j} a_{ki} w_{ji} = 0 \quad \forall\ (j, k) \qquad$ (10.10a)

$$\sum_i G_{\ell i} x_i - g_\ell \geq \sum_{i<j} G_{\ell i} w_{ij} + \sum_{i>j} G_{\ell i} w_{ji} + (G_{\ell j} - g_\ell) x_j \geq 0 \quad \forall\ (j, \ell) \qquad (10.10b)$$

$$x_i - w_{ij} \geq 0 \quad \forall\ (i, j),\ i < j \qquad (10.10c)$$

$$x_j - w_{ij} \geq 0 \quad \forall\ (i, j),\ i < j \qquad (10.10d)$$

$$-x_i - x_j + w_{ij} \geq -1 \quad \forall\ (i, j),\ i < j \qquad (10.10e)$$

$$w_{ij} \geq 0 \qquad \forall\ (i, j),\ i < j \qquad (10.10f)$$

$$x \in X_1,\ x \in X_3,\ x \text{ binary.} \qquad (10.10g)$$

A few remarks might be worth noting at this point. First, recall that (10.10b) implies the constraints $x \in X_2$, and hence, $x \in X_2$ is not explicitly included in (10.10). Second, one ought to preprocess QP and perform standard logical tests in order to possibly restrict the values of some binary variables to zero or one at the outset. These tests could include verifying whether or not the minimum and maximum values which each variable x_j can take over the region X are indeed zero and one respectively. Third, depending on the specific problem structure, some constraints in MIP may be omitted without affecting the continuous relaxation. For example, given that X_3 (or some subset thereof) is implied by X_1 and X_2, as is the case when X_1 consists of set partitioning constraints, Problem MIP need not include X_3 (or its corresponding subset) and need not include the associated constraints (10.10c, d, and e).

Finally, an even tighter relaxation could have been generated by considering pairwise products of the constraints defining X_2 as well. The results in this chapter do not include such additional RLT constraints, but their benefit is amply demonstrated in the context of continuous quadratic programs in Chapter 11. As shown in Adams and Sherali (1986), the stated formulation MIP is itself tighter than the aforementioned alternative linearizations in that these latter representations can be recovered via surrogates or relaxations of constraints of MIP.

10.2.2. Design of an Algorithm

The proposed algorithm follows a similar composition of strategies as discussed in Subsection 10.1.2 for Problem MIBLP. At node zero (the beginning), we perform logical tests to probe and possibly *a priori* restrict some x-variables at binary values. Thereafter, we determine $v_L = \min\{\sum_i x_i : x \in X\}$ and $v_U = \max\{\sum_i x_i : x \in X\}$ and generate the constraints

$$\lceil v_L \rceil \leq \sum_i x_i \leq \lfloor v_U \rfloor \tag{10.11}$$

where $\lceil v_L \rceil$ and $\lfloor v_U \rfloor$ are respectively the rounded-up and rounded-down values of v_L and v_U. Constraints (10.11) are added to X and are considered a part thereof henceforth; they serve not only to potentially further tighten the continuous relaxation, but they are also used in the sequel to compute knapsack-based lower bounds.

Next, we consider a Lagrangian dual of $\overline{\text{MIP}}$, the continuous relaxation of MIP, and solve this problem to near optimality using a subgradient optimization method. The motivation here is to quickly determine a good quality dual feasible solution with which to generate a (strongest surrogate) Benders' cut in the x-variables. (The direct solution of the linear program $\overline{\text{MIP}}$ may be too time consuming because of its size as demonstrated in Subsection 10.1.3.

Let us denote this cut (at node 0) as

$$z \geq \delta^0 + \sum_i \gamma_i^0 x_i. \tag{10.12}$$

Upon deriving (10.12), we record the lower bound

$$v(\overline{\text{MIP}}) \equiv \min\{\delta^0 + \sum_i \gamma_i^0 x_i : x \in X\} \tag{10.13}$$

on the optimal objective value of MIP, and use the solution to (10.13) in order to possibly obtain an incumbent solution to MIP by truncating or rounding this solution in the spirit of Balas and Martin (1980). Of course, if X possesses a special structure such as a network structure or a set covering structure, then an incumbent is more readily available through (10.13).

Next, the algorithm enters into an implicit enumeration phase. Given a certain node N being explored, with J^+, J^-, and J_f, respectively denoting the index sets of x-variables fixed at one, fixed at zero, and currently free, an attempt is made either to fathom the node or to select a branching variable to further partition and explore the reduced-node problem. This search process includes conducting logical tests, deriving one or more lower bounds as deemed judicious via updated versions of (10.12) and other generated Benders' cuts, and conducting local explorations in order to detect improved incumbent solutions.

Specifically, after creating node N by branching on some designated variable, logical tests are performed in an attempt to further restrict some x-variables at binary values. During the entire node exploration process, whenever $|J_f| \le 2$, the remaining possible completions are explicitly enumerated and the node is fathomed. Thereafter, an updated version of (10.12) at node N is generated. Denote this as

$$z \geq \delta^N + \sum_{i \in J_f} \gamma_i^N x_i. \tag{10.14}$$

Using (10.14), a trivial lower bound LB1 is first computed via the following bounded-interval knapsack linear program

$$\text{LB1} = \min\{\delta^N + \sum_{i \in J_f} \gamma_i^N x_i : \lceil v_L \rceil \leq \sum_{i \in J_f} x_i + |I^+| \leq \lfloor v_U \rfloor, \ 0 \leq x_i \leq 1 \quad \forall \ i \in J_f\}. \tag{10.15}$$

Let v^* denote the current incumbent objective value. If $(\varepsilon + \text{LB1}) \geq v^*$, where $\varepsilon > 0$ is some suitable tolerance, then the node is fathomed. Otherwise, denoting $\hat{x}_i$, for $i \in J_f$, as the computed optimal (binary) solution to (10.15), a binary vector $\tilde{x}$ is obtained by fixing $\tilde{x}_i = 1$ for $i \in J^+$, $\tilde{x}_i = 0$ for $i \in J^-$, and $\tilde{x}_i = \hat{x}_i$ for $i \in J_f$, and a Benders' cut

$$z \geq \Delta^N + \sum_{i \in J_f} \Gamma_i^N x_i \tag{10.16}$$

is generated based on $\tilde{x}$. Furthermore, in case $\tilde{x}$ is feasible to X, the incumbent solution and value are updated if the objective value of $\tilde{x}$ in Problem QP, say $\tilde{z}$ (which may be obtained by substituting $x_i = \tilde{x}_i$, $\forall \ i \in J_f$, in (10.16)) is lesser than v^*. (The current node is fathomed if $\tilde{x}$ is feasible to X and $\tilde{z} = \text{LB1}$.) Failing this, a strengthened lower bound LB2 is computed via the following problem:

$$\text{LB2} = \min\{z\text{: (10.14), (10.16), and the constraints in (10.15)}\}. \tag{10.17}$$

The motivation for (10.17) is that the constraints (10.14) and (10.16) complement each other well in that (10.16) lends strength to (10.14) in the vicinity of the solution $\tilde{x}$ at which (10.14) is the weakest, over the region defined by the constraints in (10.15). Note that LB2 may be easily computed by solving the Lagrangian dual of (10.17) obtained by dualizing (10.14) and (10.16). Since the dual variables to these constraints are nonnegative and must sum to one, the problem reduces to maximizing a concave univariate function over the constraints in (10.15). We used the tangential approximation method (see Bazaraa, Sherali and Shetty (1983)) to solve this problem, exploiting the fact that, since it is a univariate problem, only the two currently binding constraints in the outer linearization need be maintained. Generally, convergence was obtained in about four iterations. If LB2 fails to fathom node N, then the cuts (10.14) and (10.16) are surrogated using the optimal dual multipliers available through (10.17) to yield the cut, say,

$$z \geq p^N + \sum_{i \in J_f} q_i^N x_i. \tag{10.18}$$

In case the optimal dual variable associated with (10.14) in the solution to (10.17) is not unity (or else (10.17) and (10.18) would essentially be (10.15) and (10.14), respectively), Problem (10.15) is solved with its objective function replaced by the expression in (10.18). Denote by $\bar{x}$ the binary solution obtained by completing the current partial solution with the optimal (zero-one) solution to this problem. This solution is available when solving the final incumbent Lagrangian dual subproblem for (10.17). The solution $\bar{x}$ is then used to possibly update the current incumbent if it is feasible and improving. If node N is still not fathomed, then, using (10.18) along with the constraints of X, logical tests are performed to check if any further x-variables can be fixed at zero or one.

If so, the partial solution is accordingly incremented and all the above operations at node N are repeated. Otherwise, a check is made to determine if it would be worthwhile to compute a somewhat more expensive lower bound. That is, if $\bar{x}$ is infeasible in X, and if LB2 and v^* are close enough in the sense that LB2 + $0.1(p^N + \sum_{i \in J_f} q_i^N - \text{LB2}) \geq v^*$, then the linear program

$$\text{LB3} = \min\{p^N + \sum_{i \in J_f} q_i^N x_i : x \in X, x_i = 1 \text{ for } i \in J^+, x_i = 0 \text{ for } i \in J^-\} \qquad (10.19)$$

is solved to compute a potentially tighter lower bound, LB3. Let x' denote an optimal solution to (10.19). If no fathoming results from the use of LB3, then the index $i \in J_f$ for which x'_i is a maximum, with ties broken by selecting that index corresponding to the smallest value of q_i^N, is chosen as the branching variable index to be placed in J^+ upon the next selection of the active node N. Furthermore, the optimal solution x' to (10.19) may be rounded or truncated in order to possibly find an improved incumbent solution. If, however, LB3 is not computed, then a branching variable is chosen as above by using $\bar{x}$ in lieu of x'.

We found another feature to be useful in the foregoing schema. Whenever a new incumbent solution is discovered, we check the extreme points of the unit hypercube adjacent to this solution in order to possibly determine a further improving, feasible binary solution. These local explorations detected improving solutions on several occasions.

10.2.3. Computational Experience

In this subsection, we discuss our computational experience for solving Problem QP via Problem MIP. As a point of illustration, we chose to compare this performance with that of solving QP via the linearization proposed by Glover and Woolsey (1974) which in essence deletes (10.10a and b), but adds back the constraints $x \in X_2$. Let us call this representation MIP1. Except for this choice of relaxation, however, the same algorithmic strategies as proposed in Subsection 10.2.2 were employed.

As a test-bed, we chose problems having the set X represented by set covering constraints. This structure arises in an important application, the dynamic set covering problem, which seeks a sequence of facility relocations over time (see Chrissis *et al.* (1978)). Moreover, this structure lends only moderate additional strength to our linearization, and hence the results are indicative of what one might expect in no better than an average situation. No special set covering logical tests were used. Problems of size up to 25 constraints and 70 binary variables were generated for (the more difficult) instances having positive D_{ij} and negative c_i coefficients, and having a density of either 0.1 or 0.25.

Detailed results are given in Adams and Sherali (1986) — here, we provide a summary of our observations. In all cases, the time for the implicit enumeration phase (beyond the Lagrangian dual effort to obtain (10.12)) was considerably less for MIP than for MIP1. Several test cases (33%) remained unsolved using MIP1 within the set computational limit (1200 cpu seconds on an IBM 3081 Series D24 Group K computer), after enumerating 160,000-383,000 nodes. Of the remaining problems, MIP1 enumerated at an average 193 times more nodes than MIP. Moreover, for the lower density problems, the total execution time for solving Problem QP via Problem MIP was significantly smaller than the total execution time when Problem MIP1 was used, with the difference increasing dramatically with problem size. At an average, over the cases solvable by MIP1, the cpu effort was twice as much for MIP1 as compared with that for MIP. For the high density problems, by the nature of the set covering constraints, the additional strength lent to MIP over MIP1 and the *overall* benefits derived from MIP over MIP1 do not manifest themselves until the problem size increases sufficiently, although the total number of nodes enumerated is always significantly smaller for MIP.

We also attempted to evaluate the benefits of the other strategies employed in the algorithm. The Benders' cuts helped tighten both formulations, especially MIP1. Both formulations included instances that remained unsolved within the set computational limit without these cuts. Also, the linear programming bound (10.19) enhanced the performance when computed at judiciously chosen nodes as prescribed in Subsection 10.2.2. In general, the tighter the linearization, the more beneficial was the computation of this bound. Typically, this bound led to a fathoming roughly one-third of the time. Overall, the computational experience reported provides a strong indication of the benefits of strengthening the linearization through the additional RLT constraints appearing in MIP.

10.3. Miscellaneous Applications

The foregoing two sections have presented two particular applications for which RLT constructs have been gainfully employed to design successful solution algorithms. Of particular importance in this exposition is a discussion of the supporting algorithmic strategies that must be used to exploit the structure of the RLT relaxations and cope with its size, in order to enjoy the potential benefits of its relative tightness. In this section, we describe several other applications to discrete optimization problems where this type of an overall RLT-based approach has proven to be successful in designing competitive solution procedures.

As a continuation of our work in Section 10.2, the first and second level RLT relaxations have been used to develop strong lower bounds for the quadratic assignment problem. As seen in Chapter 2, because the assignment constraints are equality restrictions, these RLT relaxations are produced by simply multiplying the constraints with individual or with pairs of variables, respectively. Adams and Johnson (1994, 1996) have shown that the strategy of generating lower bounds using the first level relaxation itself subsumes a multitude of known lower bounding techniques in the literature, including a host of matrix reduction strategies. By designing a heuristic dual ascent procedure for the level-one relaxation and by incorporating dual-based cutting planes with an enumerative algorithm, an exact solution technique has been developed and tested that can competitively solve problems up to size 17. In an effort to make this algorithm generally applicable, no special exploitation of flow and/or distance symmetries was considered. As far as the strength of the RLT relaxation is concerned, on a set of standard test problems of sizes 8-20, the lower bounds produced by the dual ascent procedure uniformly dominated 12 other competing lower bounding schemes except for one problem of size 20, where an RLT procedure yielded a lower bound of 2142, while an eigenvalue-based procedure produced a lower bound of 2229, the optimum value being 2570 for this problem. Recently, Resende *et al.* (1995) have been able to solve the first level RLT relaxation exactly for problems of size up to 30 using an interior point method that employs a preconditioned conjugate gradient technique to solve the system of equations for computing the search directions. (For the aforementioned problem of size 20, the exact solution value of the lower bounding RLT relaxation turned out to be 2182, compared to the dual ascent value of 2142.) More strikingly, Hahn *et al.* (1998) have used a clever exploitation of the level-1 RLT representation to solve the size 25 instance of Nugent *et al.* (1968).

Sherali and Brown (1994) have applied RLT to the problem of assigning aircraft to gates at an airport, with the objective of minimizing passenger walking distances. The problem is modeled as a variant of the quadratic assignment problem with partial assignment and set packing constraints. The quadratic problem is then equivalently linearized by applying the first level of the RLT. In addition to simply linearizing the problem, the application of this technique generates additional constraints that provide a tighter linear programming representation. Since even the first level relaxation can get quite large, several alternative relaxations are investigated that either delete or aggregate classes of RLT constraints. All these relaxations are embedded in a heuristic that solves a sequence of such relaxations, automatically selecting at each state the tightest relaxation that can be solved within an acceptable estimated effort. Based on the solution obtained, the method fixes a suitable subset of variable to 0-1 values. This process is repeated until a feasible solution is constructed. The procedure was computationally tested using realistic data obtained from *US Airways* for problems having up to 7 gates and 36 flights. For all the test problems ranging from 4 gates and 36 flights to 7 gates and 14 flights, for which the size of the first level relaxation was manageable (having 14,494 and 4,084 constraints, respectively, for these two problem sizes), this initial relaxation itself always produced an optimal 0-1 solution.

The RLT strategy has been used to derive very effective algorithms for capacitated, multifacility, location-allocation problems that find applications in service facility or warehouse location, or manufacturing facility flow-shop design problems. Given n demand locations (customers or machines) having known respective demands, the problem is to simultaneously determine the *locations* of some m supply centers (service facilities, warehouses, interacting machines, or tooling centers) having known respective capacities, and an *allocation* of products from each source to each destination, in order to minimize total distribution costs. For the *rectilinear distance* variant of this problem that arises in applications where the flow of goods or materials occur along grids of city streets or factory aisles, the cost is directly proportional to the shipment volume and the rectilinear distance through which this shipment occurs. This problem can be equivalently reformulated as a mixed-integer, zero-one, bilinear programming problem of the general form discussed in Section 10.1. Sherali *et al.* (1994) specialized the level-one RLT based procedure for the above problem and were able to solve these difficult nonconvex problems having up to 5 sources and 20 customer locations to optimality. In addition, because of the tight relaxations obtained, this algorithm also provides an efficient heuristic which upon premature termination is capable of obtaining provably good quality solutions (with 5-10% of optimality) for larger sized problems.

Predating this work, Sherali and Adams (1984) had studied a discrete variant of the location-allocation problem in which the m capacitated service facilities are to be assigned in a one-to-one fashion to some m discrete sites in order to serve the n customers. Here, the cost per unit flow is determined by some general facility-customer separation based penalty function. This problem also turns out to have the structure of a separably constrained mixed-integer bilinear programming problem, and a partial first level RLT relaxation that includes only a subset of the constraints developed in Subsection 10.1.1, some in an aggregated form, was used to generate lower bounds. A set of 16 problems with (m, n) ranging up to (7, 50)-(11, 11) were solved using a Benders' partitioning approach. For these problem instances, even the partial, aggregated first level RLT relaxation produced lower bounds within 90-95% of optimality.

Sherali *et al.* (1998) describe another application that deals with the design of a local access and transport area network. This problem arises in the context of providing a special access service by a local exchange facility. The design involves locating suitable hub facilities having adequate multiplexing capacity, and routing the demand traffic from various clients (end offices) through their hubs, or alternatively, through a special Point-of-Presence node, in order to minimize total costs. These costs include various traffic distribution and equipment component costs, where the latter exhibit nonlinear economies of scale. Sherali *et al.* develop a model for this problem and apply RLT to construct various enhanced tightened versions of the proposed model. Efficient Lagrangian dual schemes are also devised for solving the LP relaxation of these enhanced models and to construct an effective heuristic procedure for deriving good quality solutions in this process. Extensive computational results exhibit that the proposed procedures yield solutions for the practical sized problems having an optimality gap of less than 2% with respect to the derived lower bound, within a reasonable effort that involves the solution of a single linear program.

A final application we mention is the celebrated traveling salesman problem (TSP). Specifically, in this context, Desrochers and Laporte (1991) have shown how the Miller-Tucker-Zemlin (MTZ) subtour elimination constraints can be lifted into facet defining inequalities, and hence tighten the related TSP formulation that has traditionally been considered as weak (see Nemhauser and Wolsey, 1988). Sherali and Driscoll (1997) have shown that an application of the generalized RLT procedure discussed in Chapter 3 automatically derives these lifted MTZ constraints at the first level, and moreover, provides further enhancements by subsuming additional classes of valid inequalities. For example, we applied the IP option of OSL to directly solve the test case br17 available from the TSPLIB collection. The proposed RLT formulation identified an optimum of objective value 39 after enumerating 28 nodes using OSL's branch-and-bound scheme. On the other hand, both the original MTZ formulation as well as the Desrochers-Laporte enhancement of it halted prematurely after reaching the 100,000 iteration limit, reporting incumbent solution values of 52 and 41, respectively, and having enumerated 2,214 and 2,522 nodes at termination, respectively. Such reformulations can also prove to be quite useful in solving many scheduling and vehicle routing problems where TSP-MTZ substructures are frequently used in developing models (see Desrochers and Laporte, 1991). These concepts have been extended to provide new tightened model representations for the precedence constrained traveling salesman problem. Some computational experience is provided to demonstrate the efficacy of these enhanced model formulations.

In this chapter, we have described our experience in designing specialized RLT based relaxations for solving various specific classes of discrete optimization problems. Similar experience for continuous nonconvex problems is described next in Chapter 11. We are currently investigating RLT designs for many other applications, notably, some telecommunication design and air traffic management problems, as well as the development of general purpose algorithmic strategies for solving linear mixed-integer 0-1 programming problems. Specific special structures inherent within such problems can be exploited by the RLT process as discussed in Chapter 3.

11

APPLICATIONS TO CONTINUOUS PROBLEMS

In Part II of this book, we have presented a Reformulation-Linearization/Convexification Technique for generating tight polyhedral or convex relaxations for polynomial programming problems. We have shown how the lower bounds generated by this technique can be embedded within a branch-and-bound algorithm and used in concert with a suitable partitioning procedure in order to induce infinite convergence, in general, to a global optimum for the underlying nonconvex polynomial program. In some special cases, as we shall see in this chapter, finite convergence can be obtained by exploiting inherent problem structures and characteristics. In Chapters 8 and 9, we have also presented some particular RLT strategies for generating tight relaxations for quadratic as well as for general polynomial programs and have illustrated the computational strength of the proposed procedures. In the present chapter, we complement this discussion by describing the design of RLT-based algorithms for some other special applications. The purpose of this exposition is to exhibit by way of illustration how RLT can be used for constructing such algorithms for a variety of nonconvex programming problems.

We begin in Section 11.1 by discussing the design of an algorithm to solve capacitated location-allocation problems using squared Euclidean distance based separation or penalty costs. We project this problem onto the space of the allocation-variables, transforming it into an equivalent instance of minimizing a concave quadratic function over the transportation constraints. For this equivalent representation, we devise a suitable application of the RLT concept by generating a selective set of first level bound-factor based RLT constraints. Our computational tests reveal that the bounds obtained from this relaxation are substantially superior to three different lower bounds obtained using standard techniques. The RLT-based initial linear programming relaxation itself produces solutions within 2-4% of optimality, therefore significantly enhancing the size of problems solvable by a branch-and-bound algorithm. We have solved problems having the number of sources m and the number of customers n ranging from $(m,n)= (6,120)$ – $(20,60)$ within about 150 cpu secs on an IBM 3090 computer, while the methods employing some four standard lower bounding techniques were able to handle problems of size up to only $m = 4$ and $n = 6$ within 370 cpu secs on the same computer.

In Section 11.2 we discuss an RLT approach to solve general linear complementarity problems (LCP) where the underlying defining matrix M does not possess any special property. Formulating such problems equivalently as mixed-integer bilinear programming problems, an RLT process is designed using suitable constraint-factor cross-products, and by exploiting the fact that the optimal objective function value is zero if and only if an LCP solution exists. Accordingly, an equivalent linear mixed-integer program is generated for this *continuous* problem whose linear programming solution value is *guaranteed* to match with that of the discrete problem if an LCP solution exists. Hence, the problem reduces to searching for a discrete vertex optimum to the linear programming relaxation in this case. In the reported computational results, on a total of 70 test problems using negative definite and indefinite matrices M of size up to 25×25, all problem instances except for one, were solved at the root node itself via the solution of a single linear program. For the one exception, the LP solver CPLEX quit after hitting a limit of 10,000 iterations. However, when a subgradient based Lagrangian dual approach was applied to this problem, the LCP was again solved at the root node itself. In general, although the Lagrangian dual approach was unable to attain the same tight bounds as CPLEX did due to convergence difficulties, and as a result, it sometimes led to an enumeration of 2 or 3 nodes, it was still 3-4 times faster in terms of the overall effort required as compared with the CPLEX based approach.

Finally, we conclude this chapter by discussing some miscellaneous applications in Section 11.3.

11.1. Squared Euclidean Distance Location-Allocation Problem

Location-allocation problems are concerned with determining the location and the flow allocations of a number of supply centers, given the demand requirements and locations of a set of customers, so as to minimize the total location and transportation costs. In this research we will consider a specific location-allocation problem. Given the discrete locations (a_j, b_j), $j = 1,\ldots,N$, of N customers on a continuous plane, and their associated demands d_j, $j = 1,\ldots,N$, the problem seeks to determine the locations (x_i, y_i), $i = 1,\ldots,M$, of M supply centers with known capacities s_i, $i = 1,\ldots,M$, so as to satisfy the demand requirements of the customers at minimal total cost. Hence, the decision variables are to determine the locations (x_i, y_i) of the supply centers $i = 1,\ldots,M$, and the amount of shipment w_{ij} to be sent from each supply center $i = 1,\ldots,M$ to each customer $j = 1,\ldots,N$. The objective is to minimize the total cost which is assumed to be directly proportional to the amount shipped and to the squared-Euclidean distance over which this shipment occurs. We also assume a balanced situation in which total supply is equal to the total demand. Mathematically, this problem can be formulated as follows.

$$\text{Minimize } \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij}[(x_i - a_j)^2 + (y_i - b_j)^2] \tag{11.1a}$$

$$\text{subject to } \sum_{j=1}^{N} w_{ij} = s_i, \qquad \text{for } i = 1,\ldots,M, \tag{11.1b}$$

$$\sum_{i=1}^{M} w_{ij} = d_j, \qquad \text{for } j = 1,\ldots,N, \tag{11.1c}$$

$$w_{ij} \geq 0, \quad \text{for } i = 1,\ldots,M, \ j = 1,\ldots,N. \tag{11.1d}$$

The above problem is a location-allocation problem in that for a fixed set of allocations $w \equiv (w_{ij}, i = 1,\ldots,M, \ j = 1,\ldots,N)$, the problem reduces to a pure location problem, while for a fixed set of locations (x_i, y_i), $i = 1,\ldots,M$, the problem reduces to a simple transportation/allocation problem. While these separable problems are easy to solve, the joint location-allocation problem is a difficult nonconvex programming problem, which is NP-hard even with all the demand points located along a straight line, as follows from Sherali and Nordai (1988).

However, note that if we fix the transportation flows w, the unconstrained minimum of the strictly convex objective function is readily obtained as the solution

$$x_i = \frac{\sum_{j=1}^N a_j w_{ij}}{\sum_{j=1}^N w_{ij}} = \frac{1}{s_i}\sum_{j=1}^N a_j w_{ij}, \text{ and}$$

$$y_i = \frac{\sum_{j=1}^N b_j w_{ij}}{\sum_{j=1}^N w_{ij}} = \frac{1}{s_i}\sum_{j=1}^N b_j w_{ij}, \quad \text{for } i = 1,\dots,M. \tag{11.2}$$

Substituting (11.2) into the objective function (11.1a), we can project problem (11.1) onto the space of the w variables to obtain the following equivalent representation:

$$\text{minimize}\left\{\sum_{i=1}^M \sum_{j=1}^N \frac{w_{ij}}{s_i^2}\left[\left(\sum_{k=1}^N w_{ik}a_k - s_i a_j\right)^2 + \left(\sum_{k=1}^N w_{ik}b_k - s_i b_j\right)^2\right]: \quad w \in W\right\}$$

where $w \equiv [w_{1,1}, w_{1,2}, \dots, w_{M,N}]^t$, and W is the transportation constraint set defined by (11.1b)-(11.1d). Simplifying the above objective function under $w \in W$, we can rewrite it as

$$-\sum_{i=1}^M \frac{1}{s_i}\left[\left(\sum_{k=1}^N w_{ik}a_k\right)^2 + \left(\sum_{k=1}^N w_{ik}b_k\right)^2\right] + \sum_{j=1}^N (a_j^2 + b_j^2)d_j.$$

Letting $w_i \equiv [w_{i1}, \ldots, w_{iN}]^t$, $\forall\ i = 1, \ldots, M$, $a \equiv [a_1, \ldots, a_N]^t$, and $b \equiv [b_1, \ldots, b_N]^t$, we obtain the following equivalent representation (**LAP**) of the location-allocation problem.

$$\textbf{LAP}:\ \text{Maximize} \left\{ z = \sum_{i=1}^{M} \frac{1}{s_i}\left[\left(w_i^t a\right)^2 + \left(w_i^t b\right)^2\right] \equiv \sum_{i=1}^{M} \frac{1}{s_i}\left(w_i^t T w_i\right)\right\} \tag{11.3a}$$

$$\text{subject to}\quad w \in W \tag{11.3b}$$

where

$$T \equiv aa^t + bb^t. \tag{11.3c}$$

Observe that Problem LAP is a convex maximization (or concave minimization) problem, and so, a local maximum need not be a global optimum. However, as is well known, a maximum is attained at an extreme point of W (see Bazaraa *et al.*, 1993, for example). Based on this extreme point optimality property and the nature of the transportation constraints, we propose a partitioning scheme using the dichotomy that a variable can be zero or positive at optimality. This scheme is unhampered by degeneracy effects, and it admits the use of efficient logical tests for deriving tight lower and upper bounds on the flow variables. Moreover, it induces a finitely convergent branch-and-bound process. The principal feature of the algorithm is that it employs a specialized application of RLT for deriving a bounding linear programming problem which usually yields upper bounds within 1%-3% of optimality. This linear program is further enhanced by the generation of cycle prevention inequalities derived from cut-sets defined by the forest subgraph representing a given partial solution. Only those partial solutions are maintained as active that can lead to improving feasible *extreme point* solutions of the transportation constraint set.

11.1.1. RLT-Based Relaxation

To construct the proposed RLT relaxation, consider the addition of the following constraints to (11.3) based on the shown constraint products, in order to derive an equivalent **reformulated** problem as stated below.

$$\text{Maximize } \sum_{i=1}^{M} \sum_{k=1}^{N} \sum_{\ell=k}^{N} c_{ik\ell} w_{ik} w_{i\ell} \tag{11.4a}$$

$$\text{subject to } (u_{ik} - w_{ik})(w_{i\ell} - \ell_{i\ell}) \geq 0, \quad \forall\ i = 1, \ldots, M,\ 1 \leq k \leq \ell \leq N, \tag{11.4b}$$

$$(u_{i\ell} - w_{i\ell})(w_{ik} - \ell_{ik}) \geq 0, \quad \forall\ i = 1, \ldots, M,\ 1 \leq k < \ell \leq N, \tag{11.4c}$$

$$w_{i\ell} \sum_{k=1}^{N} w_{ik} = s_i w_{i\ell}, \quad \forall\ i = 1, \ldots, M,\ \ell = 1, \ldots, N, \tag{11.4d}$$

$$(u_{ik} - w_{ik})(u_{i\ell} - w_{i\ell}) \geq 0, \quad \forall\ i = 1, \ldots, M,\ 1 \leq k \leq \ell \leq N, \tag{11.4e}$$

$$(w_{ik} - \ell_{ik})(w_{i\ell} - \ell_{i\ell}) \geq 0, \quad \forall\ i = 1, \ldots, M,\ 1 \leq k \leq \ell \leq N, \tag{11.4f}$$

$$w \in W, \tag{11.4g}$$

where from (11.3) we have,

$$c_{ik\ell} \equiv \begin{cases} (a_k^2 + b_k^2) / s_i, & \forall\ i = 1, \ldots, M, k = 1, \ldots, N, \ell = k \\ 2(a_k a_\ell + b_k b_\ell) / s_i, & \forall\ i = 1, \ldots, M, 1 \leq k < \ell \leq N \end{cases}$$

and where $\ell_{ik} \le w_{ik} \le u_{ik}$, $i = 1,\ldots,M$, $k = 1,\ldots,N$, are valid lower and upper bounds on the transportation variables. If there are no explicitly specified bounds available on w_{ik}, then we can adopt $\ell_{ik} = 0$, and $u_{ik} = \min\{s_i, d_k\}$, $\forall\ i = 1,\ldots,M$, $k = 1,\ldots,N$.

Let us now substitute

$$y_{ik\ell} = w_{ik}w_{i\ell}, \quad \forall\ i = 1,\ldots,M,\ \ 1 \le k \le \ell \le N, \tag{11.5}$$

in Problem (11.4) in order to linearize it into the following linear program LP. Notice that the additional nonlinear implied constraints incorporated in (11.4) play the role of relating the w and y variables in LP via the transformation (11.5). Note also that based on the problem structure, we have only introduced a particular subset of cross-product terms involving the w-variables as given by (11.5).

$$\textbf{LP:}\quad \text{Maximize} \quad c^t y \equiv \sum_{i=1}^{M}\sum_{k=1}^{N}\sum_{\ell=k}^{N} c_{ik\ell}y_{ik\ell} \tag{11.6a}$$

$$\text{subject to} \quad \ell_{i\ell}w_{ik} + u_{ik}w_{i\ell} - y_{ik\ell} \ge (u_{ik}\ell_{i\ell}), \quad \forall\ i = 1,\ldots,M,\ \ 1 \le k \le \ell \le N, \tag{11.6b}$$

$$u_{i\ell}w_{ik} + \ell_{ik}w_{i\ell} - y_{ik\ell} \ge (u_{i\ell}\ell_{ik}), \quad \forall\ i = 1,\ldots,M,\ \ 1 \le k < \ell \le N, \tag{11.6c}$$

$$y_{i\ell\ell} + \sum_{k=1}^{\ell-1} y_{ik\ell} + \sum_{k=\ell+1}^{N} y_{i\ell k} = s_i w_{i\ell}, \quad \forall\ i = 1,\ldots,M,\ \ \ell = 1,\ldots,N \tag{11.6d}$$

$$(u_{ik}w_{i\ell} + u_{i\ell}w_{ik}) - y_{ik\ell} \le u_{ik}u_{i\ell}, \quad \forall\ i = 1, \ldots, M,$$

$$1 \le k \le \ell \le N, \qquad (11.6e)$$

$$(\ell_{ik}w_{i\ell} + \ell_{i\ell}w_{ik}) - y_{ik\ell} \le \ell_{ik}\ell_{i\ell}, \quad \forall\ i = 1, \ldots, M,$$

$$1 \le k \le \ell \le N, \qquad (11.6f)$$

$$w \in W. \qquad (11.6g)$$

Problem LP is a relaxation of Problem LAP in that for each feasible solution w to LAP, there corresponds a feasible solution (w, y) to LP having the same objective value, where y is defined as in (11.5). However, since (11.5) does not necessarily hold for any feasible solution (w, y) to LP, the converse is not necessarily true. Hence, the solution to LP gives an *upper bound* on the value of LAP.

Problem LP has $MN(N+1)/2$ linearization variables y in addition to the MN original variables w of Problem LAP, and it has $2MN(N+1) + M + N$ constraints (other than the nonnegativity constraints). The number of constraints in LP can be reduced if some of the constraints can be shown to be ineffective. Toward this end, let $\mathbf{LP}_{\mathbf{e}}$ denote the relaxed version of Problem LP obtained by dropping the restrictions (11.6e) from the constraints of LP. For various sizes of randomly generated problems LAP, problems LP and LP_{e} were solved by using the MINOS 5.1 mathematical programming package (Murtagh and Saunders (1987)). The results indicated a very small difference (typically less than 0.1%) between the values of LP and LP_{e}, thereby implying that (11.6e) is not very effective in tightening the relaxation, while it adds $MN(N+1)/2$ constraints to the size of the problem. This is understandable because the constraints in (11.6e) generate a convex *underestimating* function, whereas a *concave envelope* is more relevant in this context. Hence, unless some y variables are forced positive by these constraints in a manner that prevents other more attractive y variables from taking on larger values, these constraints do not have a significant effect. In contrast, we found the other classes of constraints defining LP to be playing a useful role. Hence, we decided to use LP_{e} as our bounding linear program.

Furthermore, in order to conserve effort in solving LP_{e} for obtaining an upper bound, we employed a Lagrangian relaxation scheme in which the constraints (11.6c) and (11.6d) were dualized using Lagrange multipliers $\alpha \equiv (\alpha_{ik\ell},\ i = 1,\ldots,M,\ k = 1,\ldots,N-1,$ $\ell = k+1,\ldots,N)$, and $\beta \equiv (\beta_{i\ell}, i = 1,\ldots,M,\ \ell = 1,\ldots,N)$, respectively. The corresponding Lagrangian dual problem is then given by

$$\textbf{LD:}\ Z_{LD} = \text{minimum}\{\theta(\alpha,\beta):\ \alpha \geq 0,\ \beta \text{ unrestricted in sign}\}, \tag{11.7}$$

where

$$\theta(\alpha,\beta) = -\sum_{i=1}^{M}\sum_{k=1}^{N-1}\sum_{\ell=k+1}^{N} \alpha_{ik\ell}u_{i\ell}\ell_{ik} + \text{maximum}\{g(w)+h(y)\} \tag{11.8a}$$

subject to

$$\ell_{ik}w_{i\ell} + \ell_{i\ell}w_{ik} - \ell_{ik}\ell_{i\ell} \leq y_{ik\ell} \leq \ell_{i\ell}w_{ik} + u_{ik}w_{i\ell} - u_{ik}\ell_{i\ell},$$

$$\text{for } i = 1,\ldots,M,\ 1 \leq k \leq \ell \leq N, \tag{11.8b}$$

$$w \in W, \tag{11.8c}$$

and where

$$g(w) \equiv \sum_{i=1}^{M} \sum_{k=1}^{N-1} \sum_{\ell=k+1}^{N} \alpha_{ik\ell}(u_{i\ell}w_{ik} + \ell_{ik}w_{i\ell}) - \sum_{i=1}^{M} \sum_{\ell=1}^{N} s_i\beta_{i\ell}w_{i\ell}, \tag{11.8d}$$

$$h(y) \equiv \sum_{i=1}^{M} \sum_{k=1}^{N-1} \sum_{\ell=k+1}^{N} (c_{ik\ell} - \alpha_{ik\ell})y_{ik\ell}$$

$$+\sum_{i=1}^{M} \sum_{\ell=1}^{N}\left[c_{i\ell\ell}y_{i\ell\ell} + \beta_{i\ell}\left(y_{i\ell\ell} + \sum_{k=1}^{\ell-1} y_{ik\ell} + \sum_{k=\ell+1}^{N} y_{i\ell k}\right)\right]. \tag{11.8e}$$

Note that given any (α, β), we can easily evaluate $\theta(\alpha, \beta)$ as follows. We first solve for y in terms of w by putting each $y_{ik\ell}$ equal to its lower bound or its upper bound in (11.8b), according to whether its coefficient in $h(y)$ of (11.8e) is nonpositive or positive, respectively. Having obtained $y(w)$, say, in this manner, we then maximize the linear function $g(w) + h[y(w)]$ over $w \in W$ of (11.8c) by solving the associated transportation problem, and hence compute $\theta(\alpha, \beta)$.

The minimization of $\theta(\alpha, \beta)$ in (11.7) was accomplished by using the block-halving, average direction, conjugate subgradient procedure as described in Sherali and Ulular (1989), except that the step-size formula given therein was used at each iteration, and its scaling parameter was halved as prescribed within each block. The starting solution was taken as $\alpha^0 \equiv 0$ and $\beta^0 \equiv (\beta_{ik} = -c_{ikk}, \ \forall\ i = 1, ..., M, \ k = 1, \ldots, N)$. For this solution, it can be verified that $\theta(\alpha^0, \beta^0) = \sum_{k=1}^{N}(a_k^2 + b_k^2)d_k$. All other parameter values were selected as in Sherali and Ulular (1989). Note that since we only require an upper bound on the problem, we need not solve the Lagrangian dual problem exactly. Using a maximum limit of 200 iterations of the conjugate subgradient algorithm, we were able to obtain upper bounds within 0.2%-0.5% of the true LP_e optimal value, but with a reduction in effort by a factor of 15-40 over that required by the simplex-based MINOS 5.1 routine.

As a comparison, we also attempted to use three other traditional upper bounding schemes for this problem. Of these three, the following method produced the best competing results (see Sherali and Tuncbilek, 1992, for details).

Denote by T_j the jth row of T as defined by (11.3c), and compute

$$\gamma_{ij} = \max\{T_j w_i : \ w \in W\}, \ j = 1, \ldots, N, \text{ for each } i = 1, \ldots, M.$$

Then, we obtain from (11.3) that a valid upper bounding function on (11.3a) is given by

$$z \leq z_{UB}(w) = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\gamma_{ij}}{s_i} w_{ij}. \tag{11.9}$$

By maximizing $z_{UB}(w)$ over $w \in W$, we obtain an alternative upper bound z_{UB} on Problem LAP. Table 11.1 presents some comparative results for LP_e versus the foregoing upper bounding scheme. For each problem, using each of these two bounding techniques, we report the upper bound obtained, along with the value of the heuristic solution derived by applying Zoutendijk's (1960) improving feasible direction algorithm to the particular solution $w \in W$ obtained from the upper bounding scheme. (No branch-and-bound is performed in Table 11.1.) Notice that the upper bound obtained via LP_e is of a significantly better quality (typically within 1%-2% of optimality, as opposed to 14%-30% using Z_{UB}), although the heuristic solution obtained from the

Table 11.1. Comparison of upper bounding schemes.

M	N	Z_{UB}: UB	Z_{UB}: Z_{heur}	Z_{UB}: t	$Z_{UB} \div Z^*_{heur}$
3	5	22432.5	19579.5	0.00	1.14
3	20	60946.9	51203.2	0.01	1.17
4	20	87853.8	67849.1	0.02	1.28
5	12	40566.0	32568.5	0.01	1.25
5	16	60674.9	44247.3	0.01	1.37
5	20	68386.1	52568.3	0.02	1.30

M	N	Z_{LD} for LP_e of Eq. (11.7): UB	Z_{LD} for LP_e of Eq. (11.7): Z_{heur}	Z_{LD} for LP_e of Eq. (11.7): t	$Z_{LD} \div Z^*_{heur}$
3	5	20584.6	19245.3	0.40	1.05
3	20	52399.3	51711.6	3.11	1.01
4	20	69854.5	66590.5	5.72	1.02
5	12	33187.6	32670.4	2.23	1.02
5	16	45300.9	44423.1	3.70	1.02
5	20	53625.2	51231.8	5.50	1.02

Legend: UB = upper bound value, Z_{heur} = heuristic LAP objective function value obtained by starting with the upper bounding solution, Z^*_{heur} = best overall heuristic value obtained across all methods (including two other bounding schemes not shown here), t = CPU seconds for computing the upper bound using an IBM 3090 computer. (The time for obtaining Z_{heur} is uniform across the methods.)

other bounding schemes is often better. Note also that the computational effort for solving Z_{UB}, although reasonably low, is substantially more than that for obtaining the other upper bound Z_{UB}. However, as we shall see when we implement these bounds in a branch-and-bound context, the relative advantage due to the strength of the bounds derived via LP_e far outweighs its additional effort as problem size increases, in comparison with the other bounding scheme.

11.1.2. A Branch-and-Bound Enumeration Algorithm

Recall that Problem LAP is a convex maximization problem for which there exists an extreme point optimal solution. In this section, we present a branch-and-bound algorithm for Problem LAP that uses this fact in partitioning the problem based on the dichotomy that a variable is either positive or zero at an extreme point optimum, with the arcs corresponding to the positive flows forming a forest subgraph of the bipartite transportation graph. (This graph is a tree in the absence of degeneracy; see Bazaraa, Jarvis, and Sherali (1990).) Hence, a binary branch-and-bound tree is developed in a depth-first fashion (see Geoffrion, 1967), where each node represents a partial solution for which the arcs can be divided into the three disjoint sets $J^+ = \{(i,k)\colon\ w_{ik} > 0\}$, $J^0 = \{(i,k)\colon\ w_{ik} = 0\}$, and $J^F = \{(i,k)\colon\ (i,k) \notin J^+ \cup J^0\}$. We will refer to the variable w_{ik} as *positive fixed* if $(i,k) \in J^+$, as *zero fixed* if $(i,k) \in J^0$, and as a *free variable* if $(i,k) \in J^F$. To guarantee that the positive fixed variables can be made part of the set of basic variables corresponding to a basic feasible solution of W, the forest constructed by the arcs in J^+ is kept cycle free at all stages of the algorithm. Also, if any arc $(i,k) \in J^F$ has the potential of forming a cycle in the current forest, the variable w_{ik} is fixed to zero, and (i,k) is moved from the set J^F to the set J^0. Hence, whenever all the variables are fixed, *i.e.*, $J^F = \varnothing$, the resulting solution corresponds to a vertex of W.

As the algorithm progresses, we modify the lower and upper bounds on the variables w via the partitioning strategy described below. These bounds are then further tightened using logical tests based on the transportation constraints.

For computing upper bounds, given the sets J^+, J^0, and J^F at some current node of the branch-and-bound tree, we solve the Lagrangian dual of the corresponding restricted problem $\mathrm{LP_e}$ as described above. In this restriction, the imposed bounds on the flow variables w are accommodated within the set W. Also note that for all the arcs $(i,k) \in J^0$, we have $u_{ik} = \ell_{ik} = 0$, thereby implying that $w_{ik} = 0$, $y_{i\ell k} = 0$, for $\ell = 1,\dots,k$, and $y_{ik\ell} = 0$, for $\ell = k+1,\dots,N$. Therefore, for all $(i,k) \in J^0$, w_{ik} and the associated y variables are dropped from the bounding problem. Also, for all $(i,k) \in J^+$ such that $u_{ik} = \ell_{ik} (> 0)$, we substitute the following in the bounding problem:

$$w_{ik} = u_{ik},\ y_{ikk} = u_{ik}u_{ik},\ y_{ik\ell} = u_{ik}w_{i\ell},$$
$$\text{for } k < \ell \le N, \text{ and } y_{i\ell k} = u_{ik}w_{i\ell}, \text{ for } 1 \le \ell < k. \tag{11.10}$$

Consequently, the particular constraint of (11.6) that has been generated by multiplying both sides of the supply capacity constraint of source i by w_{ik} becomes redundant. Also, the constraints that contain y variables which are all fixed in value (positive or zero) become redundant and are hence dropped from the bounding problem.

Additionally, we derive a class of valid constraints based on the current forest graph represented by J^+, and include these in $\mathrm{LP_e}$, dualizing them in the Lagrangian dual LD.

Let C_p and C_q be two components of the current forest such that the cardinality of the cut set $CC_{pq} = [\{(i,k): \ i \in C_p, \ k \in C_q\} \cup \{(i,k): \ i \in C_q, \ k \in C_p\}] \cap J^F$ is at least two. For all such component pairs, we can define the following constraint based on the requirement that the arcs with positive flows should not form a cycle, and so, at most one arc in CC_{pq} can carry a positive flow.

$$\sum_{(i,k) \in CC_{pq}} w_{ik} \le u(CC_{pq}), \tag{11.11}$$

where $u(CC_{pq}) = \max\{u_{ik}: \ (i,k) \in CC_{pq}\}$. Notice that when $|CC_{pq}| = 1$, (11.11) represents a simple upper bound of the type $w_{ik} \le u_{ik}$. Hence, we can expect (11.11) to be more useful when $|CC_{pq}|$ is large. In particular, when the number of forest components decreases, (11.11) may lead to the detection of infeasibility, given that there exists no basic feasible solution completion to the current partial solution. From our computational experience, we have seen that (11.11) sometimes benefits the linear program LP_e, although often, the Lagrange multipliers associated with (11.11) remain zero. However, since generating (11.11) does not impose any significant additional computational burden, we always include this set of constraints in the problem LP_e.

At the branching step, some free variable w_{pq} is chosen, as indicated below, to further partition the problem associated with the current node of the branch-and-bound tree into two subproblems. Along one branch, w_{pq} is "positive fixed," and along the other branch, it is "zero fixed." For the branch where $w_{pq} = 0$, we explicitly set $y_{pkq} = 0$, $\forall \ k \le q$ and $y_{pqk} = 0$, $\forall \ k > q$. For the other branch where $w_{pq} > 0$, an initial lower bound of $\ell_{pq} = 1$ is used to tighten the related constraints in $\mathrm{LP_e}$. Also, the introduction of this new link in the current forest connects two components of this forest, and so, the flows on all the links in the cut-set corresponding to these two components are set to zero in order to maintain a forest in any completion of the partial solution.

Since the branching process tightens the relaxed problem $\mathrm{LP_e}$ in the above fashion, we would like to choose a branching variable that gives a tighter relaxation than other alternatives. Let $(\bar{\alpha}, \bar{\beta})$ be the incumbent solution obtained for the Lagrangian dual (11.7) to the problem $\mathrm{LP_e}$ corresponding to the current branch-and-bound node, and let $\bar{w} \in W$ be the solution obtained in (11.8) when evaluating $\theta(\bar{\alpha}, \bar{\beta})$. Using (11.6b) and (11.6c), determine an accompanying value for $\bar{y}$ according to

$$\bar{y}_{ik\ell} = \min\{\ell_{i\ell}\bar{w}_{ik} + u_{ik}\bar{w}_{i\ell} - u_{ik}\ell_{i\ell}, u_{i\ell}\bar{w}_{ik} + \ell_{ik}\bar{w}_{i\ell} - u_{i\ell}\ell_{ik}\},$$

$$\forall\ i = 1, \ldots, M,\ 1 \le k \le \ell \le N.$$

The magnitude of violation of Equation (11.5) for the components of the solution $(\bar{w}, \bar{y})$, and the contribution of this violation to the upper bound are two main factors that drive our choice for the branching variable. The following branching rule is prescribed, guided by computational experimentation. Note that this is a specialized rule that contrasts with the general recommendations given in Chapters 7 and 8.

Branching Variable Selection Rule

Step 1. Compute

$$v_1 = \max_{(i,k)\in J^F} \left\{ \sum_{\substack{\ell>k \\ (i,\ell)\in J^F \cup J^+}} c_{ik\ell} \max\{0, (\bar{y}_{ik\ell} - \bar{w}_{ik}\bar{w}_{i\ell})\} \right.$$

$$\left. + \sum_{\substack{\ell<k \\ (i,\ell)\in J^F \cup J^+}} c_{i\ell k} \max\{0, (\bar{y}_{i\ell k} - \bar{w}_{ik}\bar{w}_{i\ell})\} + c_{ikk} \max\{0, (\bar{y}_{ikk} - \bar{w}_{ik}\bar{w}_{ik})\} \right\}. \tag{11.12}$$

If $v_1 = 0$, go to Step 2. Otherwise, if $v_1 > 0$, and $(\boldsymbol{i}, \boldsymbol{k})$ uniquely evaluates (11.12), then select $(p,q) = (\boldsymbol{i}, \boldsymbol{k})$ as the branching variable indices. If $v_1 > 0$, but there is more than one arc that gives the maximum value v_1, then let $T_{IE1} \subseteq J^F$ be the set of arcs that tie in (11.12a), $|T_{IE1}| > 1$, and compute

$$v_2 = \max_{(i,k)\in T_{IE1}} \left\{ \sum_{\substack{\ell>k \\ (i,\ell)\in J^F \cup J^+}} \max\{0, (\bar{y}_{ik\ell} - \bar{w}_{ik}\bar{w}_{i\ell})\} \right.$$

$$\left. + \sum_{\substack{\ell<k \\ (i,\ell)\in J^F \cup J^+}} \max\{0, (\bar{y}_{i\ell k} - \bar{w}_{ik}\bar{w}_{i\ell})\} + \max\{0, (\bar{y}_{ikk} - \bar{w}_{ik}\bar{w}_{ik})\} \right\}. \tag{11.12b}$$

If the value v_2 is given uniquely by the arc $(\boldsymbol{i}, \boldsymbol{k})$, then select $(p,q) = (\boldsymbol{i}, \boldsymbol{k})$. If there is still a tie, then letting $T_{IE2} \subseteq T_{IE1}$ be the set of arcs that tie in (11.12b), select that branching variable which corresponds to

$$(p,q) = \arg\max\{\bar{w}_{ik} : \ (i,k) \in T_{IE2}\} \tag{11.12c}$$

with persisting ties broken arbitrarily.

Step 2. Define $\text{BR}_1 = \{(i,k,\ell)\colon \quad (i,k) \in J^F, \quad (i,\ell) \in J^F, \quad k \le \ell\}$, $\text{BR}_2 = \{(i,k,\ell)\colon \quad (i,k) \in J^+, \quad (i,\ell) \in J^F, \quad k < \ell\}$, and $\text{BR}_3 = \{(i,k,\ell)\colon$ $(i,k) \in J^F,\ (i,\ell) \in J^+,\ k < \ell\}\}$. Find $(\boldsymbol{i}, \boldsymbol{k}, \ell)$ according to

$$(\boldsymbol{i}, \boldsymbol{k}, \ell) = \arg\max\{c_{ik\ell}\bar{y}_{ik\ell}\colon\ (i,k,\ell) \in \text{BR}_1 \cup \text{BR}_2 \cup \text{BR}_3\}, \tag{11.13a}$$

with ties broken arbitrarily. If $c_{\boldsymbol{ik}\ell}\bar{y}_{\boldsymbol{ik}\ell} > 0$ then select the branching variable indices as

$$(p,q) = \begin{cases} \arg\max\{\overline{w}_{\boldsymbol{ik}}, \overline{w}_{\boldsymbol{i}\ell}\}, & \text{if } (\boldsymbol{i},\boldsymbol{k},\ell) \in \text{BR}_1, \\ (\boldsymbol{i},\ell), & \text{if } (\boldsymbol{i},\boldsymbol{k},\ell) \in \text{BR}_2, \\ (\boldsymbol{i},\boldsymbol{k}), & \text{if } (\boldsymbol{i},\boldsymbol{k},\ell) \in \text{BR}_3, \end{cases} \tag{11.13b}$$

If $c_{\boldsymbol{ik}\ell}\bar{y}_{\boldsymbol{ik}\ell} = 0$, then proceed to Step 3.

Step 3. Select the branching variable indices (p,q) as

$$(p,q) = \arg\max\{\overline{w}_{ik}\colon\ (i,k) \in J^F\}, \tag{11.14}$$

with ties broken arbitrarily.

11.1.3. Computational Results

The proposed algorithm was coded in FORTRAN and was implemented on an IBM 3090 computer. The transportation problems were solved using an improved specialized version of NETFLO, the network flow code developed by Kennington and Helgason (1980). Data for the problems were randomly generated by using the scheme described in

Sherali and Tuncbilek (1992). To avoid undue excessive computations involved in sifting through alternative optimal solutions or close to global optimal solutions, the fathoming criterion is modified as follows:

$$\text{UB} \le (1+\delta)z^*, \tag{11.15}$$

where $0 < \delta < 1$, UB is a valid upper bound at the current branch-and-bound node, and z^* is the current best (incumbent) solution value. Hence, when the algorithm stops, we can claim that the global maximum is within $100 \times \delta\%$ of the current best solution. Table 11.2(a) presents results for problems with $MN \le 45$, for which the optimality criterion δ was taken as 0.001. In Table 11.2(b) we present results for larger sized problems for which δ was taken as 0.01. Except for some difficult instances, most problems were solved with a reasonable amount of effort. However, observe from the columns of υ^0 and υ^1 that the algorithm detects near-optimal solutions early in the process, thereby prompting that it can be terminated prematurely without much loss in solution quality. Moreover, the initial linearization-based upper bound is usually within 1%-2% of the optimal solution value, except for a few cases like Problem 1 where this gap is about 5%. In fact, Table 11.2(c) gives results for the problems in Table 11.2(b) that were not solved at the initial node itself, using an optimality tolerance of $\delta = 0.015$. Note that all the problems are now solved to completion quite easily, indicating that with a slight increase in the tolerance δ, we can solve even larger sized problems.

This assertion is verified in Table 11.3 which reports results for larger sized problems for which a value of $\delta = 0.02$ was used. Other than Problem 4 for which nine nodes were generated, the remaining problems were solved at the initial node itself. Observe that the surprising low computational times for Problems 5 and 6 indicate that the Lagrangian dual problem computed good-quality upper bounds fairly quickly, leading to an early fathoming for these problems.

Finally, we report on two other computational experiments. In the first of these, we tried to assess the worth of computing more accurate upper bounds from LP_e by solving this problem via the simplex method, using the mathematical programming package MINOS 5.1 as an embedded subroutine in the branch-and-bound algorithm. Problems 1-30 of Table 11.2 (a) and 11.2 (b) were run with the stated tolerance. For most of the problems, the computational effort increased significantly. Exceptions were Problems 6, 9, and 16 which were solved in 20.16, 122.97, and 149.11 cpu seconds, after enumerating 25, 71, and 5 nodes, respectively. On the other hand, this scheme was not able to solve the Problems 27 and 30 of Table 11.2 (b), and, for example, Problems 22-26 required 166.92, 238.39, 130.43, 195.61, and 90.76 cpu seconds, respectively. Hence, this method is not suitable for solving large sized problems, even after increasing the optimality tolerance.

In the final experiment, in order to test the compromise between solution effort and the quality of upper bounds as observed in Table 11.1, we made a set of runs using only the three objective-function-based functional upper bounds of the type (11.9) at each node of

Table 11.2. Results for the branch-and-bound algorithm.

Problem	M	N	v^0	v^1	n_g	t
			(a) $\delta = 0.001$			
1	3	5	1.0514	1.0514	71	13.32
2	3	5	1.0141	1.0141	5	1.24
3	3	5	1.0275	1.0275	17	3.17
4	3	10	1.0184	1.0184	83	60.60
5	3	15	1.0118	1.0118	37	45.79
6	4	6	1.0108	1.0108	205	75.69
7	4	6	1.0038	1.0038	21	8.38
8	4	6	1.0134	1.0151	129	61.45
9	4	8	1.0212	1.0350	351	227.09
10	4	8	1.0113	1.0148	109	71.65
			(b) $\delta = 0.01$			
11	3	20	1.0133	1.0133	15	37.65
12	3	20	1.0129	1.0129	15	40.31
13	4	10	1.0121	1.0121	13	13.95
14	4	10	1.0125	1.0125	29	31.34
15	4	10	1.0126	1.0174	46	42.25
16	4	15	1.0159	1.0164	119	222.07
17	4	15	1.0099	1.0099	1	1.55
18	4	20	1.0099	1.0099	1	1.75
19	4	20	1.0100	1.0100	1	2.53
20	4	20	1.0062	1.0062	1	4.59
21	4	24	1.0158	1.0209	73	370.00+
22	5	12	1.0110	1.0158	5	9.96
23	5	12	1.0074	1.0074	1	1.81
24	5	16	1.0085	1.0085	1	3.15
25	5	16	1.0101	1.0158	17	60.08
26	5	16	1.0100	1.0100	1	3.79
27	5	16	1.0146	1.0303	41	117.88
28	5	16	1.0167	1.0167	122	370.00+
29	5	20	1.0166	1.0189	74	370.00+
30	6	14	1.0101	1.0118	23	64.22

Table 11.2. Results for the branch-and-bound algorithm (cont.).

(c) $\delta = \mathbf{0.015}$

Problem	M	N	v^0	v^1	n_g	t
11	3	20	1.0149	1.0149	1	2.29
12	3	20	1.0139	1.0139	1	2.66
13	4	10	1.0150	1.0150	1	0.68
14	4	10	1.0148	1.0148	1	0.73
15	4	10	1.0149	1.0174	2	2.46
16	4	15	1.0164	1.0164	7	14.75
21	4	24	1.0158	1.0209	18	98.02
22	5	12	1.0158	1.0158	7	11.56
25	5	16	1.0139	1.0157	2	6.14
27	5	16	1.0146	1.0303	6	20.19
28	5	16	1.0167	1.0167	7	24.82
29	5	20	1.0166	1.0189	12	61.48
30	6	14	1.0149	1.0149	1	1.40

Legend: v^0 = initial upper bound/final incumbent value, v^1 = initial upper bound/initial incumbent value, n_g = number of branch-and-bound nodes generated, t = total execution time in cpu seconds ("370+" indicates a premature termination after 370 cpu seconds).

Table 11.3. Results for the branch-and-bound algorithm with optimality tolerance $\delta = \mathbf{0.02}$.

Problem	M	N	v^0	v^1	n_g	t
1	4	20	1.0198	1.0198	1	60.4
2	4	40	1.0199	1.0199	1	10.06
3	4	45	1.0141	1.0141	1	16.45
4	5	36	1.0165	1.0210	9	133.23
5	6	30	1.0182	1.0182	1	6.39
6	6	35	1.0176	1.0176	1	7.04
7	6	35	1.0184	1.0184	1	16.00
8	6	45	1.0197	1.0197	1	19.85
9	6	45	1.0199	1.0199	1	10.36
10	6	60	1.0174	1.0174	1	32.63
11	6	75	1.0181	1.0181	1	56.36
12	6	90	1.0100	1.0200	1	72.61
13	6	120	1.0195	1.0195	1	151.34
14	7	90	1.0162	1.0162	1	108.61
15	7	110	1.0199	1.0199	1	72.57
16	8	100	1.0192	1.0192	1	158.79
17	10	80	1.0175	1.0175	1	80.23
18	15	80	1.0189	1.0189	1	33.07
19	20	60	1.0145	1.0145	1	24.99

Legend: v^0 = initial upper bound/final incumbent value, v^1 = initial upper bound/ initial incumbent value, n_g = number of branch-and-bound nodes generated, t = total execution time in cpu seconds.

---

the branch-and-bound algorithm. Accordingly, the branching variable selection rule was modified by taking $\overline{w}$ as that produced by the best performing functional upper bound, and using this at Step 3 of this procedure. We observed that Problem 2 with $(M, N) = (3, 5)$, for example, took only 0.41 seconds to solve, while Problem 8 with $(M, N) = (4, 6)$ took 56.51 seconds of execution time, and Problems 13-30 of size $(M, N) = (4, 10)$ and beyond remained unsolved within the time limit of 370 cpu seconds. Hence, while these functional bounds can be computed relatively quickly, this does not compensate for their comparative lower quality, and in an overall algorithmic scheme, the RLT-based bounds are crucial for the enhanced performance of the algorithm.

11.2. Linear Complementarity Problem

The *linear complementarity problem* (**LCP**) can be stated as follows:

Find a solution to

LCP: $Mx + q \geq 0,\ x \geq 0,\ \textit{such that}\ \ x^t(Mx + q) = 0,$ (11.16)

or determine that no such solution exists, where x, $q \in R^n$ and $M \in R^{n\times n}$.

Problems of this type arise in many scientific applications that include economic equilibrium analysis, fluid flow analysis, game theory, and mathematical programming. Details of such applications and a discussion on the vast body of literature that exists on this topic can be found in Murty (1988) and Cottle *et al.* (1992).

There are several alternative formulations of the LCP that have been exploited for developing solution procedures. The LCP has been viewed as a quadratic global optimization problem (Dorn, 1961), a stationary point problem (Eaves, 1971), a system of nonlinear equations (Mangasarian, 1977), a mixed-integer 0-1 linear programming problem (Pardalos and Rosen, 1988), and as a jointly constrained (continuous or mixed-integer) bilinear program (Al-Khayyal, 1990). In this section, we commence with the lattermost formulation, and we apply the Reformulation-Linearization Technique (RLT) to this model in order to derive a specialized, tight linear programming relaxation for the problem. This relaxation is then embedded within an enumeration scheme to solve the LCP.

11.2.1. A Reformulation of the LCP

The LCP can be equivalently stated as a mixed-integer bilinear programming problem as follows.

$$\textbf{MIBLP:}\qquad \text{Minimize}\quad y^t(Mx+q)+(e-y)^t x \tag{11.17a}$$

$$\text{subject to}\quad x \in X \equiv \{x: Mx+q \geq 0, x \geq 0\} \tag{11.17b}$$

$$y \text{ binary}, \tag{11.17c}$$

where X is assumed to be nonempty, and where e is a vector of n ones. In this formulation, y could be alternatively treated as being a continuous variable that is required to lie in the unit hypercube defined by

$$Y_H \equiv \{y: 0 \leq y \leq e\}, \tag{11.18}$$

since an optimum to MIBLP is attained at an extreme point of $X \times Y_H$. However, we will exploit the fact that y can be restricted to be binary valued in order to derive a tight linear programming relaxation of MIBLP by designing a suitable application of the RLT procedure.

Toward this end, we first augment MIBLP by (optionally) adding to it a set of implied inequalities. For example, Sherali *et al.* (1985) have prescribed a method for generating strengthened reverse polar cutting planes for solving the equivalent continuous version of

MIBLP. Actually, the presented procedure is capable of solving MIBLP while generating a single or a pair of cutting planes. However, we could use the same type of scheme to generate cuts that strengthen the standard intersection cut to any intermediate level up to the latter type of cuts. Let us assume that a sequence of T such valid cuts have been pre-generated, and let us accommodate these cuts within MIBLP. Defining this set of cuts as

$$Y_C = \{y: \sum_{i=1}^{n} \alpha_{it} y_i \geq \beta_t \text{ for } t = 1, \ldots, T\}, \tag{11.19}$$

and letting M_{ij} denote the (i, j)th element of M, we can re-write MIBLP as follows.

MIBLP: Minimize $$q^t y + \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} y_i x_j + (e - y)^t x \tag{11.20a}$$

subject to $$\sum_{j=1}^{n} M_{kj} x_j + q_k \geq 0 \text{ for } k = 1, \ldots, n \tag{11.20b}$$

$$x \geq 0,\ y \in Y_H \cap Y_C,\ y \text{ binary}. \tag{11.20c}$$

We now present below our proposed specialized design for applying the RLT procedure to problem MIBLP, in order to derive a suitable, equivalent linear mixed-integer 0-1 programming formulation (MIP).

Reformulation Phase

Let us add to MIBLP the following sets of valid constraints:

(i) $2n^2$ constraints obtained by multiplying (11.20b) by y_i, and $(1 - y_i)$ for $i = 1, \ldots, n$.

(ii) nT constraints obtained by multiplying (11.19) by x_j, for $j = 1, \ldots, n$.

(iii) nT constraints obtained by multiplying the constraints in (11.19) by those in (11.20b).

(iv) $2n^2$ simple and variable bounding constraints obtained by multiplying each $x_j \geq 0$ by y_i and $(1 - y_i)$, for $i, j = 1, \ldots, n$.

Linearization Phase

Incorporate the new constraints defined by (i) - (iv) into MIBLP and substitute

$$w_{ij} = y_i x_j \text{ for } i, j = 1, \ldots, n. \tag{11.21}$$

Furthermore, observe that if there exists a solution to LCP, then noting (11.17a), there exists a solution to MIBLP such that

$$(1 - y_j)x_j = 0 \ \forall \ j = 1, \ldots, n, \textit{ i.e. } \ w_{jj} = x_j \ \ \forall \ j = 1, \ldots, n. \tag{11.22}$$

Substituting (11.21) and (11.22) in the reformulated problem, we derive the following 0-1 mixed-integer linear program MIP. (Note that in the objective function (11.20a), we have used the fact that under (11.22), $(e - y)^t x \equiv 0$.)

$$\textbf{MIP:} \quad \text{Minimize } q^t y + \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} w_{ij} \tag{11.23a}$$

subject to

$$\sum_j M_{kj} w_{ij} + q_k y_i \geq 0 \;\; \forall\; (i, k) \tag{11.23b}$$

$$\sum_j M_{kj} x_j + q_k \geq \sum_j M_{kj} w_{ij} + q_k y_i \;\; \forall\; (i, k) \tag{11.23c}$$

$$\sum_j \alpha_{it} w_{ij} \geq \beta_t x_j \;\; \forall\; (t, j) \tag{11.23d}$$

$$\sum_i \sum_j \alpha_{it} M_{kj} w_{ij} + q_k \sum_i \alpha_{it} y_i - \beta_t \sum_j M_{kj} x_j \geq \beta_t q_k \;\; \forall\; (k, t) \tag{11.23e}$$

$$0 \leq w_{ij} \leq x_j \;\; \forall\; (i, j), \text{ with } w_{jj} = x_j \;\; \forall\; j \tag{11.23f}$$

$$y \in Y_C \cap Y_H, y \text{ binary.} \tag{11.23g}$$

Remark 11.1. Note that similar to the argument used for (11.22), we could have also required the product constraint generated by multiplying each y_i with the i^{th} constraint in (11.20b) to be an equality, which must hold true if an LCP solution exists. In other words, for $i = k$ in (11.23b) we could have written this constraint as an equality as follows, while retaining (11.23c) only for all $i \neq k$.

$$\sum_j M_{kj} w_{kj} + q_k y_k = 0 \;\; \forall\; k. \tag{11.24}$$

However, (11.24) would make the objective function in (11.23a) identically zero, thereby reducing the problem to one of simply finding a feasible solution to the resulting problem (11.23). If this feasibility problem is posed as a "Phase I" optimization problem of minimizing the sum $\sum_{k=1}^{\cdot\cdot} x_{ak}$ of artificial variables, where each artificial variable x_{ak} is equated to the expression in (11.24) within the constraints, then this optimization problem is readily seen to be equivalent to MIP itself. Hence, we do not enforce this restriction in analogy with (11.22). □

Before proceeding with a further analysis of (11.23), we remark here that observing (i) and (ii), by symmetry, one might be tempted to generate additional RLT constraints by deriving upper bounds u_j on the variables x_j over X, when they exist, and using the corresponding nonnegative factors $(u_j - x_j)\ \forall\ j$, to multiply the constraints defining Y_H and Y_C. However, similar to Proposition 8.2, this can be shown to be unnecessary since the constraints that have already been generated actually imply such constraints.

Under the assumption that X is bounded, following Chapter 2, we can assert that Problem MIP is equivalent to LCP in the sense that LCP has a solution if and only if MIP has an optimum of objective value equal to zero. Moreover, any optimum to MIP having an objective value of zero corresponds to an LCP solution. In particular, if we define MIBLP′ as the problem MIBLP under the added restriction that $(e - y)^t x = 0$, then MIBLP′ and MIP are equivalent in the sense that for any feasible solution to MIBLP′ there corresponds a feasible solution to MIP having the same objective value, and vice versa.

To treat the case of an unbounded set X, we can incorporate an additional constraint

$$\sum_{j=1}^{n} x_j \leq U \tag{11.25}$$

into MIBLP, where U is large enough so as to preserve all the extreme points of X. Recognizing that if an LCP solution exists, then it occurs at an extreme point of X, we will continue to have an equivalent problem under (11.25). For constructing the corresponding problem MIP, we generate $2n$ additional constraints obtained by multiplying (11.25) by each y_i, and $(1 - y_i)$ for $i = 1, \dots, n$. This produces the $2n$ restrictions given by

$$\left(U - \sum_j x_j\right) \geq \left(Uy_i - \sum_j w_{ij}\right) \geq 0 \;\; \forall\; i = 1, \dots, n. \tag{11.26}$$

Incorporating these constraints into MIP, the aforementioned equivalence of MIP to LCP continues to hold true. Note that in practice, if U is too large, then this can adversely affect the tightness of the linear programming relaxation of MIP. Hence, if a reasonable value of U cannot be estimated, then it might be worthwhile to consider partitioned subproblems of MIP in which $\sum_j x_j$ is required to be within specified ranges.

Remark 11.2. Observe from (11.23b) written for all $i = k$, that the objective value of $\overline{\text{MIP}}$, the continuous relaxation of MIP, is lower bounded by zero. Furthermore, if LCP has a solution, then this solution is readily seen to be feasible to $\overline{\text{MIP}}$ and to yield an objective value of zero. Hence, if an LCP solution exists, then $\overline{\text{MIP}}$ will have an objective value of zero, and moreover, the set of LCP solutions will be alternative optimal solutions to $\overline{\text{MIP}}$. This construction therefore provides the facility to generate

LCP solutions *via a single linear program possibly*, with enumeration required to search among alternative optimal solutions to $\overline{\text{MIP}}$ in case an LCP solution is not directly obtained via the initial optimum. □

11.2.2. Proposed Algorithm

The proposed algorithm is basically an implicit enumeration scheme. The enumeration tree is constructed using a combination of depth-first and best-first approaches as in Sherali and Meyers (1986). In this approach, a maximum number (MAXACT) of active tree end nodes is specified, and a best-first scheme is adopted so long as the number of active nodes is less than MAXACT, before reverting to a depth-first strategy. When selecting an active node to explore, the one having the least lower bound is selected. In case the number of active nodes is less than MAXACT, then a breadth-wise extension of the enumeration tree is performed at a node that is closest to the root node on the path connecting the root node to this selected node, and for which both descendants have not as yet been generated. Otherwise, the selected node is explored in a depth-first fashion.

At each node of the enumeration tree in this scheme, we would have a partial solution for which some y variables are fixed at 0 or 1, while others are free. Denote

$$I_0 = \{i\colon\ y_i \text{ is fixed at } 0\},\quad I_+ = \{i\colon\ y_i \text{ is fixed at } 1\},\quad I = \{i\colon\ y_i \text{ is free}\}. \tag{11.27}$$

Examining MIBLP, since we are interested in zero objective values, we can therefore restrict

$$x_i = 0\ \forall\ i \in I_0, \text{ and } M_{i\cdot}x + q_i = 0\ \forall\ i \in I_+. \tag{11.28}$$

Hence, using (11.28), Equation (11.20b) reduces to

$$\sum_{j \notin I_0} M_{kj} x_j + q_k \geq 0 \ \forall \ k \notin I_+, \text{ and } \sum_{j \notin I_0} M_{kj} x_j + q_k = 0 \ \forall \ k \in I_+, \tag{11.29}$$

and similarly, (11.19) reduces to the set of constraints

$$\sum_{i \in I} \alpha_{it} y_i \geq \overline{\beta}_t \equiv \beta_t - \sum_{i \in I_+} \alpha_{it} \ \forall \ t = 1, \ldots, T. \tag{11.30}$$

Applying RLT to this reduced problem in a similar fashion to the construction of MIP above, we obtain the following node subproblem. (Note that MIP $\equiv$ MIP$(\emptyset, \emptyset, N)$ where $N \equiv \{1, \ldots, n\}$.)

MIP(I_0, I_+, I):

$$\text{Minimize} \quad \sum_{i \in I} q_i y_i + \sum_{i \in I} \sum_{j \notin I_0} M_{ij} w_{ij} \tag{11.31a}$$

subject to

$$\sum_{j \notin I_0} M_{kj} w_{ij} + q_k y_i \geq 0 \ \forall \ k \notin I_+, i \in I \tag{11.31b}$$

$$\sum_{j \notin I_0} M_{kj} x_j + q_k \geq \sum_{j \notin I_0} M_{kj} w_{ij} + q_k y_i \ \forall \ k \notin I_+, i \in I \tag{11.31c}$$

$$\sum_{j \notin I_0} M_{kj} x_j + q_k = 0 \ \forall \ k \in I_+ \tag{11.31d}$$

$$\sum_{j \notin I_0} M_{kj} w_{ij} + q_k y_i = 0 \ \forall \ k \in I_+, i \in I \tag{11.31e}$$

$$\sum_{i \in I} \alpha_{it} y_i \geq \overline{\beta}_t \ \forall \ t = 1, \ldots, T \tag{11.31f}$$

$$\sum_{i \in I} \alpha_{it} w_{ij} \geq \overline{\beta}_t x_j \quad \forall\ t = 1, \ldots, T, j \notin I_0 \tag{11.31g}$$

$$\sum_{i \in I} \sum_{j \notin I_0} \alpha_{it} M_{kj} w_{ij} + q_k \sum_{i \in I} \alpha_{it} y_i - \overline{\beta}_t \sum_{j \notin I_0} M_{kj} x_j \geq \overline{\beta}_t q_k \quad \forall\ t, k \notin I_+ \tag{11.31h}$$

$$\left(U - \sum_{j \notin I_0} x_j \right) \geq \left(U y_i - \sum_{j \notin I_0} w_{ij} \right) \geq 0 \quad \forall\ i \in I \text{ (if } X \text{ is unbounded)} \tag{11.31i}$$

$$0 \leq w_{ij} \leq x_j \quad \forall\ i \in I, j \notin I_0 \text{ with } w_{jj} = x_j \quad \forall\ j \in I \tag{11.31j}$$

$$y \in Y_H, y \text{ binary.} \tag{11.31k}$$

Let us now denote the linear programming relaxation of any node problem MIP $(\cdot)$ by $\overline{\text{MIP}}\ (\cdot)$. Our node strategy is as follows. First we perform standard logical tests on the constraints of Y_C (given by (11.19) or (11.31f)) in order to possibly fix certain y-variables at zero or one. Then, using the updated sets I_0, I_+, and I as defined in (11.27), we construct the mixed-integer program $\text{MIP}(I_0, I_+, I)$. We next solve the linear program $\overline{\text{MIP}}(I_0, I_+, I)$ in order to derive a lower bound on the problem. If this bound is positive, then we fathom the current node. Otherwise, using the solution $(\overline{x}, \overline{y})$ obtained for $\overline{\text{MIP}}\ (\cdot)$, we execute the heuristic described in Sherali *et al.* (1995) in order to possibly detect an LCP solution. If this is successful, then the algorithm terminates. Otherwise, if T is less than some limit $T_{\max}$, we generate a reverse polar cut as described in Sherali *et al.* (1995), and append this to the set Y_C. We also identify a suitable branching variable index for partitioning this node whenever it is selected for this purpose and then proceed to the top of the loop at which an active node having the least lower bound is selected for further exploration. Details of some relevant steps mentioned in this algorithmic description are discussed below.

Solution to $\overline{\textbf{MIP}}(\cdot)$

The problem $\overline{\text{MIP}}$ has $3n^2 + 2nT - n$ constraints and $n^2 + 2n$ variables and hence can be very large, even for a moderately sized LCP. Solving $\overline{\text{MIP}}$ $(\cdot)$ using a straightforward simplex based approach or even an interior point approach, can become prohibitively expensive, especially in a branch-and-bound scheme (see the computational experience in Adams and Sherali, 1993). Hence, in order to effectively cope with the size of $\overline{\text{MIP}}$, we recommend a Lagrangian relaxation approach, by solving (perhaps inexactly) the linear program $\overline{\text{MIP}}$ via a suitable Lagrangian dual LD. In order to describe this dual formulation, let $U = \max\{\sum_j x_j : x \in X\}$. If X is unbounded $(U = \infty)$, then artificially bound it using a suitable value of U as suggested for (11.25) above, and incorporate this constraint within X. Subsequently, determine $u_j = \max\{x_j : x \in X\}$ for $j = 1, \ldots, n$. Note that these bound computations are performed only once at the beginning of the algorithm. (Of course, during the course of solving these linear programs, if an LCP solution is detected, we can terminate the entire procedure.)

For convenience in notation, given any node subproblem defined by (I_0, I_+, I), let us denote ξ as the vector of x and w variables in (11.31), and y as the vector of the variables $(y_i, i \in I)$. Accordingly, let us write $\overline{\text{MIP}}(I_0, I_+, I)$ equivalently as follows.

$$\overline{\textbf{MIP}}(I_0, I_+, I): \quad \text{Minimize} \quad q^t y + c^t \xi \tag{11.32a}$$

$$\text{subject to} \quad A\xi + Dy \geq b \tag{11.32b}$$

$$G\xi + Hy = f \tag{11.32c}$$

$$(\xi, y) \in \Lambda \tag{11.32d}$$

where q is now assumed to have components $(q_i, i \in I)$, the constraints (11.32b) and (11.32c) include the constraints (11.31b-i), the terms w_{jj} have all been replaced by $x_j \ \forall\ j \in I$ as in (11.31j), and where

$$\Lambda = \{(\xi, y)\colon 0 \leq w_{ij} \leq x_j \ \forall\ i \in I,\ j \notin I_0,\ j \neq i,$$

$$0 \leq x_j \leq u_j y_j \ \forall\ j \notin I_0,$$

$$\text{where } y_j \equiv 1 \text{ if } j \in I_+),\ 0 \leq y_i \leq 1 \ \forall\ i \in I\}. \tag{11.33}$$

The Lagrangian dual of (11.32) can hence be formulated as

$$\textbf{LD:} \quad \max\{\theta(\pi, \mu)\colon \pi \geq 0,\ \mu \text{ restricted}\} \tag{11.34}$$

where

$$\theta(\pi, \mu) = \min\{q^t y + c^t \xi + \pi^t (b - A\xi - Dy) +$$

$$\mu^t (f - G\xi - Hy)\colon (\xi, y) \in \Lambda\}. \tag{11.35}$$

At some iteration k, given $(\pi, \mu) = (\pi^k, \mu^k)$, let (ξ^k, y^k) solve (11.35). Then a subgradient of θ at (π^k, μ^k) is given by

$$\delta^k = \begin{pmatrix} b - A\xi^k - Dy^k \\ f - G\xi^k - Hy^k \end{pmatrix}. \tag{11.36}$$

Also, given (π^k, μ^k), (11.35) can be easily solved by first solving for w in terms of x, by noting the signs of the coefficients of w in the objective function in (11.35), setting $w_{ij} = x_j$ if this coefficient for w_{ij} is negative, and setting $w_{ij} = 0$ otherwise, for all defined i, j. Similarly, after accordingly revising the coefficients of the x variables, we can next solve for x in terms of y, and finally solve for y itself over the simple bounding constraints $0 \le y_i \le 1 \quad \forall\ i \in I$, hence recovering an optimum to (11.35) by backtracking through the results of this process. The Lagrangian dual problem LD given by (11.34) is a nondifferentiable optimization problem. To solve this problem, we adopted the conjugate subgradient deflection scheme of Sherali and Ulular (1989), which uses an average direction subgradient deflection strategy together with a block-halving step-length strategy, (see also Bazaraa *et al.* (1993)).

Note that the foregoing subgradient optimization scheme to solve the Lagrangian dual LD does not produce a primal solution that can be used to generate heuristic solutions or to implement branching decisions. However, Sherali and Choi (1994) have presented some generalized strategies for recovering primal solutions within the context of pure and deflected subgradient methods following the pioneering work of Shor (1985) on this topic. Based on these results, we can adopt the following strategy. Denote (x^k, y^k) to be (part of) an optimal solution obtained in (11.35) when evaluating $\theta(\pi^k, \mu^k)$. Then for the final 50-100 iterations of the subgradient algorithm used to solve LD, (where such an algorithm is usually run up to a specified maximum limit number of iterations), we begin with $K = 1$ and with $(\bar{x}^1, \bar{y}^1)$ as the subproblem solution (x^p, y^p) obtained say, for some iteration $k = p$. Then, given $(\bar{x}^K, \bar{y}^K)$ for $K \geq 1$, we update this solution as follows.

$$(\bar{x}^{K+1}, \bar{y}^{K+1}) = \frac{K}{K+1}(\bar{x}^K, \bar{y}^K) + \frac{1}{K+1}(x^{p+K}, y^{p+K})$$

$$\text{for } K = 1, 2, \ldots \tag{11.37}$$

Let $(\bar{x}, \bar{y})$ be the final solution obtained in this manner upon termination.

Strongest Surrogate Constraint

Upon termination of the foregoing subgradient optimization algorithm for solving LD, let (π^*, μ^*) be the incumbent solution. Consider the subproblem (11.35) for $(\pi, \mu) = (\pi^*, \mu^*)$, and let the final stage of solving this subproblem when we obtain an objective expression only in terms of y yield the problem

$$\text{minimize } \{\sum_{i \in I} \gamma_i y_i - \gamma_0 : 0 \leq y_i \leq 1 \ \forall \ i \in I\}. \tag{11.38}$$

Hence, since we are not interested in an objective value for $\overline{\text{MIP}}$ that exceeds zero, we can impose the constraint

$$\sum_{i \in I} \gamma_i y_i \leq \gamma_0. \tag{11.39}$$

Note that (11.39) can be viewed as a surrogate constraint obtained by surrogating $q^t y + c^t \xi \leq 0$ based on the objective function (11.32) with the constraints of (11.32) except for the constraints $0 \leq y_i \leq 1 \ \forall \ i \in I$, using the Lagrange multipliers corresponding to the incumbent dual solution. Since (11.38) yields the incumbent

Lagrangian dual value, we refer to (11.39) as a *strongest surrogate constraint*, following standard terminology (see Rardin and Unger, 1976).

We can now perform logical tests on this constraint in order to possibly fix additional y variables at zero or one for this node and its descendants.

Branching Variable Selection Strategy

Note that the coefficients of y in (11.39) correspond to the reduced cost coefficients at the dual (near) optimal solution (π^*, μ^*). Since the basic y variables that would have the tendency to be fractional at an optimum for $\overline{\text{MIP}}$ would have zero reduced costs, and since we would like to resolve any fractionality resulting from a linear programming optimum (for which $\bar{y}$ obtained via (11.37) serves as an estimate), we implemented the following branching strategy.

Branching Rule. Select a variable that has the highest fractionality in the solution $\bar{y}$ obtained via (11.37) as the branching variable, breaking ties by favoring the variable that has the least absolute coefficient in (11.39).

11.2.3. Computational Results

In this section we present some computational experience with the RLT based implicit enumeration algorithm. For conducting this study, the constraint matrix M and the constant vector q were generated as prescribed by Pardalos and Rosen (1988). Two types of test matrices were considered:

1. The matrix M is negative-definite. The feasible region X is automatically bounded in this case (see Pardalos and Rosen, 1987).

2. The matrix M is indefinite with the feasible region being bounded.

The procedure for generating test problems having such matrices M and that also admit a known LCP solution is described in Sherali *et al.*, 1996. The proposed algorithm was implemented in C and FORTRAN. The XMP subroutine package of Marsten (1981) was used in the generation of the intersection cut and in the heuristic search in the branch-and-bound tree. We used the CPLEX callable library to solve the node 0 problem in RLT-CPLEX-LD (see below). All the runs were made on a Sun Sparcstation, Model 10 running Solaris 2.3.

The code was tested on matrices M of sizes n = 10, 15, 20, and 25. A total of seventy problems were generated. The zero tolerance for detecting an LCP solution was set to 0.001. The maximum number of subgradient iterations for solving the Lagrangian dual problem at the root node was set at 1500, while it was set to 1000 iterations at the other branching nodes. To define the initial set Y_C (Equation (11.19)), we generated a pair of cuts as in Sherali *et al.* (1995), but used only the ordinary intersection (non-strengthened) version of these cuts. Also, as an alternative method to solve MIP, we used the simplex package CPLEX to solve $\overline{\text{MIP}}$ at the root node and then used the deflected subgradient scheme to compute lower bounds at subsequent nodes, via the suggested Lagrangian dual formulation.

The results obtained are presented in Tables 11.4 - 11.7. The table headings used are defined as follows.

RLT-LD: This version uses the deflected subgradient algorithm to solve the proposed Lagrangian dual formulation of $\overline{\text{MIP}}$.

RLT-CPLEX-LD: This version uses the simplex package CPLEX to solve $\overline{\text{MIP}}$ at node 0 and then uses the deflected subgradient algorithm on the Lagrangian dual at subsequent nodes.

v**:** Objective value obtained for $\overline{\text{MIP}}$ at node 0 via either CPLEX or LD. (Theoretically, this should be zero as discussed in Remark 11.2.)

E: Number of nodes explored in the branch-and-bound process.

Time: Total CPU time elapsed in seconds.

Itr.: Number of simplex iterations taken by CPLEX for the node 0 subproblem, where applicable.

From Tables 11.4 and 11.6 we can see that using only the Lagrangian dual approach (RLT-LD) yields a better performance than using the RLT-CPLEX-LD method. Also, all the solutions were recovered at node 0 itself for the RLT-CPLEX-LD version (see Remark 11.2). However, because of the convergence difficulties experienced by the subgradient approach, some enumeration was required by RLT-LD. As the size of the problem increases, we can see that despite this additional enumeration, the RLT-LD method is about 3-4 times faster than the RLT-CPLEX-LD method.

For the indefinite case (Tables 11.5 and 11.7), for all but one problem of size 25, an LCP solution was obtained at the root node 0 itself using the RLT-CPLEX-LD method. (The one exception was actually terminated by CPLEX at node 0 after exceeding a maximum iteration limit of 10,000.) In comparison, the RLT-LD method performs quite well computationally, detecting an LCP solution within the first few nodes (never more than 5) of the branch-and-bound tree with a significantly reduced CPU effort.

Although there is a loss in the tightness of the lower bounds at the node 0 problem while using the Lagrangian dual approach, there is a significant improvement in the computational efficiency of the algorithm. (There is still a potential for further improvement by enhancing the Lagrangian dual scheme.) On the other hand, the use of CPLEX is not recommended for solving larger sized LCPs. One reason for the increased effort for the RLT-CPLEX-LD method is due to the ill-conditioning of the basis matrix as the size of the problem increases. We noticed that the condition number of the final basis matrix (scaled) was between 10^6 - 10^9 for problem sizes greater than 20.

The average time taken by a problem of size 20 by our algorithm running on a Sun workstation was 16.5 seconds using the proposed RLT-LD strategy. Moreover, a *single linear program* invariably obtained an LCP solution for all problems, and additional nodes

Table 11.4. Runs for negative definite M for n = 10-20.

		RLT-LD			RLT-CPLEX-LD			
Num.	n	v	E	Time	v	E	Time	Itr.
1	10	0.0	—	0.1	0.0	—	0.1	—
2	10	0.0	—	0.1	0.0	—	0.1	—
3	10	0.0	—	0.1	0.0	—	0.1	—
4	10	0.0	—	0.2	0.0	—	0.2	—
5	10	0.0	—	0.1	0.0	—	0.1	—
6	10	0.0	—	0.1	0.0	—	0.1	—
7	10	0.0	—	0.1	0.0	—	0.1	—
8	10	0.0	—	0.1	0.0	—	0.1	—
9	10	0.0	—	0.2	0.0	—	0.2	—
10	10	0.0	—	0.1	0.0	—	0.1	—
11	15	–5.36	1	5.8	0.0	1	10.3	516
12	15	–9.5	1	5.7	0.0	1	25.4	1261
13	15	0.0	—	0.1	0.0	—	0.1	—
14	15	–5.8	1	5.8	0.0	1	6.5	316
15	15	–6.40	1	6.4	0.0	1	8.6	433
16	15	0.0	—	0.1	0.0	—	0.1	—
17	15	–67.5	1	5.7	0.0	1	7.1	365
18	15	0.0	—	0.1	0.0	—	0.1	—
19	15	–7.51	1	5.8	0.0	1	8.6	432
20	15	0.0	1	0.2	0.0	0	0.2	—
21	20	–20.19	1	12.6	0.0	1	18.8	471
22	20	0.0	—	0.3	0.0	—	0.3	—
23	20	–19.23	1	12.8	0.0	1	56.9	1561
24	20	–12.59	1	13.8	0.0	1	77.3	1983
25	20	0.0	1	2.2	0.0	1	69.4	1825
26	20	–23.66	1	12.8	0.0	1	57.2	1499
27	20	0.0	—	0.2	0.0	—	0.2	—
28	20	0.0	1	3.5	0.0	1	66.8	1914
29	20	0.0	—	0.3	0.0	—	0.3	—
30	20	–27.14	1	12.4	0.0	1	73.0	1774

Table 11.5. Runs for indefinite M for n = 10-20.

		RLT-LD			RLT-CPLEX-LD			
Num.	n	v	E	Time	v	E	Time	Itr.
1	10	0.0	—	0.1	0.0	—	0.1	—
2	10	0.0	—	0.1	0.0	—	0.2	—
3	10	–6.13	1	2.1	0.0	1	1.6	160
4	10	–18.89	1	2.5	0.0	1	2.6	316
5	10	–7.49	5	4.2	0.0	1	3.9	255
6	10	–12.21	2	2.9	0.0	1	1.9	239
7	10	0.0	—	0.1	0.0	—	0.1	—
8	10	0.0	—	0.1	0.0	—	0.1	—
9	10	–15.53	4	4.6	0.0	1	3.3	230
10	10	0.0	—	0.1	0.0	—	0.1	—
11	15	–7.98	5	12.6	0.0	1	26.6	1335
12	15	–56.45	1	5.8	0.0	1	25.7	1298
13	15	–22.20	4	28.0	0.0	1	10.4	539
14	15	–56.72	2	27.1	0.0	1	15.7	659
15	15	–15.60	1	5.8	0.0	1	22.0	1193
16	15	–55.20	3	10.4	0.0	1	16.4	875
17	15	–58.04	1	5.8	0.0	1	21.9	1242
18	15	–40.37	3	9.3	0.0	1	13.4	653
19	15	–11.51	2	13.1	0.0	1	21.9	1173
20	15	–18.38	1	5.9	0.0	1	24.2	1233
21	20	0.0	—	0.2	0.0	—	0.1	—
22	20	–41.1	1	15.9	0.0	1	91.1	2413
23	20	–46.53	1	20.5	0.0	1	79.2	1786
24	20	–33.9	2	30.8	0.0	1	76.7	1899
25	20	–24.6	1	12.1	0.0	1	136.4	3502
26	20	–27.35	3	18.8	0.0	1	109.7	2757
27	20	0.0	—	0.6	0.0	—	0.6	—
28	20	–107.3	2	18.7	0.0	1	87.8	2123
29	20	0.0	—	0.7	0.0	—	0.7	—
30	20	–73.1	2	46.4	0.0	1	106.0	2649

Table 11.6. Runs for negative definite M for n = 25.

		RLT-LD			RLT-CPLEX-LD			
Num.	n	v	E	Time	v	E	Time	Itr.
1	25	–17.7	1	29.0	0.0	1	206.5	2928
2	25	–43.6	1	30.3	0.0	1	189.0	2433
3	25	–44.42	1	28.9	0.0	1	268.6	3926
4	25	–0.72	1	2.3	0.0	1	169.1	2435
5	25	–23.57	1	29.2	0.0	1	183.2	2576

Table 11.7. Runs for indefinite M for n = 25.

		RLT-LD			RLT-CPLEX-LD			
Num.	n	v	E	Time	v	E	Time	Itr.
1	25	–59.7	1	74.1	—	—	—	>10000
2	25	–45.6	2	143.6	0.0	1	296.4	3958
3	25	–157.1	2	150.4	0.0	1	181.6	2458
4	25	–11.72	3	60.3	0.0	1	302.6	4235
5	25	–120.49	1	109.2	0.0	1	212.9	2890

(≤ 5) needed to be generated only when this linear program was solved approximately via a Lagrangian dual approach.

In conclusion, we suggest some recommendations that might lead to further improvements in the proposed algorithms. One such avenue lies in the application of RLT to MIBLP. For example, we could construct certain additional pairwise product RLT constraints of the type

$$\left(\sum_{i=1}^{n} \alpha_{ik} y_i - \beta_k\right)\left(\sum_{i=1}^{n} \alpha_{ij} y_i - \beta_j\right) \geq 0 \quad \forall\ k, j = 1, \ldots, T \tag{11.40a}$$

$$\left(\sum_{i=1}^{n} \alpha_{ik} y_i - \beta_k\right) y_j \geq 0 \quad \forall\ k = 1, \ldots, T, j = 1, \ldots, n \tag{11.40b}$$

$$\left(\sum_{i=1}^{n} \alpha_{ik} y_i - \beta_k\right)(1 - y_i) \geq 0 \quad \forall\ k = 1, \ldots, T, j = 1, \ldots, n \tag{11.40c}$$

and use the substitution $z_{ij} = y_i y_j \ \forall\ i < j$, and $y_i = y_i^2 \ \forall\ i$. Incorporating (11.40) into MIP, introduces an additional $n^2 - n$ new variables and $2nT + T^2$ new constraints, but attempts to better approximate the convex hull of $Y_C \cap Y_H$, and might yield a tighter linear programming representation and hence lead to enumerating fewer nodes in the branch-and-bound tree (see Remark 11.2). Also, the generation of forming higher-order product constraints that might prove to be beneficial in producing tighter relaxations could be investigated. Notwithstanding these foregoing comments, since the current relaxation itself appears to be very tight, a more fruitful line of research might be to enhance the performance of the solution scheme used for optimizing the Lagrangian dual formulation. This can have a significant potential impact on further improving the overall efficiency of the algorithm.

11.3. Miscellaneous Applications

In the foregoing two sections, we have provided details for generating RLT relaxations and designing suitable algorithms along with useful implementation strategies for some particular classes of problems. Several other such applications of RLT have been conducted, and we briefly mention some of these below.

In Section 11.1, we described the use of RLT for solving capacitated location-allocation problems where the distribution costs vary as the square of the Euclidean distance separating the source and the demand locations. In other applications where the flow of goods or materials occurs along grids of city streets or factory aisles, the cost is more aptly proportional to the shipment volume and the *rectilinear distance* through which this shipment occurs. This corresponding problem can be equivalently reformulated as a mixed-integer zero-one, bilinear programming problem in the space of the location-decision variables. Sherali *et al.* (1994) describe a specialization of the RLT procedure for generating tight relaxations within a branch-and-bound scheme for solving this problem. Using this approach, they were able to solve difficult instances of this nonconvex problem having up to 5 sources and 20 customer locations to optimality. In addition, because of the tight relaxations obtained, this algorithm also provides an efficient heuristic which upon premature termination is capable of obtaining provably good quality solutions (with 5-10% of optimality) for larger sized problems.

Predating this work, we had also studied a discrete variant of the location-allocation problem in which the *m* capacitated service facilities are to be assigned in a one-to-one fashion to some *m* discrete sites in order to serve the *n* customers, where the cost per unit flow is determined by some general facility-customer separation based penalty function (see Sherali and Adams, 1984). This problem also turns out to have the structure of a separably constrained mixed-integer bilinear programming problem, and a partial first level RLT relaxation that includes only a subset of the RLT constraints, some in an aggregated form, was used to generate lower bounds. A set of 16 problems with (m,n) ranging up to (7,50)-(11,11) were solved using a Benders' partitioning approach. For these problem instances, even the partial, aggregated first level RLT relaxation produced lower bounds with 90-95% of optimality.

In a similar vein, another problem of interest is the **capacitated ℓ_p distance location-allocation problem.** This can be stated as follows:

$$\text{minimize} \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} w_{ij} \left[\left| x_i - a_j \right|^p + \left| y_i - b_j \right|^p \right]^{q/p} : \ w \varepsilon W, (x,y) \varepsilon Z \right\},$$

where there are m capacitated supply centers to be located to serve n demand locations that are situated at coordinates (a_j, b_j), $j = 1, \ldots, n$, and where (x_i, y_i) denotes the location decision for supply center i, $i = 1, \ldots, m$, w_{ij} denotes the product distribution decision from supply center i to customer j, $i = 1, \ldots, m$, $j = 1, \ldots, n$, W represents the supply-demand constraints, and Z restricts the facility locations to lie in some hyperrectangular region. Here, p and q are assumed to be rational, and given by $p = r / s$ and $q = t / u$, where r, s, t, u are natural numbers, the quantity $\left[\left| x_i - a_j \right|^p + \left| y_i - b_j \right|^p \right]$ gives the ℓ_p distance between (x_i, y_i) and (a_j, b_j), and the exponent q suitably adjusts this distance based penalty factor in the objective function. For $1 < p < 2$, the ℓ_p distance norm is considered to more accurately model reality in distribution problems (see Brimberg and Love, 1991). Love and Juel (1982) and Bongartz *et al.* (1994) have proposed local search methods for an uncapacitated version of this problem, but no global approach has been developed and tested. To cast this problem as a polynomial program of the form given in Chapter 7, let us put $\alpha_{ij} = \left|x_i - a_j\right| = \gamma^s_{ij}$, $\beta_{ij} = \left|y_i - b_j\right| = \delta^s_{ij}$, and $\gamma^r_{ij} + \delta^r_{ij} = \theta^{ur}_{ij} \quad \forall\ (i,j)$. Then this problem can be equivalently written as follows, where UB is an appropriately derived upper bound vector on the set of variables $(\alpha, \beta, \gamma, \delta, \theta)$ based on $(x, y)\varepsilon Z$.

$$\text{Minimize} \qquad \sum_i \sum_j c_{ij} w_{ij} \theta^{ts}_{ij}$$

$$\text{subject to} \qquad w\varepsilon W, (x, y)\varepsilon Z$$

$$\alpha_{ij} \geq (x_i - a_j),\ \alpha_{ij} \geq (a_j - x_i),\ \alpha_{ij} = \gamma^s_{ij} \quad \forall\ (i,j)$$

$$\beta_{ij} \geq (y_i - b_j),\ \beta_{ij} \geq (b_j - y_i),\ \beta_{ij} = \delta^s_{ij} \quad \forall\ (i,j)$$

$$\gamma^r_{ij} + \delta^r_{ij} = \theta^{ur}_{ij} \quad \forall\ (i,j)$$

$$0 \leq (\alpha, \beta, \gamma, \delta, \theta) \leq UB.$$

Interesting special cases of practical importance that need investigation are the Euclidean distance problem where $(p,q) = (2,1)$, and based on empirical data (see Love and Juel, 1982), the ℓ_p distance problem with $(p,q) = (5/3, 1)$. Note that we have discussed above the efficient solution of this problem for $(p,q) = (1,1)$ and $(p,q) = (2,2)$. Similar techniques can be developed for solving the above, more general, version of this problem.

Another special polynomial program of considerable interest is concerned with the **design and maintenance of a reliable water distribution network** that meets demand and pressure head requirements over several years of a planning horizon. This problem is of national concern, given the decay in the pipe network infrastructure. The nonconvexity arises here due to frictional head loss constraints (see Loganathan *et al.*, 1990 for a mathematical formulation and Sherali and Smith, 1993, for a discussion on related literature). While several local optimization schemes have been suggested for this problem, until recently as noted below, no global approach or even a satisfactory lower bounding scheme had been developed to evaluate the quality of solutions generated.

A specialized RLT procedure that uses the concepts of Chapter 7 to handle both integer and rational exponents in the frictional head-loss constraints has been developed by Sherali and Smith (1995). Here, for a classic test problem from the literature for which a series of imposed heuristic solutions have been published over the past 20 years we have obtained for the first time a true, provable global optimum for this problem. (Eiger, *et al.*, 1994, have proposed an alternative global optimization approach for this problem, but they solve only a restricted version of this test problem to near feasibility and optimality.) Global optima for some other variants of this test problem are reported in Sherali and Smith (1996), and further refinements and results are forthcoming.

In conclusion, we have described over Chapters 10 and 11 our experience in designing specialized RLT based relaxations for solving various specific classes of discrete and continuous nonconvex problems. We are currently investigating RLT designs for many other applications, notably, some telecommunication design problems, as well as the development of general purpose algorithmic strategies for solving linear mixed-integer 0-1 programming problems, and continuous nonconvex polynomial programming problems. We hope that the discussion in this book will encourage readers to use the RLT approach and constructs for solving nonconvex optimization problems that arise in their applications of interest.

REFERENCES

Adams, W. P. and P. M. Dearing, "On the Equivalence Between Roof Duality and Lagrangian Duality for Unconstrained 0-1 Quadratic Programming Problems," *Discrete Applied Mathematics*, **48**(1), 1-20, 1994.

Adams, W. P. and T. A. Johnson, "An Exact Solution Strategy for the Quadratic Assignment Problem Using RLT-Based Bounds," Working Paper, Department of Mathematical Sciences, Clemson University, Clemson, SC, 1996.

Adams, W. P. and T. A. Johnson, "Improved Linear Programming-Based Lower Bounds for the Quadratic Assignment Problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* "Quadratic Assignment and Related Problems," eds. P. M. Pardalos and H. Wolkowicz, **16**, 43-75, 1994.

Adams, W. P. and H. D. Sherali, "A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems," *Management Science*, **32**(10), 1274-1290, 1986.

Adams, W. P. and H. D. Sherali, "Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems," *Operations Research*, **38**(2), 217-226, 1990.

Adams, W. P. and H. D. Sherali, "Mixed-Integer Bilinear Programming Problems," *Mathematical Programming*, **59**(3), 279-305, 1993.

Adams, W. P., A. Billionnet, and A. Sutter, "Unconstrained 0-1 Optimization and Lagrangean Relaxation," *Discrete Applied Mathematics*, **29**(2-3), 131-142, 1990.

Adams, W. P., J. B. Lassiter, and H. D. Sherali, "Persistency in 0-1 Optimization," *Mathematics of Operations Research,* **23**, 359-389, 1998.

Alameddine, R. A, "A New Reformulation-Linearization Technique for the Bilinear Programming and Related Problems with Applications to Risk Management," Ph.D. Dissertation, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, 1990.

Al-Khayyal, F. A., "Linear, Quadratic, and Bilinear Programming Approaches to the Linear Complementarity Problem," *European Journal of Operational Research*, **24**, 216-227, 1986.

Al-Khayyal, F. A., "An Implicit Enumeration Procedure for the Linear Complementarity Problem," *Mathematical Programming Study 31,* **17**, 1-21, 1987.

Al-Khayyal, F. A., "On Solving Linear Complementarity Problems as Bilinear Programs," *Arab. J. for Science and Engineering*, **15**, 639-646, 1990.

Al-Khayyal, F. A., "Jointly Constrained Bilinear Programs: An Overview," *Journal of Computers and Mathematics with Applications*, **19**(11), 53-62, 1992.

Al-Khayyal, F. A. and J. E. Falk, "Jointly Constrained Biconvex Programming," *Mathematics of Operations Research*, 273-286, 1983.

Al-Khayyal, F. A. and C. Larson, "Global Minimization of a Quadratic Function Subject to a Bounded Mixed Integer Constraint Set," *Annals of Operations Research*, **25**, 169-180, 1990.

Al-Khayyal, F. A., C. Larson, and T. Van Voorhis, "A Relaxation Method for Nonconvex Quadratically Constrained Quadratic Programs," Working Paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georia 30332, 1994.

Araoz, J., J. Edmonds, and V. Griffin, "Lifting the Facets of Polyhedra," *Progress in Combinatorial Optimization*, Academic Press, 3-12, 1984.

Balakrishnan, A., T. L. Magnanti, A. Shulman, and R. T. Wong, "Models for Planning Capacity Expansion in Local Access Telecommunication Networks," *Annals of Operations Research*, **33**, 239-284, 1991.

Balas, E., "Facets of the Knapsack Polytope," *Mathematical Programming*, **8**(2), 146-164, 1975.

Balas, E., "Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems," *SIAM Journal on Algebraic and Discrete Methods*, **6**(3), 466-486, 1985.

Balas, E., "Finding Out Whether a Valid Inequality is Facet Defining," Management Science Research Report No. MSRR 558(R), Graduate School of Business Administration, Carnegie-Mellon University, Pittsburgh, PA 15213, 1990.

Balas, E., "Nonconvex Quadratic Programming via Generalized Polars," *SIAM Journal on Applied Mathematics*, **28**, 335-349, 1975.

Balas, E., "Some Valid Inequalities for the Set Partitioning Problems," *Annals of Discrete Mathematics*, **1**, 13-47, 1977.

Balas, E., "On the Facial Structure of Scheduling Polyhedra," *Mathematical Programming Study* **24**, 179-218, 1986.

Balas, E. and J. B. Mazzola, "Nonlinear 0-1 Programming: I. Linearization Techniques," *Mathematical Programming*, **30**, 2-12, 1984a.

Balas, E. and J. B. Mazzola, "Nonlinear 0-1 Programming: II. Dominance Relations and Algorithms," *Mathematical Programming*, **30**, 22-45, 1984b.

Balas, E. and N. Christofides, "A Restricted Lagrangian Approach to the Traveling Salesman Problem," *Mathematical Programming*, **21**, 19-46, 1981.

Balas, E. and P. Landweer, "Traffic Assignment in Communication Satellites," *Operations Research Letters* 2, 141-147, 1983.

Balas, E. and R. Martin, "Pivot and Complement: A Heuristic for 0-1 Programming," *Management Science*, **26**, 86-96, 1980.

Balas, E. and S. M. Ng, "On the Set Covering Polytope: II. Lifting the Facets with Coefficients in {0, 1, 2}," *Mathematical Programming*, **45**, 1-20, 1989.

Balas, E. and M. W. Padberg, "On the Set Covering Problem: II. An Algorithm for Set Partitioning," *Operations Research*, **23**(3), 74-90, 1975.

Balas, E. and M. W. Padberg, "Set Partitioning: A Survey," *SIAM Review*, **18**, 710-760, 1976.

Balas, E. and E. Zemel, "Critical Cutsets of Graphs and Canonical Facets of Set-Packing Polytopes," *Mathematics of Operations Research*, **2**(1), 15-19, 1977.

Balas, E. and E. Zemel, "Facets of the Knapsack Polytope from Minimal Covers," *SIAM Journal of Applied Mathematics*, **34**, 119-148, 1978.

Balas, E. and E. Zemel, "Lifting and Complementing Yields all the Facets of Positive Zero-One Polytopes," *Mathematical Programming*, Elsevier Publishing Co., 13-24, 1984.

Balas, E., S. Ceria, G. Cornuéjols, and G. Pataki, "Polyhedral Methods for the Maximum Clique Problem, Research Report, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

Balas, E., S. Ceria, and G. Cornuéjols, "A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs," *Mathematical Programming*, **58**(3), 295-324, 1993.

Balinski, M. L. "Integer Programming: Methods, Uses, Computation," *Management Science*, **12**, 253-313, 1965.

Balintify, J. L., G. T. Ross, P. Sinha, and A. A. Zoltners, "A Mathematical Programming System for Preference and Compatibility Maximized Menu Planning and Scheduling," *Mathematical Programming*, **15**, 63-76, 1978.

Barahona, F., M. Junger, and G. Reinelt, "Experiments in Quadratic 0-1 Programming," *Mathematical Programming* **44**, 127-137, 1989.

Barahona, F. and A. R. Mahjoub, "On the Cut Polytope," *Mathematical Programming*, **36**, 137-173, 1986.

Barany, I., T. J. Van Roy, and L. A. Wolsey, "Strong Formulations for Multi-Item Capacited Lot-Sizing," CORE Discussion Paper No. 8312, Center for Operations Research and Econometrics, University Catholique de Louvain, Belgium, 1983.

Bazaraa, M. S. and H. D. Sherali, "Benders' Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem," *Naval Research Logistics Quarterly*, **27**, 29-41, 1980.

Bazaraa, M. S. and H. D. Sherali, "On the Choice of Step Size in Subgradient Optimization," *European Journal of Operational Research*, **7**, 380-388, B-3, 1981.

Bazaraa, M. S. and J. J. Goode, "A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality," *European Journal of Operational Research*, **3**(4), 322-338, 1979.

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, second edition, John Wiley and Sons, Inc., New York, NY, 1993.

Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, second edition, John Wiley & Sons, New York, 1990.

Benacer, R. and Pham Dinh Tao, "Global Maximization of a Nondefinite Quadratic Function Over a Convex Polyhedron," in *Fermat Days 85: Mathematics for Optimization*, J. B. Hiriart-Urruty (ed.), North-Holland, Amsterdam, 65-76, 1986.

Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik*, **4**, 238-252, 1962.

Benson, H. P., "Separable Concave Minimization Via Partial Outer Approximatin and Branch-and-Bound," *Operations Research Letters*, **9**, 389-394, 1990.

Billionnet, A. and B. Jaumard, "A Decomposition Method for Minimizing Quadratic Pseudo-Boolean Functions," *Operations Research Letters*, **8**, 161-163, 1989.

Billionnet, A. and A. Sutter, "Persistency in Quadratic 0-1 Optimization," *Mathematical Programming*, **54**, 115-119, 1992.

Bixby, R. E., E. A. Boyd, and R. R. Indovina, "MIPLIB: A Test Set of Mixed Integer Programming Problems," *SIAM News*, **16**, 1992.

Blair, C. E., R. G. Jeroslow, and J. K. Lowe, "Some Results and Experiments in Programming Techniques for Propositional Logic," *Computers and Operations Research*, **13**(5), 633-645, 1986.

Bomze, I. M., "Copositivity Conditions for Global Optimality of Indefinite Quadratic Programming Problems," *Czechoslovak Journal for Operations Research*, **1**, 1-19, 1992.

Bongartz, I., P. H. Calamai, and A. R. Conn, "A Projection Method for ℓ_p Norm Location-Allocation Problems," *Mathematical Programming*, **66**, 283-312, 1994.

Bourjolly, J.-M. "An Extension of the König-Egerváry Property to Node-Weighted Bidirected Graphs," *Mathematical Programming*, **41**, 375-384, 1988.

Bourjolly, J.-M., "Stable Sets, Max-Cuts and Quadratic 0-1 Optimization," Working Paper, Concordia University, 1990.

Boros, E. and P. L. Hammer, "Cut-Polytopes, Boolean Quadratic Polytopes and Nonnegative Quadratic Pseudo-Boolean Functions," DIMACS Technical Report 90-37, 1990.

Boros, E., Y. Crama, and P. L. Hammer, "Upper Bounds for Quadratic 0-1 Maximization Problems," *Operations Research*, **9**, 73-79, 1990.

Bradley, G. H., "Transformation of Integer Programs to Knapsack Problems," *Discrete Mathematics*, **1**(1), 29-45, 1971.

Bradley, G. H., P. L. Hammer, and L. Wolsey, "Coefficient Reduction for Inequalities in 0-1 Variables," *Mathematical Programming*, **7**, 263-282, 1974.

Brimberg, J. and R. F. Love, "Estimating Travel Distances by the Weighted ℓ_p Norm," *Naval Research Logistics*, **38**, 241-259, 1991.

Bromberg, M., and T. Chang, "One Dimensional Global Optimization Using Linear Lower Bounds," in *Recent Advances in Global Optimization*, P. M. Pardalos, and C. A. Floudas (eds.), Princeton University Press, New Jersey, 200-220, 1992.

Camerini, P. M., L. Fratta, and F. Maffioli, "On Improving Relaxation Methods by Modified Gradient Techniques," *Mathematical Programming Study 3*, North-Holland Publishing Co., New York, NY, 26-34, 1975.

Cannon, T. L. and K. L. Hoffman, "Large-Scale 0-1 Programming on Distributed Workstations," *Annals of Operations Research*, **22**, 181-218, 1990.

Ceria, S., "Lift-and-Project Methods for Mixed 0-1 Programs," Ph.D. Dissertation, Graduate School of Industrial Administration, Carnegie Mellon University, 1993.

Chan, T. J. and C. A. Yano, "A Multiplier Adjustment Approach for the Set Partitioning Problem," *Operations Research*, **40**, S40-S47, 1992.

Choi, G., "Nondifferentiable Optimization Algorithms with Application to Solving Lagrangian Dual Problems," Ph.D. Dissertation, Department of ISE, VPI&SU, Blacksburg, VA 24061, 1993.

Chrissis, J. W., R. P. Davis, and D. M. Miller, "The Dynamic Set Covering Problem," Paper presented at the TIMS/ORSA Joint National Meeting, New York, 1978.

Chvatal, V., "Hard Knapsack Problems," *Operations Research*, **28**(6), 1402-1411, 1980.

Chvátal, V., "Edmonds Polytopes and a Hierarchy of Combinatorial Problems," *Discrete Mathematics*, **4**(4), 305-337, 1973.

Chvátal, V., "Hard Knapsack Problems," *Operations Research*, **28**(6), 1402-1411, 1980.

Cornuejols, G. and A. Sassano, "On the 0, 1 Facets of the Set Covering Problem," *Mathematical Programming*, **43**, 45-57, 1989.

Cornuejols, G., G. L. Nemhauser, and L. A. Wolsey, "The Uncapacitated Facility Location Problem," in *Discrete Location Theory*, (eds.) R. L. Francis and P. Mirchandani, Wiley, 119-172, 1990.

Cottle, R. W., J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press Inc., San Diego, CA, 1992.

CPLEX, "Using the CPLEX Linear Optimizer," CPLEX Optimization, Inc., Suite 279, 930 Tahoe Blvd., Bldg. 802, Incline Village, NV 89451, 1990.

Crowder, H. and M. W. Padberg, "Solving Large-Scale Symmetric Traveling Salesman Problems to Optimality," *Management Science*, **26**, 495-509, 1980.

Crowder, H., E. L. Johnson, and M. W. Padberg, "Solving Large-Scale Zero-One Linear Programming Problems," *Operations Research*, **31**, 803-834, 1983.

Dantzig, G., D. Fulkerson, and S. Johnson, "Solution of a Large Scale Traveling Salesman Problem," *Operations Research*, **2**, 393-410, 1954.

Demyanov, V. F. and L. V. Vasilev, *Nondifferentiable Optimization*, Springer-Verlag, Berlin-Heidelberg-New York, 1985.

Denizel, M., S. S. Erenguc, and H. D. Sherali, "Convex Envelope Results and Strong Formulations for a Class of Mixed Integer Problems," *Naval Research Logistics*, to appear, 1996.

Desrochers, M. and G. Laporte, "Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints," *Operations Research Letters*, **10**(1), 27-36, 1991.

Deza, M. and M. Laurent, "Facets for the Complete Cut Cone," Research Memorandum, Department of Math. Eng. and Inst. Physics, University of Tokyo, 1988.

Deza, M. and M. Laurent, "Facets for the Complete Cut Cone II: Clique-Web Inequalities," Document 57, LAMSADE, Universite de Paris Dauphine, 1989.

Deza, M. and M. Laurent, "New Results on Facets for the Cut Cone," Research Report. B-227, Tokyo Institute of Technology, 1989.

Deza, M. K. Fukuda, and M. Laurent, "The Inequicut Cone," Research Report 89-04, University of Tsukuba, 1989.

Dietrich, B. L. and L. F. Escudero, "Coefficient Reduction for Knapsack-Like Constraints in 0-1 Problems with Variable Upper Bounds," *Operations Research Letters*, **9**, 9-14, 1990.

Du, D-Z, "Minimax and Its Applications," in *Handbook of Global Optimization*, Nonconvex Optimization and its Applications, eds. R. Horst and P. M. Pardalos, Kluwer Academic Publishers, 339-368, 1995.

Duffin, R. J., E. L. Peterson, and C. Zener, *Geometric Programming*, John Wiley & Sons, Inc., New York, NY, 1967.

Edwards, C. S., "A Branch and Bound Algorithm for the Koopmans-Beckmann Quadratic Assignment Problem," *Mathematical Programming Study,* **13**, 35-53, 1980.

Eiger, G., U. Shamir, and A. Ben-Tal, "Optimal Design of Water Distribution Networks," *Water Resources Research,* **30**(9), 2637-2646, 1994.

Elmaghraby, S. E., "The Knapsack Problem with Generalized Upper Bounds," *European Journal of Operations Research,* **38**, 242-254, 1989.

Eppen, G. D. and R. K. Martin, "Solving Multi-Item Capacitated Lot Sizing Problems Using Variable Redefinition," Graduate School of Business, University of Chicago.

Erenguc, S. S. and H. P. Benson, "An Algorithm for Indefinite Integer Quadratic Programming," *Computers Math. Applications,* **21**(6, 7), 99-106, 1991.

Eyser, J. W. and J. A. White, "Some Properties of the Squared Euclidean Distance Location Problem," *IIE Transactions,* **5**(3), 275-280, 1973.

Falk, J. E., "Lagrange Multipliers and Nonconvex Programs, *SIAM Journal on Control,* **7**(4), 534-545, 1969.

Falk, J. E. and K. L. Hoffman, "Concave Minimization via Collapsing Polytopes," *Operations Research,* **34**(6), 919-929, 1986.

Fisher, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science,* **27**(1), 1-18, 1981.

Fisher, M. L. and P. Kedia, "Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristics," *Management Science,* **36**, 674-688, 1990.

Floudas, C. A. and P. M. Pardalos, "A Collection of Test Problems for Constrained Global Optimization Algorithms," *Lecture Notes in Computer Science 455*, eds. G. Goos and J. Hartmanis, Springer-Verlag, Berlin, 1990.

Floudas, C.A. and P. M. Pardalos, "A Collection of Test Problems for Constrained Global Optimization Algorithms," *Lecture Notes in Computer Science*, 455, eds. G. Goos and J. Hartmanis, Springer-Verlag, Berlin, 1987.

Floudas, C. A. and V. Visweswaran, "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLP's-1. Theory," *Computers and Chemical Engineering,* **14**, 1397-1417, 1990.

Floudas, C. A. and V. Visweswaran, "A Primal-Relaxed Dual Global Optimization Approach," Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, 1991.

Floudas, C. A. and V. Visweswaran, "A Primal-Relaxed Dual Global Optimization Approach," *Journal of Optimization Theory and Applications,* **78**(2), 1993.

Floudas, C. A. and V. Visweswaran, "Quadratic Optimization," in *Handbook of Global Optimization,* Nonconvex Optimization and its Applications, eds. R. Horst and P. M. Pardalos, Kluwer Academic Publishers, 217-270, 1995.

Fortet, R., "L'algebre de Boole et Ses Applications en Recherche Operationnelle," *Cahiers Centre Etudes Recherche Operationnelle,* **1**, 5-36, 1959.

Fortet, R., "Applications de l'algebre de Boole en Recherche Operationnelle," *Rev. Francaise Informat. Recherche Operationnelle,* **4**, 17-26, 1960.

Francis, R. L. and J. A. White, *Facility Layout and Location: An Analytical Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

Francis, R. L., L. F. McGinnis, Jr., and J. A. White, *Facility Layout and Location: An Analytical Approach*, second edition, Prentice-Hall, Englewood Cliffs, NJ, 1992.

Garfinkel, R. S. and Nemhauser, G. L., "The Set Partitioning Problem: Set Covering with Equality Constraints," *Operations Research*, **17**, 848-856, 1969.

Garfinkel, R. S. and G. L. Nemhauser, "A Survey of Integer Programming Emphasizing Computation and Relations Among Models," In *Mathematical Programming: Proceedings of an Advanced Seminar*, T. C. Hu and S. Robinson (eds.), Academic Press, New York, NY, 77-155, 1973.

Gehner, K. R., "Necessary and Sufficient Optimality Conditions for the Fritz-John Problem with Linear Equality Constraints," *SIAM J. Control*, **12**(1), 140-149, 1974.

Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," *SIAM Review*, **7**, 178-190, 1967.

Geoffrion, A. M., "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications*, **10**, 237-260, 1972.

Geoffrion, A. M., "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study 2*, M. L. Balinski (ed.), North-Holland Publishing Co., Amsterdam, 82-114, 1974.

Geoffrion, A. M. and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition," *Management Science*, **20**, 822-844, 1974.

Geoffrion, A. M. and R. McBryde, "Lagrangian Relaxation Applied to Facility Location Problems," *AIIE Transactions*, **10**, 40-47, 1979.

Glover, F., "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems," *Management Science*, **22**(4), 455-460, 1975.

Glover, F., "Polyhedral Convexity Cuts and Negative Edge Extensions," *Zeitschrift fur Operations Research*, **18**, 181-186, 1974.

Glover, F. and E. Woolsey, "Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program," *Operations Research*, **22**(1), 180-182, 1974.

Glover, F. and E. Woolsey, "Further Reduction of Zero-One Polynomial Programming Problems to Zero-One Linear Programming Problems," *Operations Research*, **21**, 156-161, 1973.

Glover, F., H. D. Sherali, and Y. Lee, "Generating Cuts from Surrogate Constraint Analysis for Zero-One and Multiple Choice Programming," *Computational Optimization and Applications*, to appear, 1996.

Goffin, J. L., "On the Convergence Rate of Subgradient Methods," *Mathematical Programming*, **13**, 329-347, 1977.

Golub, G. H. and C. F. Van Loan, *Matrix Computations*, second edition, The Johns Hopkins University Press, Baltimore, 1989.

Gottlieb, E. S., "Facet Defining Inequalities for the Dichotomous Knapsack Problem," *Operations Research Letters*, **9**, 21-29, 1990.

Gottlieb, E. S. and M. R. Rao, "Facets of the Knapsack Polytope Derived from Disjoint and Overlapping Index Configurations," *Operations Research Letters*, **7**(2), 95-100, 1988.

Granot, D. and F. Granot, "Generalized Covering Relaxation of 0-1 Programs," *Operations Research*, **28**, 1442-1449, 1980.

Granot, D. and F. Granot, "Minimal Covers, Minimal Sets and Canonical Facets of the Polynomial Knapsack Polytope," *Discrete Applied Mathematics*, **9**, 171-185, 1984.

Granot, D., F. Granot, and J. Kallberg, "Covering Relaxation for Positive 0-1 Polynomial Programs," *Management Science*, **25**, 264-273, 1979.

Granot, F. and P. L. Hammer, "On the Use of Boolean Functions in 0-1 Programming," *Mathematics of Operations Research*, **12**, 154-184, 1971.

Grossmann, I. E., *"Global Optimization in Engineering Design,"* Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Boston, **9**, 1996.

Grötschel, M. and O. Holland, "Solving Matching Problems with Linear Programming," *Mathematical Programming*, **33**, 243-259, 1985.

Grötschel, M. and M. Padberg, "Partial Linear Characterizations of the Asymmetric Traveling Salesman Polytope," *Mathematical Programming*, **8**, 378-381, 1975.

Grotzinger, S. J., "Supports and Convex Envelopes," *Mathematical Programming*, **31**, 339-347, 1985.

Gu, Z., G. L. Nemhauser, and M. W. P. Savelsbergh, "Lifted Cover Inequalities for 0-1 Integer Programs I: Computation, II: Complexity, and III: Fast Algorithms," Working Paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1995.

Gu, Z., G. L. Nemhauser, and M. W. P. Savelsbergh, "Sequence Independent Lifting," Working Paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1995.

Guignard, M. and S. Kim, "Lagrangian Decomposition: A Model Yielding Stronger Lagrangean Bounds," *Mathematical Programming*, **39**, 215-228, 1987.

Hahn, P., W. Hightower, and T. Johnson, Private Communication (paper in preparation), 1998.

Hamed, A. S. E. and G. P. McCormick, "Calculation of Bounds on Variables Satisfying Nonlinear Inequality Constraints," *Journal of Global Optimization*, **3**, 25-47, 1993.

Hammer, P. L., Hansen, P., and B. Simeone, "Vertices Belonging to All or to No Maximum Stable Sets of a Graph," *SIAM Journal on Algebraic and Discrete Methods*, **3**(4), 511-522, 1982.

Hammer, P. L., P. Hansen and B. Simeone, "Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization," *Mathematical Programming*, **28**, 121-155, 1984.

Hammer, P. L., E. L. Johnson and U. N. Peled, "Facets of Regular 0-1 Polytopes," *Mathematical Programming*, **8**(20), 179-206, 1975.

Handler, G. Y. and P. B. Mirchandani, *Location on Networks*, MIT Press, Cambridge, MA, 1979.

Hansen, E., "Global Optimization Using Interval Analysis," *Monographs and Textbooks in Pure and Applied Mathematics*, No. 165, Marcel Dekker, Inc., New York, 1992.

Hansen, P, "Methods of Nonlinear 0-1 Programming," *Annals of Discrete Mathematics*, **5**, 53-71, 1979.

Hansen, P. and B. Jaumard, "Reduction of Indefinite Quadratic Progress to Bilinear Programs," *Journal of Global Optimization*, **2**(1), 41-60, 1992.

Hansen, P., B. Jaumard, and S. Lu, "An Analytical Approach to Global Optimization," *Mathematical Programming,* **52**, 227-254, 1991.

Hansen, P., B. Jaumard, and V. Mathon, "Constrained Nonlinear 0-1 Programming," Report G-89-38, Gerard-Montreal, 1989.

Hansen, P., B. Jaumard, and J. Xiong, "Decomposition and Interval Arithmetic Applied to Global Minimization of Polynomial and Rational Functions," *Journal of Global Optimization*, **3**, 421-437, 1993.

Hayes, A. C. and D. G. Larman, "The Vertices of the Knapsack Polytope," *Discrete Applied Mathematics*, **6**, 135-138, 1983.

Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Tree," *Operations Research*, **18**(6), 1138-1162, 1970.

Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming*, **1**(1), 6-25, 1971.

Held, M., P. Wolfe, and H. D. Crowder, "Validation of Subgradient Optimization," *Mathematical Programming*, **6**(1), 62-88, 1974.

Hochbaum, D. S., "Efficient Bounds for the Stable Set, Vertex Cover and Set Packing Problems," *Discrete Applied Mathematics*, **6**, 243-254, 1983.

Hochbaum, D. S., N. Megiddo, J. Naor, and A. Tamir, "Tight Bounds and 2-Approximation Algorithms for Integer Programs with Two Variables per Inequality," *Mathematical Programming*, **62**, 69-83, 1993.

Hoffman, K. and M. Padberg, "LP-Based Combinatorial Problem Solving," *Annals of Operations Research*, **4**, 145-194, 1985/6.

Hoffman, K. L. and M. Padberg, "Improving LP-Representations of Zero-One Linear Programs for Branch-and-Cut," *ORSA Journal on Computing,* **3**(2), 121-134, 1991.

Hoffman, K. L. and M. Padberg, "Solving Large Set-Partitioning Problems with Side Constraints," *ORSA/TIMS Joint National Meeting*, San Francisco, 1992.

Hoffman, K. L. and M. Padberg, "Techniques for Improving the LP-Representation of 0-1 Linear Programming Problems," preprint, 1989.

Horst, R., "An Algorithm for Nonconvex Programming Problems," *Mathematical Programming,* **10**, 312-321, 1976.

Horst, R., "A General Class of Branch-and-Bound Methods in Global Optimization with Some New Approaches for Concave Minimization," *Journal of Optimization Theory and Applications*, **51**(2), 271-191, 1986.

Horst, R., "Deterministic Methods in Constrained Global Optimization: Some Recent Advances and New Fields of Application," *Naval Research Logistics*, **37**, 433-471, 1990.

Horst, R., "On the Global Minimization of Concave Functions: Introduction and Survey," *Operations Research Spektrum*, **6**, 195-205, 1984.

Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin, 1990.

Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches,* 2nd ed., Springer-Verlag, Berlin, Germany, 1993.

Jeroslow, R. G., "Trivial Integer Programs Unsolvable by Branch-and-Bound," *Mathematical Programming*, **6**, 105-109, 1974.

Jeroslow, R. G., "Representation of Unbounded Optimizations as Integer Programs," *Journal of Optimization Theory and Applications*, **30**, 339-351, 1980.

Jeroslow, R. G. and J. K. Lowe, "Modeling with Integer Variables," *Mathematical Programming Study* **22**, 167-184, 1984.

Jeroslow, R. G., "Representability of Functions," College of Industrial Management, Georgia Institute of Technology, 1984a.

Jeroslow, R. G., "Representability in Mixed Integer Programming, I: Characterization of Results," College of Industrial Management, Georgia Institute of Technology, 1984b.

Jeroslow, R. G., "Representability in Mixed Integer Programming II: A Lattice of Relaxations," College of Industrial Management, Georgia Institute of Technology, 1985.

Jeroslow, R. G., "Alternative Formulations of Mixed-Integer Programs," College of Industrial Management, Georgia Institute of Technology, Atlanta, GA 30332, 1993.

Jeroslow, R. G. and J. K. Lowe, "Experimental Results on New Techniques for Integer Programming Formulations," *Journal of the Operational Research Society*, **36**, 393-403, 1985.

Johnson, E. L., "Modeling and Strong Linear Programs for Mixed Integer Programming," *Algorithms and Model Formulations in Mathematical Programming*, NATO ASI 51, (ed.) S. Wallace, Springer-Verlag, 3-43, 1989.

Johnson, E. L., "Subadditive Lifting Methods for Partitioning and Knapsack Problems," *Journal of Algorithms*, **1**, 75-96, 1980.

Johnson, E. L. and M. Padberg, "A Note on the Knapsack Problems with Special Ordered Sets," *Operations Research Letters*, **1**, 18-23, 1981.

Johnson, E. L. and U. H. Suhl, "Experiments in Integer Programming," *Discrete Applied Mathematics*, **2**, 39-55, 1980.

Johnson, E. L., M. M. Kostreva, and U. H. Suhl, "Solving 0-1 Integer Programming Problems Arising From Large Scale Planning Models," *Operations Research*, **33**(4), 803-819, 1985.

Júdice, J. J. and A. M. Faustino, "An Experimental Investigation of Enumeration Methods for Linear Complementarity Problems, *Computers Opers. Res.*, **15**(5), 417-426, 1988.

Kalantari, B. and J. B. Rosen, "An Algorithm for Global Minimization of Linearly Constrained Concave Quadratic Functions," *Mathematics of Operations Research*, 544-561, 1987.

Kennington, J. L. and R. V. Helgason, *Algorithms for Network Programming*, Wiley, New York, 1980.

Kindervater, G. A. P. and J. K. Lenstra, "An Introduction to Parallelism in Integer Programming," *Discrete Applied Mathematics*, **14**, 135-156, 1986.

Kiwiel, K. C., "An Aggregate Subgradient Methods for Nonsmooth Convex Minimization," *Mathematical Programming*, **27**, 320-341, K-2, 1983.

Kiwiel, K. C., *Methods of Descent of Nondifferentiable Optimization*, Lecture Notes in Mathematics, 1133, Springer-Verlag, Berlin, 1985.

Kiwiel, K. C. "Proximity Control in Bundle Methods for Convex Nondifferentiable Minimization," *Mathematical Programming*, **46**, 105-122, 1990.

Kiwiel, K. C., "The Efficiency of Subgradient Projection Methods for Convex Optimization, Part I: General Level Methods; Part II: Implementations and Extensions," *SIAM Journal on Control and Optimization,* **34**(2) (to appear), 1996.

Kojima, M., N. Megiddo, T. Noma, and A. Yoshise,, *A Unified Approach to Interior Point Algorithms for LInear Complementarity Problems,* Volume 538 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, 1991.

Konno, H., "Bilinear Programming, Part I: An Algorithm for Solving Bilinear Programs," Technical Report No. 71-9, Operations Research House, Stanford University (Stanford, CA), 1971.

Konno, H., "Bilinear Programming, Part II: Applications of Bilinear Programming," Technical Report No. 71-10, Operations Research House, Stanford University (Stanford, CA), 1971.

Konno, H., "A Cutting Plane Algorithm for Solving Bilinear Programs," *Mathematical Programming,* **11**, 14-27, 1976.

Konno, H., "Maximization of a Convex Quadratic Function Under Linear Constraints," *Mathematical Programming,* **11**, 117-127, 1976.

Konno, H. and T. Kuno (KK), *Generalized Linear Multiplicative and Fractional Programming,* forthcoming, 1989.

Kortanek, K. L., X. Xu, and Y. Ye, "An Infeasible Interior-Point Algorithm for Solving Primal and Dual Geometric Programs," Manuscript, Department of Management Science, The University of Iowa, Iowa City, IA 52242, 1995.

Kostreva, M. M. and L. A. Kinard, "A Differentiable Homotopy Approach for Solving Polynomial Optimization Problems and Noncooperative Games," *Computers Math. Applic.*, **21**(6/7), 135-143, 1991.

Kough, P. F., "The Indefinite Quadratic Programming Problem," *Operations Research,* **27**(3), 516-533, 1979.

Kozlov, M. K., S. P. Taransov, and L. G. Hachijan, "Polynomial Solvability of Convex Quadratic Programming," *Soviet Mathematics Doklady*, **20**(5), 1979.

Larsson, T. and Z. Liu, "A Primal Convergence Result for Dual Subgradient Optimization with Application to Multicommodity Network Flows," Research Report, Department of Mathematics, Linkoping Institute of Technology, S581 83, Linkoping, Sweden, 1989.

Lasdon, L. S., A. D. Waren, S. Sarkas, and F. Palacios, "Solving the Pooling Problem Using Generalized Reduced Gradient and Successive Linear Programming Algorithms," *ACM SIGMAP Bulletin*, **27**, E, 9, 1979.

Lassiter, J. B., "Persistency in 0-1 Optimization," Ph.D. Dissertation, Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-1907, 1993.

Lassiter , J. B. and W. P. Adams, "Strategies for Converting Various 0-1 Polynomial Programs into Persistent Mixed 0-1 Linear Problems," Working Paper, Department of Mathematical Sciences, Clemson University, Clemson, SC, 1994.

Lassiter, J. B. and W. P. Adams, "Constructing Persistent Mixed-Integer Linear Reformulation of 0-1 Polynomial Programs," Working Paper, Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-1907, 1997.

Lawler, E. L. J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley Interscience, John Wiley & Sons, Inc., New York, NY, 1985.

Lee, Y., "Dedicated Digital Service and Carrier Network Optimization Problems," US West Advanced Technologies, 4003 Discovery Drive, Boulder, CO, 80303, 1993.

Lemarechal, C., "An Extension of Davidon Methods to Non-Differentiable Problems," *Mathematical Programming Study 3,* 95-109, L-3, 1975a.

Lemarechal, C., "An Extension of Davidon Methods to Non-Differentiable Problems," *Mathematical Programming Study 3,* North-Holland Publishing Co., New York, NY, 95-109, 1975b.

Lemarechal, C., "Bundle Method in Nonsmooth Optimization," *Nonsmooth Optimization: Proceedings of IIASA Workshop*, C. Lemarechal and R. Mifflin (eds.), 79-109, 1978.

Lemarechal, C., "Numerical Experiments in Nonsmooth Optimization," *Progress in Nondifferentiable Optimization*, E. A. Nurminski (ed.), IIASA, 61-84, 1982.

Lemarechal, C., J. J. Strodiot, and A. Bihain, "On a Bundle Algorithm for Nonsmooth Optimization," *Nonlinear Programming,* **4**, Mangasarian, Meyer, Robinson (eds.), 245-282, 1980.

Leung, J. M. Y. and T. L. Magnanti, "Facets and Algorithms for Capacitated Lot Sizing," *Mathematical Programming*, **45**, 331-360, 1989.

Leung, J. M. Y. and T. L. Magnanti, "Valid Inequalities and Facets of the Capacitated Plant Location Problem," *Mathematical Programming*, **44**, 271-292, 1989.

Loganathan, G. V., H. D. Sherali, and M. P. Shah, "A Two-Phase Network Design Heuristic for Minimum Cost Water Distribution Systems Under a Reliability Constraint," *Engineering Optimization*, **15**, 311-336, 1990.

Lovász, L. and A. Schrijver, "Cones of Matrices and Set Functions, and 0-1 Optimization," *SIAM J. Opt.*, **1**, 166-190, 1991.

Love, R. F. and H. Juel, "Properties and Solution Methods for Large Location-Allocation Problems," *Journal of the Operational Research Society*, **33**, 443-452, 1982.

Lu, S. H., and A. C. Williams, "Roof Duality for Polynomial 0-1 Optimization," *Mathematical Programming*, **37**, 357-360, 1987.

Lustig, I. J., R. E. Marsten, and D. F. Shanno, "Computational Experience with a Primal-Dual Interior Point Method for Linear Programming," preprint, 1979.

Magnanti, T. L. and R. T. Wong, "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria," *Operations Research*, **29**, 464-484, 1981.

Magnanti, T. L. and R. T. Wong, "Network Design and Transportation Planning: Models and Algorithms," *Transportation Science*, **18**(1), 1-55, 1984.

Manas, M., "An Algorithm for a Nonconvex Programming Problem," *Econ Math Obzor Acad. Nacl. Ceskoslov*, **4**(2), 202-212, 1968.

Mangasarian, O. L., "Simplified Characterizations of Linear Complementarity Problems Solvable as Linear Programs," *Mathematics of Operations Research*, **4**, 268-273, 1979.

Maranas, C. D. and C. A. Floudas, "Global Optimization in Generalized Geometric Programming," Working Paper, Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5363, 1994.

Marsten, R. E., "An Algorithm for Large Set Partitioning Problems," *Management Science*, **20**, 774-787, 1974.

Marsten, R. E., "The Design of the XMP Linear Programming Library," *ACM Trans. Math. Software*, **7**, 481-497, 1981.

Marsten, R. E., "OB1," Personal Communication, 1990.

Marsten, R. E., "XMP and XMP-ZOOM," User's Guide, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, 1990.

Martello, S. and P. Toth, *Knapsack Problems*, John Wiley and Sons, 1990.

Martin, K. R., "Generating Alternative Mixed-Integer Programming Models Using Variable Redefinition," *Operations Research*, **35**, 820-831, 1987.

Martin, K. R. and L. Schrage, "Constraint Aggregation and Coefficient Reduction Cuts for Mixed-0/1 Linear Programming," Graduate School of Business, University of Chicago, Chicago, Illinois, 1983.

Martin, R. K. and L. Schrage, "Subset Coefficient Reduction Cuts for 0-1 Mixed Integer Programming," *Operations Research*, **33**(3), 505-526, 1985.

Mazzola, J. B., "Compact Linearization of Nonlinear 0-1 Programs Containing Negative Coefficients," Working Paper, Fuqua School of Business, Duke University, 1988.

McBride, R. D. and J. S. Yormark, "An Implicit Enumeration Algorithm for Quadratic Integer Programming," *Management Science*, **26**, 282-296, 1980.

McCormick, G. P., "Computability of Global Solutions to Factorable Nonconvex Programs: Part I — Convex Underestimating Problems," *Mathematical Programming*, **10**, 147-175, 1976.

McDaniel, D. and M. Devine, "A Modified Benders' Partitioning Algorithm for Mixed Integer Programming," *Management Science*, **24**, 312-319, 1977.

Meyer, G. G., "Convergence of Relaxation Algorithms by Averaging," *Mathematical Programming*, **40**, 205-212, 1988.

Meyer, R. R., "Integer and Mixed-Integer Programming Models: General Properties," *Journal of Optimization Theory and Applications*, **16**(3-4), 191-206, 1975.

Meyer, R. R., "Mixed-Integer Minimization Models for Piecewise-Linear Functions of a Single Variable," *Discrete Mathematics*, **16**, 163-171, 1976.

Meyer, R. R., M. V. Thakkar, and W. P. Hallman., "Rational Mixed-Integer and Polyhedral Union Minimization Models," *Mathematics of Operations Research*, **5**, 135-146, 1980.

Meyer, R. R., "A Theoretical and Computational Comparison of 'Equivalent' Mixed-Integer Formulations," *Naval Research Logistics Quarterly*, **28**, 115-131, 1981.

Mifflin, R., "Convergence of a Modification of Lemarechal's Algorithm for Nonsmooth Optimization," *Progress in Nondifferentiable Optimization*, E. A. Nurminski (ed.), IIASA, M-5, 1982.

Minoux, M., "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications," *Networks*, **19**, 313-360, 1989.

Mirchandani, P. B. and R. L. Francis, *Discrete Location Theory*, John Wiley and Sons, Inc., New York, NY, 1990.

Mueller, R. K., "A Method for Solving the Indefinite Quadratic Programming Problem," *Management Science*, **16**(5), 333-339, 1970.

Murtagh, B. A. and M. A. Saunders, "MINOS 5.1 User's Guide," Technical Report Sol 83-20R, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, 1987. (Update: MINOS 5.4.)

Murty, K. G., *Linear Complementarity, Linear and Nonlinear Programming*, Helderman Verlag, Berlin, Germany, 1988.

Muu, L. D. and W. Oettli, "An Algorithm for Indefinite Quadratic Programming with Convex Constraints," *Operations Research Letters,* 10, 323-327, 1991.

Naddef, D., "The Hirsch Conjecture is True for (0, 1) Polytopes," *Mathematical Programming*, 45, 109-110, 1989.

Nauss, R. M., "The 0-1 Knapsack Problem With Multiple Choice Constraints," *European Journal of Operational Research,* **2**, 125-131, 1979.

Nemhauser, G., "Branch-and-Cut: Column Generation for Solving Huge Integer Programs," *15th International Symposium on Mathematical Programming*," The University of Michigan, Ann Arbor, Michigan, August 15-19, 1994.

Nemhauser, G. L., and L. E. Trotter, Jr., "Vertex Packings: Structural Properties and Algorithms," *Mathematical Programming*, **8**, 232-248, 1975.

Nemhauser, G. L. and G. M. Weber, "Optimal Set Partitioning, Matchings and Lagrangean Duality," *Naval Research Logistics Quarterly*, **26**, 553-563, 1979.

Nemhauser, G. L. and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.

Nemhauser, G. L. and L. A. Wolsey, "Integer Programming," in Handbooks in Operations Research and Management Science, **1**, *Optimization*, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd (eds.), 427-528, 1989.

Nemhauser, G. L. and L. A. Wolsey, "A Recursive Procedure for Generating all Cuts for Mixed-Integer Programs," *Mathematical Programming*, **46**, 379-390, 1990.

Nemhauser, G. L., G. Sigismondi, and P. Vance, "A Characterization of the Coefficients in Facet-Defining Lifted Cover Inequalities," ISyE Report No. J-89-06, Georgia Institute of Technology, Atlanta, GA 30332, 1989.

Nemhauser, G. L., M. Savelsbergh and G. Sigismondi, "MINTO: A Mixed INTeger Optimizer," School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, 1991.

Nugent, C. E., T. E. Vollmann, and J. Rume, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," *Operations Research*, **16**, 150-173, 1968.

Oley, L. A. and R. J. Sjouquist, "Automatic Reformulation of Mixed and Pure Integer Models to Reduce Solution Time in Apex IV," Presented at the *ORSA/TIMS* Fall Meeting, San Diego, 1982.

Oral, M. and O, Kettani, "A Linearization Procedure for Quadratic and Cubic Mixed-Integer Problems," *Operations Research*, **40**(1), S109-S116, 1992.

OSL, "Optimization Subroutine Library, Guide and Reference," IBM, August, 1990.

Overton, M. and H. Wolkowicz, "Semidefinite Programming," *Mathematical Programming*, **77**(2), 105-110, 1997.

Padberg, M. W., "On the Facial Structure of Set Packing Polyhedra," *Mathematical Programming*, **5**, 199-215, 1973.

Padberg, M. W., "A Note on Zero-One Programming," *Operations Research*, **23**, 833-837, 1975.

Padberg, M. W., "Covering, Packing and Knapsack Problems," *Annals of Discrete Mathematics*, **4**, 265-287, 1979.

Padberg, M. W., "(1,k)-Configurations and Facets for Packing Problems," *Mathematical Programming*, **18**, 94-99, 1980.

Padberg, M., "The Boolean Quadric Polytope; Some Characteristics, Facets and Relatives," *Mathematical Programming*, **45**, 139-172, 1989.

Padberg, M. and G. Rinaldi, "A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems," *SIAM Review*, **33**, 60-100, 1991.

Padberg, M. W., "The Boolean Quadric Polytope: Some Characteristics, Facets, and Relatives," *Mathematical Programming*, **45**, 139-172, 1989.

Pardalos, P. M., "Global Optimization Algorithms for Linearly Constrained Indefinite Quadratic Programs," *Computers Math. Applic.*, **21**, 87-97, 1991.

Pardalos, P. M. and J. B. Rosen, "Reduction of Nonlinear Integer Separable Programming Problems," Computer Science Department, The Pennsylvania State University and the University of Minnesota, 1984.

Pardalos, P. M. and J. B. Rosen, "Methods for Global Concave Minimization: A Bibliographic Survey," *SIAM Review*, **28**(3), 367-379, 1986.

Pardalos, P. M. and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Springer-Verlag, Lecture Notes in Computer Science, **268**, eds. G. Goos and J. Hartmanis, Springer, 1987.

Pardalos, P. M. and J. B. Rosen, "Bounds for Solution Sets of Linear Complementarity Problems, *Discrete Applied Mathematics*, **17**, 255-261, 1987.

Pardalos, P. M. and J. B. Rosen, "Global Optimization Approach to the Linear Complementarity Problem," *SIAM Journal of Scientific and Statistical Computing*, **9**(20), 341-353, 1988.

Pardalos, P. M. and S. A. Vavasis, "Quadratic Programming with One Negative Eigenvalue is NP-Hard," *Journal of Global Optimization*, **1**, 15-22, 1991.

Pardalos, P. M., J. H. Glick, and J. B. Rosen, "Global Minimization of Indefinite Quadratic Problems," *Computing*, **39**, 281-291, 1987.

Pardalos, P. M., Y. Ye, C. G. Han, and J. Kalinski,, "Solution of *P*-Matrix Linear Complementarity Problems Using a Potential Reduction Algorithm," *SIAM Journal of Matrix Anal. and Applications*, **14**(4), 1048-1060, 1993.

Parker, R. G. and R. L. Rardin, *Discrete Optimization*, Academic Press, New York, NY, 1988.

Peterson, C., "A Note on Transforming the Product of Variables to Linear Form in Linear Programs," Working Paper, Purdue University, 1971.

Peterson, E. L., "Geometric Programming," *SIAM Review*, **18**, 1-15, 1976, 1994.

Phillips, A. T. and J. B. Rosen, "A Parallel Algorithm for Solving the Linear Complementarity Problem," *Annals of OR, Parallel Optimization on Naval Computer Architectures*, eds. S. A. Zenios and R. R. Meyer, 1987.

Phillips, A. T. and J. B. Rosen, "Guaranteed ε-Approximate Solution for Indefinite Quadratic Global Minimization," *Naval Research Logistics*, **37**, 499-514, 1990.

Picard, J., and M. Queyrenne, "On the Integer-Valued Variables in the Linear Vertex Packing Problems," *Mathematical Programming*, **12**, 97-101, 1977.

Poljak, B. T., "A General Method of Solving Extremum Problems," *Soviet Mathematics*, **8**(3), 593-597, 1967.

Poljak, B. T., "Subgradient Methods: A Survey of Soviet Research," *Nonsmooth Optimization: Proceedings of IIASA Workshop*, C. Lemarechal and R. Mifflin (eds.), 5-30, 1978.

Poljak, B. T., "The Method of Conjugate Gradient in Extremum Problems," *USSR Computational Math and Math Physics*, **9**(4), 94-112, 1969.

Poljak, B. T. and Q. Mayne, "On the Approaches to the Construction of Nondifferentiable Optimization Problem," *System Modelling and Optimization: Proceedings of the 11th IFIP Conference*, P. Thoft-Christensen (ed.), Lecture Notes in Control and Information Sciences, **59**, 331-337, P-1, 1984.

Quesada, I. and I. E. Grossmann, "A Global Optimization Algorithm for Linear Fractional and Bilinear Programs," *Journal of Global Optimization*, **6**, 39-76, 1995.

Ramachandran, B. and J. F. Pekny, "Dynamic Factorization Methods for Using Formulations Derived from Higher Order Lifting Techniques in the Solution of the Quadratic Assignment Problem," in *State of the Art in Global Optimization*, eds. C. A. Floudas and P. M. Pardalos, Kluwer Academic Publishers, **7**, 75-92, 1996.

Ramakrishnan, K. G., M. G. C. Resende, and P. M. Pardalos "A Branch and Bound Algorithm for the Quadratic Assignment Problem Using a Lower Bound Based on Linear Programming," in *State of the Art in Global Optimization*, eds. C. A. Floudas and P. M. Pardalos, Kluwer Academic Publishers, **7**, 57-74, 1996.

Ramarao, B. and C. M. Shetty, "Application of Disjunctive Programming to the Linear Complementarity Problem," *Naval Research Logistics Quarterly*, **31**, 589-600, 1984.

Rardin, R. L. and U. Choe, "Tighter Relaxations of Fixed Charge Network Flow Problems," Report Series No. J-79-18, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, 1979.

Rardin, R. L. and V . E. Unger, "Surrogate Constraints and the Strength of Bounds Derived from 0-1 Benders' Partitioning Procedures," *Operations Research*, **24**, 1169-1175, 1976.

Rhys, J. M. W., "A Selection Problem of Shared Fixed Costs and Network Flows," *Management Science*, **17**, 200-207, 1970.

Ritter, K., "A Method for Solving Maximum Problems with a Nonconcave Quadratic Objective Function," *Z Wahrscheinlihkeitstheoric*, **4**, 340-351, 1966.

Rockafellar, R. T., *Convex Analysis*, Princeton University Press, NJ, 1970.

Rogers, D. F., R. D. Plante, R. T. Wong, and J. R. Evans, "Aggregation and Disaggregation Techniques and Methodology in Optimization," *Operations Research*, **39**(4), 553-582, 1991.

Ryoo, H. S. and N. V. Sahinidis, "A Branch-and-Reduce Approach to Global Optimization," Manuscript, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champaign, 1206 W. Green Street, Urbana, IL 61801, October, 1994.

Ryoo, H. S. and N. V. Sahinidis, "Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design," *Computers and Chemical Engineering*, **19**(5), 551-566, 1995.

Salkin, H. M. and K. Mathur, *Foundations of Integer Programming*, Elsevier Science Publishing Co., Inc., New York, NY, 1989.

Savelsbergh, M. W. P., G. C. Sigismondi, and G. L. Nemhauser, "MINTO, A Mixed INTeger Optimizer," COC-91-04, Report Series, Computational Optimization Center, School of I.Sy.E., Georgia Tech, Atlanta, GA 30332, 1991.

Selim, S. S., "Biconvex Programming and Deterministic and Stochastic Location Allocation Problems," Ph.D. Dissertation, Georgia Institute of Technology, 1979.

Sen, S. and H. D. Sherali, "A Class of Convergent Primal-Dual Subgradient Algorithms for Decomposable Convex Programs," *Mathematical Programming*, **35**, 279-297, S-17, 1986.

Shapiro, J. F., "A Survey of Lagrangian Techniques for Discrete Optimization," *Annals of Discrete Mathematics*, **5**, 113-138, 1979.

Shectman, H. P. and N. V. Sahindis, "A Finite Algorithm for Global Minimization of Separable Concave Programs," Technical Report, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champagne, IL, 1994.

Shenoi, R. G. and C. Barnhart, "Models and Column Generation Techniques for Long-Haul Crew Scheduling," *15th International Symposium on Mathematical Programming*, The University of Michigan, Ann Arbor, Michigan, August 15-19, 1994.

Sherali, H. D., "Convex Envelopes of Multlinear Functions Over a Unit Hypercube and Over Special Discrete Sets," *ACTA Mathematica Vietnamica*, Special issue in honor of Professor H. Tuy eds. N. V. Trung and D. T. Luc, **22**(1), 245-270, 1997.

Sherali, H. D., "Global Optimization of Nonconvex Polynomial Programming Problems Having Rational Exponents," *Journal of Global Optimization*, **12**(3), 267-283, 1998.

Sherali, H. D. and W. P. Adams, "A Decomposition Algorithm for a Discrete Location-Allocation Problem," *Operations Research*, **32**(4), 878-900, 1984.

Sherali, H. D. and W. P. Adams, "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems," *SIAM Journal on Discrete Mathematics*, **3**(3), 411-430, 1990.

Sherali, H. D. and W. P. Adams, "A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems," *Discrete Applied Mathematics*, **52**, 83-106, 1994. (Manuscript, 1989).

Sherali, H. D. and W. P. Adams, "A Reformulation-Linearization Technique (RLT) for Solving Discrete and Continuous Nonconvex Programming Problems," Vol. XII-A, *Mathematics Today*, special issue on *Recent Advances in Mathematical Programming*, ed. O. Gupta, 61-78, 1994.

Sherali, H. D., W. P. Adams, and P. Driscoll, "Exploiting Special Structures in Constructing a Hierarchy of Relaxations for 0-1 Mixed Integer Problems," *Operations Research.*, **46**(3), 396-405, 1998.

Sherali, H. D. and A. Alameddine, "A New Reformulation-Linearization Algorithm for Solving Bilinear Programming Problems," *Journal of Global Optimization*, **2**, 379-410, 1992.

Sherali, H. D. and A. Alameddine, "An Explicit Characterization of the Convex Envelope of a Bivariate Bilinear Function Over Special Polytopes," *Annals of Operations Research*, *Computational Methods in Global Optimization*, eds. J. B. Rosen and P. Pardalos, **25**(1-4), 197-214, 1990.

Sherali, H. D., A. Alameddine and T. S. Glickman, "Biconvex Models and Algorithms for Risk Management Problems," *American Journal of Mathematical and Management Sciences*, **14**(2&3), 197-228, 1994/95.

Sherali, H. D. and E. L. Brown, "A Quadratic Partial Assignment and Packing Model and Algorithm for the Airline Gate Assignment Problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, "Quadratic Assignment and Related Problems," eds. P. M. Pardalos and H. Wolkowicz, **16**, 343-364, 1994.

Sherali, H. D. and G. Choi, "Recovery of Primal Solutions When Using Subgradient Optimization Methods to Solve Lagrangian Duals of Linear Programs," *Operations Research Letters*, to appear, 1996.

Sherali, H. D., G. Choi, and Z. Ansari "Memoryless and Limited Memory Space Dilation and Reduction Algorithm," Working Paper, Industrial and Systems Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1996.

Sherali, H. D., G. Choi, and C. H. Tuncbilek, "A Variable Target Value Method," Working Paper, Industrial and Systems Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1995.

Sherali, H. D. and P. J. Driscoll, "On Tightening the Relaxations of Miller-Tucker-Zemlin Formulations for Asymmetric Traveling Salesman Problems," Working Paper, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1996.

Sherali, H. D., R. Krishnamurthy, and F. A. Al-Khayyal, "A Reformulation-Linearization Approach for the General Linear Complementarity Problem," *Journal of Optimization Theory and Applications*, **99**(2), 481-507, 1998.

Sherali, H. D., R. S. Krishnamurthy, and F. A. Al-Khayyal, "An Enhanced Intersection Cutting Plane Approach for Linear Complementarity Problems," *Journal of Optimization Theory and Applications*, **90**(1), 183-201, 1996.

Sherali, H. D. and Y. Lee, "Models and Algorithms for Scheduling Illuminators in Surface-to-Air Defense Strategies," Research Report, Virginia Polytechnic Institute and State University, Department of Industrial and Systems Engineering, September, 1990.

Sherali, H. D. and Y. Lee, "Sequential and Simultaneous Liftings of Minimal Cover Inequalities for GUB Constrained Knapsack Polytopes," *SIAM Journal on Discrete Mathematics*, **8**(1), 133-153, 1995.

Sherali, H. D. and Y. Lee, "Tighter Representations for Set Partitioning Problems," *Discrete Applied Mathematics*, **68**, 153-167, 1996.

Sherali, H. D., Y. Lee, and W. P. Adams, "A Simultaneous Lifting Strategy for Identifying New Classes of Facets for the Boolean Quadric Polytope," *Operations Research Letters*, **17**(1), 19-26, 1995.

Sherali, H. D., Y. Lee, and D. Boyer, "Scheduling Target Illuminators in Naval Battle-Group Anti-Air Warfare," *Naval Research Logistics*, **42**, 737-755, 1995.

Sherali, H. D. and D. C. Myers, "The Design of Branch and Bound Algorithms for a Class of Nonlinear Integer Programs," *Annals of Operations Research*, **5**, 463-484. (Special Issue on *Algorithms and Software for Optimization*), 1986.

Sherali, H. D. and D. C. Myers, "Dual Formulations and Subgradient Optimization Strategies for Linear Programming Relaxations of Mixed-Integer Programs," *Discrete Applied Mathematics*, **20**(S-16), 51-68, 1989.

Sherali, H. D. and F. Nordai, "NP-Hard, Capacitated, Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands," *Mathematics of Operations Research*, **13**(1), 32-49, 1988.

Sherali, H. D., S. Ramachandran, and S. Kim, "A Localization and Reformulation Discrete Programming Approach for the Rectilinear Distance Location-Allocation Problem," *Discrete Applied Mathematics*, **49**(1-3), 357-378, 1994.

Sherali, H. D. and C. M. Shetty, "A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Face-Cuts," *Mathematical Programming*, **19**, 14-31, 1980.

Sherali, H. D. and E. P. Smith, "An Optimal Replacement-Design Model for a Reliable Water Distribution Network System," In *Integrated Computer Applications for Water Supply and Distribution*, Research Studies Press Limited, ed. B. Coulbeck, 61-75, 1993.

Sherali, H. D. and E. P. Smith, "A Global Optimization Approach to a Water Distribution Network Problem," *Journal of Global Optimization*, **11**, 107-132, 1997.

Sherali, H. D. and C. H. Tuncbilek, "A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation-Linearization Technique," *Journal of Global Optimization*, **2**, 101-112, 1992.

Sherali, H. D. and C. H. Tuncbilek, "A Squared-Euclidean Distance Location-Allocation Problem," *Naval Research Logistics Quarterly*, **39**, 447-469, 1992.

Sherali, H. D. and C. H. Tuncbilek, "Comparison of Two Reformulation-Linearization Technique Based Linear Programming Relaxations for Polynomial Programming Problems," *Journal of Global Optimization*, **10**, 381-390, 1997.

Sherali, H. D. and C. H. Tuncbilek, "A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems," *Journal of Global Optimization*, **7**, 1-31, 1995.

Sherali, H. D. and C. H. Tuncbilek, "New Reformulation-Linearization Technique Based Relaxations for Univariate and Multivariate Polynomial Programming Problems," *Operations Research Letters*, **21**(1), 1-10, 1997.

Sherali, H. D. and O. Ulular, "A Primal-Dual Conjugate Subgradient Algorithm for Specially Structured Linear and Convex Programming Problems," *Applied Mathematics and Optimization*, **20**, 193-221, 1989.

Shetty, C. M. and H. D. Sherali, "Rectilinear Distance Location-Allocation Problem: A Simplex Based Algorithm," *Lecture Notes in Economics and Mathematical Systems, Extremal Methods and System Analysis*, **174**, 442-464, 1980.

Shiau, T.-H., "Iterative Linear Programming for Linear Complementarity and Related Problems," Computer Sciences Technical Report #507, Computer Sciences Department, University of Wisconsin-Madison, 1983.

Shor, N. A., "The Rate of Convergence of the Generalized Gradient Descent Method," *Kibernetica*, **4**(3), 79-90, 1968.

Shor, N. A., *Minimization Methods for Nondifferentiable Functions*, Translated form the Russian by K. D. Kiwiel and A. Ruszczynski, Springer-Verlag, 1985.

Shor, N. Z., "Dual Quadratic Estimates in Polynomial and Boolean Programming," *Annals of Operations Research*, **25**(1-4), 163-168, 1990.

Shor, N. Z., *Minimization Methods for Nondifferentiable Functions*, Translated from Russian by K. C. Kiwiel and A. Ruszczynski, Springer-Verlag, Berlin, 1985.

Simone, C. D., "The Cut Polytope and the Boolean Quadric Polytope," *Discrete Mathematics*, **79**, 71-75, 1989.

Simone, C. D., "Lifting Facets of the Cut Polytope," *Operations Research Letters*, **9**, 341-344, 1990.

Simone, C. D., M. Deza, and M. Laurent, "Collapsing and Lifting for the Cut Polytope," Research Report R, 265 CNRS, 1989.

Sinha, P. and A. A. Zoltners, "The Multiple-Choice Knapsack Problem," *Operations Research*, **27**, 503-515, 1979.

Spielberg, K. and U. Suhl, "An Experimental Software System for Large Scale 0-1 Problems with Efficient Data Structures and Access to MPSX/370," IBM Research Report TC 8219, White Plains, NY 10604, 1980.

Strodiot, J.-J., V. H. Nguyen, and N. Heukemes, "ε-Optimal Solution in Nondifferentiable Convex Programming and Some Related Questions," *Mathematical Programming*, **25**, 307-328, 1983.

Taha, H., *Integer Programming: Theory, Applications and Computations*, Academic Press, New York, NY, 1975.

Thieu, T. V., "A Note on the Solution of Bilinear Programming Problems by Reduction to Concave Minimization," *Mathematical Programming*, **41**, 249-260, 1988.

Thoai, N. V. and H. Tuy, "Solving the Linear Complementarity Problem Through Concave Programming," *Zh. Vychisl. Mat. I. Mat. Fiz.*, **23**(3), 602-608, 1983.

Todd, M. J., "Recent Developments and New Directions in Linear Programming," *Mathematical Programming*, eds. M. Iri and K. Tanabe, KTK Scientific Publishers, Tokyo, Japan, 109-157, 1989.

Tuncbilek, C. H., "Polynomial and Indefinite Quadratic Programming Problems: Algorithms and Applications," Ph.D. Dissertation, Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, 1994.

Tuy, H., "Concave Programming Under Linear Constraints," *Doklady Akademii Nauk SSR*, 32-35, 1964. [Translated: *Soviety Mathematics Doklady*, **5**, 1437-1440, 1964.]

Tuy, H., "Global Minimization of a Difference of Two Convex Functions," *Mathematical Programming Study*, 150-182, 1987.

Vaish, H., "Nonconvex Programming with Applications to Production and Location Problems," Unpublished Ph.D. dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, 1974.

Vaish, H. and C. M. Shetty, "The Bilinear Programming Problem," *Naval Research Logistics Quarterly*, **23**, 303-309, 1976.

Vaish, H. and C. M. Shetty, "A Cutting Plane Algorithm, for the Bilinear Programming Problem," *Naval Research Logistics Quarterly*, **24**, 83-94, 1976.

Van Roy, T. J. and L. A. Wolsey, "MPSARX, A Mathematical Programming System with an Automatic Reformulation Executor," CORE Computing Report 84-B-01, Center for Operations Research and Econometrics, Universite Catholique de Louvain, Belgium, 1984.

Van Roy, T. J. and L. A. Wolsey, "Solving Mixed Integer Programs by Automatic Reformulation," *Operations Research,* **35**, 45-57, 1987.

Van Roy, T. J. and L. A. Wolsey, "Valid Inequalities for Mixed 0-1 Programs," CORE Discussion Paper No. 8316, Center for Operations Research and Econometrics, Universite Catholique de Louvain, Belgium, 1983.

Vavasis, S. A., "Approximation Algorithms for Indefinite Quadratic Programming," *Mathematical Programming,* **57**, 279-311,1992.

Visweswaran, V. and C. A. Floudas, "Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs: II. Application of Theory and Test Problems," *Comp. and Chem. Engr.*, **14**(12), 1990.

Visweswaran, V., and C. A. Floudas, "Unconstrained and Constrained Global Optimization of Polynomial Functions in One Variable," *Journal of Global Optimization,* **2**, 73-99, 1992.

Visweswaran, V. and C. A. Floudas, "New Properties and Computational Improvement of the GOP Algorithm for Problems with Quadratic Objective Function and Constraints," *Journal of Global Optimization,* **3**, 439-462, 1993.

Volkov, E. A., *Numerical Methods*, Hemisphere Publishing, New York, N.Y., 1990.

Wang, X., and T. Chang, "An Improved Linear Lower Bound and a Fast One Dimensional Global Optimization Algorithm," Working Paper, Department of Mathematics, University of California, Davis, CA 95616, 1994.

Watson, L. T., S. C. Billups and A. P. Morgan, "Algorithm 652 HOMPACK: A Suite of Codes for Globally Convergent Homotopy Alogrithms," *ACM Transactions on Mathematical Software,* **13**(3), 281-310, 1987.

Watters, L., "Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems," *Operations Research,* **15**, 1171-1174, 1967.

Williams, H. P., "Experiments in the Formulation of Integer Programming Problems," In *Mathematical Programming Study* 2, M. L. Balinski (ed.), North-Holland Publishing Company, Amsterdam, 180-197, 1974.

Williams, H. P., *Model Building in Mathematical Programming*. John Wiley and Sons (Wiley Interscience), Second Edition, New York, NY, 1985.

Wingo, D. R., "Globally Minimizing Polynomials Without Evaluation Derivatives," *International Journal of Computer Mathematics*, **17**, 287-294, 1985.

Wolfe, P., "A Method of Conjugate Subgradients for Minimizing Nondifferentiable Functions," *Mathematical Programming Study 3*, North-Holland Publishing Co., New York, NY, 145-173, 1975.

Wolsey, L. A., "Faces for a Linear Inequality in 0-1 Variables," *Mathematical Programming,* **8**(2), 165-178, 1975.

Wolsey, L. A. "Facets and Strong Valid Inequalities for Integer Programs," *Operations Research,* **24**, 367-373, 1976.

Wolsey, L. A., "Strong Formulations for Mixed Integer Programming: A Survey," *Mathematical Programming,* **45**, 173-191, 1989.

Wolsey, L. A., "Valid Inequalities for 0-1 Knapsacks and MIPs with Generalized Upper Bound Constraints," *Discrete Applied Mathematics,* **29**, 251-262, 1990.

Ye, Y. and P. M. Pardalos, "A Class of Linear Complementarity Problems Solvable in Polynomial Time," *Linear Algebra and its Applications*, **152**, 3-17, 1991.

Zangwill, W. I., "Media Selection by Decision Programming," *Journal of Advertising Research*, **5**, 30-36, 1965.

Zemel, E., "Lifting the Facets of 0-1 Polytopes," *Mathematical Programming*, **15**, 268-277, 1978.

Zemel, E., "The Linear Multiple Choice Knapsack Problem," *Operations Research*, **28**(6), 1412-1423, 1980.

Zemel, E., "Easily Computable Facets of the Knapsack Polytope," *Mathematics of Operations Research*, **14**, 760-764, 1989.

Zikan, K., "Track Initialization and Multiple Object Tracking Problem," Working Paper, Department of Operations Research, Stanford University, Stanford, CA 94305, 1990.

Zoutendijk, G., *Methods of Feasible Directions*, Elsevier, Amsterdam, and D. Van Nostrand, Princeton, NJ, 1960.

Zowe, J., "Nondifferential Optimization," *Computational Mathematical Programming*, K. Schittkowski (ed.), NATO ASI Series, 323-355, Z-1, 1985.

Zwart, P. B., "Nonlinear Programming: Counterexamples to Two Global Optimization Algorithms," *Operations Research*, **21**(6), 1260-1266, 1973.

Nonconvex Optimization and Its Applications

1. D.-Z. Du and J. Sun (eds.): *Advances in Optimization and Approximation.* 1994
ISBN 0-7923-2785-3
2. R. Horst and P.M. Pardalos (eds.): *Handbook of Global Optimization.* 1995
ISBN 0-7923-3120-6
3. R. Horst, P.M. Pardalos and N.V. Thoai: *Introduction to Global Optimization* 1995
ISBN 0-7923-3556-2; Pb 0-7923-3557-0
4. D.-Z. Du and P.M. Pardalos (eds.): *Minimax and Applications.* 1995
ISBN 0-7923-3615-1
5. P.M. Pardalos, Y. Siskos and C. Zopounidis (eds.): *Advances in Multicriteria Analysis.* 1995 ISBN 0-7923-3671-2
6. J.D. Pintér: *Global Optimization in Action.* Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications. 1996 ISBN 0-7923-3757-3
7. C.A. Floudas and P.M. Pardalos (eds.): *State of the Art in Global Optimization.* Computational Methods and Applications. 1996 ISBN 0-7923-3838-3
8. J.L. Higle and S. Sen: *Stochastic Decomposition.* A Statistical Method for Large Scale Stochastic Linear Programming. 1996 ISBN 0-7923-3840-5
9. I.E. Grossmann (ed.): *Global Optimization in Engineering Design.* 1996
ISBN 0-7923-3881-2
10. V.F. Dem'yanov, G.E. Stavroulakis, L.N. Polyakova and P.D. Panagiotopoulos: *Quasidifferentiability and Nonsmooth Modelling in Mechanics, Engineering and Economics.* 1996 ISBN 0-7923-4093-0
11. B. Mirkin: *Mathematical Classification and Clustering.* 1996 ISBN 0-7923-4159-7
12. B. Roy: *Multicriteria Methodology for Decision Aiding.* 1996 ISBN 0-7923-4166-X
13. R.B. Kearfott: *Rigorous Global Search: Continuous Problems.* 1996
ISBN 0-7923-4238-0
14. P. Kouvelis and G. Yu: *Robust Discrete Optimization and Its Applications.* 1997
ISBN 0-7923-4291-7
15. H. Konno, P.T. Thach and H. Tuy: *Optimization on Low Rank Nonconvex Structures.* 1997 ISBN 0-7923-4308-5
16. M. Hajdu: *Network Scheduling Techniques for Construction Project Management.* 1997 ISBN 0-7923-4309-3
17. J. Mockus, W. Eddy, A. Mockus, L. Mockus and G. Reklaitis: *Bayesian Heuristic Approach to Discrete and Global Optimization.* Algorithms, Visualization, Software, and Applications. 1997 ISBN 0-7923-4327-1
18. I.M. Bomze, T. Csendes, R. Horst and P.M. Pardalos (eds.): *Developments in Global Optimization.* 1997 ISBN 0-7923-4351-4
19. T. Rapcsák: Smooth Nonlinear Optimization in R^n. 1997 ISBN 0-7923-4680-7
20. A. Migdalas, P.M. Pardalos and P. Värbrand (eds.): *Multilevel Optimization: Algorithms and Applications.* 1998 ISBN 0-7923-4693-9
21. E.S. Mistakidis and G.E. Stavroulakis: *Nonconvex Optimization in Mechanics.* Algorithms, Heuristics and Engineering Applications by the F.E.M. 1998
ISBN 0-7923-4812-5

Nonconvex Optimization and Its Applications

22. H. Tuy: *Convex Analysis and Global Optimization.* 1998 ISBN 0-7923-4818-4
23. D. Cieslik: *Steiner Minimal Trees.* 1998 ISBN 0-7923-4983-0
24. N.Z. Shor: *Nondifferentiable Optimization and Polynomial Problems.* 1998 ISBN 0-7923-4997-0
25. R. Reemtsen and J.-J. Rückmann (eds.): *Semi-Infinite Programming.* 1998 ISBN 0-7923-5054-5
26. B. Ricceri and S. Simons (eds.): *Minimax Theory and Applications.* 1998 ISBN 0-7923-5064-2
27. J.-P. Crouzeix, J.-E. Martinez-Legaz and M. Volle (eds.): *Generalized Convexitiy, Generalized Monotonicity: Recent Results.* 1998 ISBN 0-7923-5088-X
28. J. Outrata, M. Kočvara and J. Zowe: *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints.* 1998 ISBN 0-7923-5170-3
29. M.A. Ridley: *Lowering the Cost of Emission Reduction: Joint Implementation in the Framework Convention on Climatic Change.* 1998 ISBN 0-7923-4914-8
30. J.F. Bard: *Practical Bilevel Optimization.* Algorithms and Applications. 1999 ISBN 0-7923-5458-3
31. H.D. Sherali and W.P. Adams: *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* 1999 ISBN 0-7923-5487-7

KLUWER ACADEMIC PUBLISHERS – DORDRECHT / BOSTON / LONDON