The Polyhedral Structure of Certain Combinatorial Optimization Problems with Application to a Naval Defense Problem

by

Youngho Lee

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Industrial and Systems Engineering

APPROVED:

Hanif D. Sherali, Chairman

Monte B. Boisen, Jr

Monte B. Boisen

C. Patrick Koelling

Joel A. Nachlas

Jeffrey D. Tew

July, 1992

Blacksburg, Virginia

The Polyhedral Structure of Certain Combinatorial Optimization Problems with Application to a Naval Defense Problem

by

Youngho Lee

Hanif D. Sherali, Chairman

(ABSTRACT)

This research deals with a study of the polyhedral structure of three important combinatorial optimization problems, namely, the generalized upper bounding (GUB) constrained knapsack problem, the set partitioning problem, and the quadratic zero-one programming problem, and applies related techniques to solve a practical combinatorial naval defense problem.

In Part I of this research effort, we present new results on the polyhedral structure of the foregoing combinatorial optimization problems. First, we characterize a new family of facets for the GUB constrained knapsack polytope. This family of facets is obtained by sequential and simultaneous lifting procedures of minimal GUB cover inequalities. Second, we develop a new family of cutting planes for the set partitioning polytope for deleting any fractional basic feasible solutions to its underlying linear programming relaxation. We also show that all the known classes of valid inequalities belong to this family of cutting planes, and hence, this provides a unifying framework for a broad class of such valid inequalities. Finally, we present a new class of facets for the boolean quadric polytope, obtained by applying a simultaneous lifting procedure.

The strong valid inequalities developed in Part I, such as facets and cutting planes, can be implemented for obtaining exact and approximate solutions for various combinatorial optimization problems in the context of a branch-and-cut procedure. In particular, facets and valid cutting planes developed for the GUB constrained knapsack polytope and the set partitioning polytope can be directly used in generating tight linear programming relaxations for a certain scheduling polytope that arises from a combinatorial naval defense problem. Furthermore, these tight formulations are intended not only to develop exact solution algorithms, but also to design powerful heuristics that provide good quality solutions within a reasonable amount of computational effort.

Accordingly, in Part II of this dissertation, we present an application of such polyhedral results in order to construct effective approximate and exact algorithms for solving a naval defense problem. In this problem, the objective is to schedule a set of illuminators in order to strike a given set of targets using surface-to-air missiles in naval battle-group engagement scenarios. The problem is conceptualized as a production floor shop scheduling problem of minimizing the total weighted flow time subject to time-window job availability and machine-downtime unavailability side constraints. A polynomial-time algorithm is developed for the case when all the job processing times are equal (and unity without loss of generality) and the data are all integer. For the general case of scheduling jobs with unequal processing times, we develop three alternative formulations and analyze their relative strengths by comparing their respective linear programming relaxations. The special structures inherent in a particular strong zero-one integer programming model of the problem enable us to derive some classes of strong valid inequalities from the facets of the GUB constrained knapsack polytope and the set-packing polytope. Furthermore, these special structures enable us to construct several effective approximate and exact algorithms that provide solutions within specified tolerances of optimality, with an effort that admits real-time processing in the naval battle-group engagement scenario. Computational results are presented using suitable realistic test data.

Acknowledgements

I would like to express my sincere appreciation to Professor Hanif D. Sherali, my advisor, whose genuine concern showed me what it means to be a teacher and whose insight and pointed guidance kept me on the right track throughout the research effort. I especially thank him for helping me open my eyes in the area of mathematical programming.

A special note of thanks is extended to the members of my dissertation committee, Dr. Monte B. Boisen, Dr. C. Patrick Koelling, Dr. Joel A. Nachlas, Dr. Subhash C. Sarin, and Dr. Jeffrey D. Tew, for generously sharing their time and knowledge along with suggestions and comments. Also, I thank Dr. Donald D. Boyer at the Naval Surface Warfare Center whose feedback helped me get ideas and research directions, especially for the second part of this dissertation.

I want to dedicate this dissertation to my family:

My mother, mother-in-law, and father-in-law, whose faith in me have never wavered. Without their prayers and unyielding support, this research would not become a reality. Sonya, my wife, whose love always kept me alive. Her endurance and sacrifice are building blocks for this dissertation. For three years, Seung-Bin, my son, was frequently ignored while I spent evenings and weekends studying and writing. I would like to give all of them my love and thanks.

Finally, I would like to gratefully acknowledge the support of the Naval Surface Warfare Center in conducting this research effort.

Table of Contents

Introduction: Motivation and Objective

In this research effort, we study the polyhedral structure of the convex hull of feasible solutions for various important classes of combinatorial optimization problems and the application of polyhedral combinatorics procedures to develop effective approximate and exact solution algorithms for specific problem instances. Accordingly, Part I of this research deals with new results on the polyhedral structure of three important combinatorial optimization problems. These problems are the generalized upper bounding (GUB) constrained knapsack problem, the set partitioning problem, and the quadratic zero-one programming problem. In Part II of this research, we develop effective approximate and exact algorithms for solving a combinatorial naval defense problem.

Combinatorial optimization problems abound in many practical real-world applications (see Nemhauser and Wolsey, 1988, Parker and Rardin, 1988, and Salkin, 1975). These applications include operational problems that arise in the context of distribution, production scheduling, machine sequencing, location-allocation, and facility maintenance situations, planning problems such as capital budgeting and production planning, as well as design problems such as communication network design, circuit design, and the design of production systems.

Although pure cutting plane algorithms have been developed to handle such combinatorial optimization problems, branch-and-bound algorithms based on linear programming relaxations have enjoyed a far greater success in effectively solving practical instances of such problems. However, in the last 10-15 years, there has been considerable progress in using cutting planes in combination with the branch-and-bound procedure for certain classes of zero-one integer programming problems. An example of these problems is the symmetric traveling salesman problem (see Crowder et al., 1983, Johnson et al., 1985, and Lawler et al., 1985). As demonstrated by Padberg and Rinaldi (1987, 1991), cutting planes play a key role in the development of effective algorithms of this type, known as branch-and-cut algorithms. For pure zero-one integer programming problems, one way of generating such cutting planes is to use the facets of the knapsack problem obtained by considering each constraint separately. This approach was successfully implemented by Crowder et al.(1983) for solving large-scale pure zero-one integer programming problems without any special structures. A similar approach was used by Van Roy and Wolsey (1987) for zero-one mixed integer programming problems. The success of such algorithms strongly depends on the strength or tightness of the reformulated linear programming representation obtained by adding strong cutting planes or auxiliary variables during the branching process. Consequently, for the last fifteen years, most of the research effort in the area of combinatorial optimization has concentrated on investigating the polyhedral structure of various problems in an attempt to find new families of strong cutting planes or facet-defining inequalities. Several remarkable advances have been reported and implemented along with the development of the theory of polyhedral combinatorics. (For a survey of polyhedral combinatorics results, see Bachem and Grotschel, 1982, Nemhauser and Wolsey, 1988, and Pulleyblank, 1983.)

In particular, the development of new polyhedral results for various combinatorial optimization problems has identified some crucial and critical facial structures of various combinatorial optimization problems. A list of important polyhedral results in the literature includes those for the knapsack polytope (Balas, 1975, Balas and

Zemel, 1979, Gottlieb and Rao, 1988, Hammer et al., 1975, Laurent, 1989, Padberg, 1979, 1980, and Wolsey, 1975), the traveling salesman polytope (Lawler et al., 1985, and Nemhauser and Wolsey, 1988 for a survey), the vertex packing polytope (Nemhauser and Trotter, 1974, 1975, Padberg, 1973, 1979, 1980, Picard and Queyranne, 1977, and Trotter, 1975), the set packing polytope (Balas and Padberg, 1976, Balas and Zemel, 1977, Chvatal, 1975, Hammer, 1975, Laurent, 1989, and Padberg, 1973, 1979, 1980), the set covering polytope (Balas and Ng, 1989 and Cornuejols and Sassano, 1989), the set partitioning polytope (Balas, 1977), the generalized assignment polytope (Gottlieb and Rao, 1990.a, 1990.b), the cut polytope (Barahona et al., 1985, Barahona and Mahjoub, 1986, Deza and Laurent, 1988, 1989.a, 1989.b, Deza et al., 1989, Simone, 1989, 1990, and Simone et al., 1989), the quadric polytope (Padberg, 1989 and Boros and Hammer, 1990), and the fixed-charge network problem (see Nemhauser and Wolsey, 1988 for a survey). The main theme of these polyhedral results is to use the structure of coefficients in the constraints to derive strong valid inequalities for the convex hull of feasible solutions. One of the main tools for deriving these strong valid inequalities is to use sequential and simultaneous lifting procedures that lift lower dimensional valid inequalities (facets) into higher dimensional valid inequalities (facets) (see Padberg, 1975, Wolsey, 1976, and Zemel, 1980). However, the determination of families of these combinatorial strong valid inequalities is more of an art than a formal methodology. Hence, most results in this research direction are restricted to finding classes of facets for specific well-structured zero-one integer programming problems.

Although polyhedral results for the foregoing problems have been successfully implemented in the context of branch-and-cut algorithms, more generally applicable techniques for tightening the formulation and providing strong valid inequalities for generally structured problems have also deservedly received more attention for the last five years. In this research vein, Sherali and Adams (1989, 1990) have developed a new reformulation-linearization-technique (RLT) that can be used to tighten the formulation of linear and polynomial zero-one mixed integer programming problems. Moreover, Sherali and Adams show how one can readily construct a hierarchy of linear relaxations leading to the convex hull representation for such problems. Through the projection process, the RLT procedure produces an implicit algebraic representation of the convex hull of any given zero-one (mixed) integer programming problems that can be exploited to derive facets. Following this work, Lovasz and Schrijver (1989) have also proposed a similar hierarchy for pure zero-one integer programming problems. However, due to the computational burden of the inherent projection process, it may be prohibitive to obtain explicit representations of the resulting linear relaxations. Similarly, Balas et al.(1991) present another implicit hierarchy of relaxations for zero-one (mixed) integer programming problems. They implement a sequential convexification procedure that is known to be a special construct of Sherali and Adams (1989). Based on this procedure, Balas et al. describe and test a finitely convergent cutting plane algorithm. However, although these RLT based algorithms provide promising computational results, the analysis and identification of the polyhedral structure of general combinatorial optimization problems remains an open question for research.

Motivated by these observations, this dissertation aims to investigate the use of RLT-based methods to analyze and obtain strong valid inequalities, including facet-defining inequalities. Consequently, the principal objective of this research effort is to study the polyhedral structure of three important combinatorial optimization problems, namely, the GUB constrained knapsack problem, the set partitioning problem, and the quadratic zero-one programming problem, and to apply related techniques for solving a practical combinatorial naval defense problem.

In Part I of this research effort, we present new results on the polyhedral structure of the foregoing combinatorial optimization problems. First, we characterize a new family of facets for the GUB constrained knapsack polytope. This family of facets is obtained by sequential and simultaneous lifting procedures of minimal GUB cover inequalities. Second, we develop a new family of cutting planes for the set partitioning polytope, which delete any fractional basic feasible solutions to its underlying linear programming relaxation. We also show that all the known classes of valid inequalities belong to this family of cutting planes, and hence, this provides a unifying framework for a broad class of such inequalities. Finally, we present a new class of facets for the boolean quadric polytope, obtained by applying a simultaneous lifting procedure. This new class of facets subsumes the well-known facets of Padberg's (1988) as special cases.

The strong valid inequalities developed in Part I, such as facets and cutting planes, can be implemented for obtaining exact and approximate solutions for various combinatorial optimization problems in the context of a branch-and-cut procedure. In particular, facets and valid cutting planes developed for the GUB constrained knapsack polytope and the set partitioning polytope can be directly used for generating tight linear programming relaxations for a certain scheduling polytope that arises from a combinatorial naval defense problem.

Accordingly, in Part II of this dissertation, we apply the polyhedral results of Part I in order to construct effective approximate and exact algorithms for solving a combinatorial naval defense problem. This problem seeks to schedule a set of illuminators in order to strike a given set of targets using surface-to-air missiles in a naval battle-group engagement scenario. The problem is conceptualized as a production floor shop scheduling problem of minimizing the total weighted flow time, subject to time-window job availability and machine-downtime unavailability side constraints. A polynomial-time algorithm is developed for the case when all the job processing times are equal (and unity without loss of generality), and the data are all integer. For the general case of scheduling jobs with unequal processing times, we develop three alternative formulations and analyze their relative strengths by comparing their respective linear programming relaxations. The special structures inherent in a particular strong zero-one integer programming model of the problem enable us to derive some classes of strong valid inequalities from the facets of the GUB constrained knapsack polytope and the set-packing polytope. Furthermore, these special structures enable us to construct several effective approximate and exact algorithms that provide solutions within specified tolerances of optimality, with an effort that admits real-time processing in the naval battle-group engagement scenario. Promising computational results are presented using suitable realistic test data.

The Polyhedral Structure of Combinatorial Optimization Problems

1. Sequential and Simultaneous Liftings of Minimal Cover Inequalities for GUB Constrained Knapsack Polytopes

1.1. Introduction

Consider the generalized upper bounding (GUB) constrained, or multiple-choice, knapsack problem defined as follows.

$$\textbf{GKP :} \quad \text{Minimize } \{ \sum_{j\in N} c_j x_j \ : \ \sum_{i\in M}\sum_{j\in N_i} a_j x_j \geq b, \ \sum_{j\in N_i} x_j \leq 1 \ \forall \ i\in M, \ \ x_j \in (0,1) \ \forall \ j\in N\},$$

where the data are all integer, $N=\{1,...,n\}$, $M=\{1,...,m\}$, and where $\bigcup_{i\in M} N_i \equiv N$, with $N_i \bigcap N_j \ = \ \phi$ 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 $0 < a_j \leq b \ \forall \ j\in N$. Hence, without loss of generality, we will also assume that $b > 0$ and that $0 < a_j \leq b \ \forall \ j\in N$. Note that if $|N_i| = 1 \ \forall i\in M$, then problem GKP is the *ordinary 0-1 knapsack problem.*

There are many useful applications of model GKP. As suggested in Sinha and Zoltners (1979), this model is appropriate for capital budgeting problems having a single resource, and where the investment opportunities are divided into disjoint subsets. Balintify et al. (1978) identify another application 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, model GKP 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. Moreover, Martin and Schrage (1985), and Hoffman and Padberg (1991) present logical implications that can be derived from GUB constrained knapsack polytopes in the context of coefficient-reductions for 0-1 integer programming problems. In the same spirit, model GKP can also be 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. In this regard, Wolsey (1990) defines a "GUB cover" inequality for problem GKP, and presents some implementations of GUB cover inequalities for solving machine sequencing problems, generalized assignment problems, and variable-upper-bounded flow problems with GUB constraints. However, we will be concerned in this chapter with the polyhedral properties of convex hull of solutions feasible to problem GKP through an extension of the well-known minimal cover inequalities for the ordinary 0-1 knapsack polytope.

The following is an outline of this chapter. In Section 2, we present a class of valid inequalities for problem GKP obtained by an extension of the minimal cover inequalities for the ordinary knapsack polytope. We also develop a necessary and sufficient condition for such an inequality to define a facet of a lower dimensional polytope.

Subsequently, in Section 3, we develop a sequential lifting procedure in order to obtain a family of facets. The sequential lifting procedure developed herein computes lifted coefficients of the variables in each GUB set simultaneously, in contrast with the usual sequential lifting procedure that lifts only one variable at a time. Moreover, we show that this sequential lifting procedure can be implemented in polynomial time of complexity O(nm). In Section 4, we use the Reformulation-Linearization-Technique of Sherali and Adams (1990) to easily characterize facets obtainable through a simultaneous lifting procedure. This characterization enables us to derive lower and upper bounds on the lifted coefficients. In Section 5, for the special case of the ordinary knapsack polytope, we use this analysis to further tighten a known lower bound on the coefficients of lifted facets derived from minimal covers. In Section 6, we develop a generalization of the GUB cover inequality and discuss its implementation. Finally, Section 7 concludes this chapter.

1.2. Valid Inequalities from Minimal GUB Covers

Denote the constraint set of model GKP as

$$X \;\equiv\; \{x \in (0,1)^n \;:\; \sum_{i \in M} \sum_{j \in N_i} a_j x_j \;\geq\; b\,, \quad \sum_{j \in N_i} x_j \;\leq\; 1 \quad \forall\, i \in M\}.$$

We start by introducing some notation. For $K \subseteq N$, let $M_K = \{i \in M : j \in N_i, \text{ for some } j \in K\}$. 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 $j(i) \in \underset{j \in N_i}{\operatorname{argmax}}(a_j)$. Similarly, given any $B \subseteq N$, for each $i \in M_B$, define a *key index* $j_B(i)$ such that $j_B(i) \in \underset{j \in N_i \cap B}{\operatorname{argmax}}(a_j)$. 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 $B_- = B - B_+$.

Let us suppose that for each $k \in N$

$$a_k + \sum_{l \in (M - M_k)} a_{j(l)} \geq b \,. \tag{1.1}$$

Otherwise, $x_k = 0$ in every feasible solution to X. Denoting the convex hull operation by conv(.), 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 1.1 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 $x^{(k,i)} \equiv \{\ x_k = 1, x_j = 1$ for $j = j(p),\ \forall p \in M - i,\ x_j = 0$ 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 $j = 1, ..., n - |M_0| + 1$. Without loss of generality, let $x^{n-|M_0|+1} \equiv \{x_j = 1$ for $j \in N_+,\ x_j = 0$ otherwise$\}$.

Construct a matrix D whose row vectors are $x^j - x^{n-|M_0|+1}$, $j = 1,...,n - |M_0|$. Then D can be readily seen to possess a block-diagonal structure, with the rows corresponding to each block being linearly independent. Hence, x^j, $j = 1, ..., n - |M_0| + 1$, are affinely independent. This completes the proof. ■

Suppose that dim(GUBKP) is less than n, i.e., $|M_0| \geq 1$. Then, we can write each inequality constraint $i \in M_0$ as an equality constraint. That is, for each $i \in M_0$, we have that $x_{jmin(i)} = 1 - \sum_{j \in (N_i - jmin(i))} x_j$ where $jmin(i) \in \underset{j \in N_i}{argmin}(a_j)$. By replacing the variable $x_{jmin(i)}$ for $i \in M_0$ and retaining the inequality constraints of the form $\sum_{j \in (N_i - jmin(i))} x_j \leq 1$ for $i \in M_0$, we get a full dimensional subpolytope of dimension of $n - |M_0|$. Hence, without loss of generality, we can assume henceforth that GUBKP is a full dimensional polytope.

Now, for a given $H \subseteq N$, define $X(H) = X \bigcap \{x \in (0,1)^n: x_{j(i)} = 1, \forall i \in M_H\}$. In a manner similar to the proof of Proposition 1.1, we have the following holding true.

Corollary 1.1 Suppose that GUBKP is full dimensional. Then, for any $A \subseteq M$, $\dim(\mathrm{conv}(X(\bigcup_{i \in A} N_i))) = |\bigcup_{i \in (M - A)} N_i|$. ■

A *facet* of an n-dimensional polytope is a maximal proper face, or, equivalently, a face of dimension n-1. The inequality $\sum_{j \in N} \pi_j x_j \geq \pi_0$, denoted as $\pi x \geq \pi_0$, is a *facet defining inequality* if (i) $x \in$GUBKP implies $\pi x \geq \pi_0$, i.e., $\pi x \geq \pi_0$ is *valid* for GUBKP, and (ii) there exist exactly n affinely independent vertices x^i of GUBKP satisfying $\pi x^i = \pi_0$, $i = 1, ..., n$. Since GUBKP is a full dimensional polytope, there exist a set of facets uniquely defined up to a positive scalar multiple. Hence, the facet defining inequalities are themselves referred to as facets of GUBKP.

The following two propositions can be readily established.

Proposition 1.2 For each $j \in N_-$, the inequality $x_j \geq 0$ is a facet of GUBKP.

Proposition 1.3 The GUB constraints $\sum_{j \in N_i} x_j \leq 1$, $i \in M$, are facets of GUBKP.

We now define a generalization of the well-known minimal cover inequality of Balas (1975). 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

$$\sum_{j \in K} x_j \geq 1. \tag{1.2}$$

By the definition of a minimal GUB cover, it follows that the minimal GUB cover inequality (1.2) is a valid inequality for GUBKP.

For a minimal GUB cover K, we define $R = \{j \in \overline{K} : a_j \geq \max_{j \in K}(a_j)\}$. An *extension of the minimal GUB cover K* of X, denoted by E(K), is defined as $E(K) = K \bigcup S$, where $S = \bigcup_{i \in M_R} N_i$.

Proposition 1.4 If K is a minimal GUB cover of X, then the inequality defined as

$$\sum_{j \in E(K)} x_j \geq 1 + |M_R| \tag{1.3}$$

is a valid inequality for GUBKP. Moreover, this inequality dominates the minimal GUB cover inequality if $R \neq \phi$.

Proof. Suppose that there exists a $\bar{x} \in X$ such that $\sum_{j \in E(K)} \bar{x}_j \leq |M_R|$. From the definition of E(K), the largest $|M_R|$ coefficients a_j among $\{a_j : j \in E(K)_+\}$ are the $\{a_j : j \in S_+\}$. But $\sum_{i \in M_{\overline{K}}} a_{j(i)} < b$ implies that even with such a solution $\bar{x} \equiv \{\bar{x}_j = 1$ for $j \in S_+$, $\bar{x}_j = 1$ for $j \in (\overline{K} - S)_+$, $\bar{x}_j = 0$ otherwise$\}$, we cannot obtain a feasible solution to X, a contradiction. Hence, the inequality (1.3) must always hold true for any $\bar{x} \in X$. Moreover, note that the inequality (1.3) can be represented as $\sum_{j \in K} x_j \geq 1 + \sum_{i \in M_R} (1 - \sum_{j \in N_i} x_j)$. Since $\sum_{j \in N_i} x_j \leq 1$, for each $i \in M_R$, the inequality (1.3) dominates the minimal GUB cover inequality if $R \neq \phi$, and this completes the proof. ■

The idea of using strong minimal covers to generate nondominated extensions as for the case of the ordinary knapsack problem can be readily extended to the GUB constrained situation as follows. We will call a minimal GUB cover K *strong* if either $E(K) = N$, or else no set of the form $S = \bigcup_{i \in (M_K - M_{j_1}) \cup \{p\}} N_i$ for any $p \in M_{\overline{E(K)}}$ is a cover, where $j_1 \in \operatorname{argmax}_{j \in K}(a_j)$. That is, a set $K \subseteq N$ is a *strong GUB cover* of X if K is a minimal GUB cover for which, if $E(K) \neq N$, then $a_{j_1} + \sum_{i \in (M_{\overline{E(K)}} - p)} a_{j(i)} \geq b \ \forall p \in M_{\overline{E(K)}}$. Hence, a minimal GUB cover K is strong if there exists no minimal GUB cover of the same size as K whose extension strictly contains that of K.

We now consider another strengthening procedure for the minimal GUB cover inequality.

Proposition 1.5 If K is a minimal GUB cover, and (K_1, K_2) is a partition of K with $K_2 \neq \phi$ such that

$$\max \{\sum_{j \in K_2} a_j x_j : \sum_{j \in N_i \cap K_2} x_j \leq 1 \ \forall \ i \in M_{K_2},\ x_j \in (0,1) \ \forall \ j \in K_2\} \leq b - \sum_{i \in M_{\overline{K}}} a_{j(i)} - 1, \qquad (1.4)$$

then the following inequality

$$\sum_{j \in K_1} x_j \geq 1 \tag{1.5}$$

is valid for GUBKP and dominates the minimal GUB cover inequality $\sum_{j \in K} x_j \geq 1$. Moreover, if $\min_{j \in K_1}(a_j) + \sum_{j \in M_{\overline{K}}} a_{j(i)} \geq b$, then the inequality (1.5) is a facet of conv$(X(K_2, \overline{K}))$ where $X(K_2, \overline{K}) = X \bigcap \{x \in (0,1)^n : x_j = 0 \ \forall j \in K_2,\ x_{j(i)} = 1 \ \forall i \in M_{\overline{K}}\}$.

Proof. (i) It is readily shown that the inequality (1.5) is valid for GUBKP. (ii) Since $\min_{j \in K_1}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$, the unit vectors e_j, for $j \in K_1$, are feasible to conv$(X(K_2, \overline{K}))$, and moreover, they satisfy (1.5) as an equality, and are linearly independent. Hence, the inequality (1.5) is a facet of conv$(X(K_2, \overline{K}))$. This completes the proof. ■

Note that the condition (1.4) of Proposition 1.5 can be restated as $\sum_{i \in M_{K_2}} \max_{j \in N_i \bigcap K_2}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} < b$. Furthermore, if no such partition with $K_2 \neq \phi$ exists, i.e., if $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$, then we have the following result.

Proposition 1.6 For a minimal GUB cover K, the minimal GUB cover inequality (1.2) is a facet of conv$(X(\overline{K}))$ if and only if $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$.

Proof. Suppose that the minimal GUB cover inequality is a facet of conv$(X(\overline{K}))$. Then, by the structure of (1.2), there must exist $|K|$ unit vectors e_j, one for each $j \in K$, which belong to conv$(X(\overline{K}))$. Hence, we have that $a_j \geq b - \sum_{i \in M_{\overline{K}}} a_{j(i)}$ for all $j \in K$, and so, $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$. Conversely, if $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} \geq b$, then the unit vectors e_j, $j \in K$, are feasible to conv$(X(\overline{K}))$, satisfying (1.2) as an equality, and are linearly independent. Hence, the valid inequality (1.2) is a facet of conv$(X(\overline{K}))$. This completes the proof. ■

Example 1.1 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 1.1. 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 $K = \{1, 2, 3, 4\}$ and a minimal GUB cover inequality is $x_1 + x_2 + x_3 + x_4 \geq 1$. This minimal GUB cover does not admit any extension. However, consider a partition of K such that $K_1 = \{2, 4\}$ and $K_2 = \{1, 3\}$. Then a strengthened minimal GUB cover inequality of the form (1.5) is $x_2 + x_4 \geq 1$, which is a facet of $\mathrm{conv}(X(K_2, \overline{K}))$, where $X(K_2, \overline{K}) \equiv X \cap \{x \in (0,1)^8 : x_1 = x_3 = 0,\ x_6 = x_8 = 1\}$. Note that the foregoing minimal cover is not strong, since an extension of the foregoing minimal GUB cover $K = \{3, 4, 5, 6\}$ is $E(K) = \{1, 2, 3, 4, 5, 6\} \supset \{1, 2, 3, 4\}$, and the extended inequality of the type (1.3) is $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 2$. This minimal GUB cover $K = \{3, 4, 5, 6\}$ can be verified to be strong. Moreover, since $\min_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} = 9 = b$, by Proposition 1.6, the corresponding minimal GUB cover inequality $x_3 + x_4 + x_5 + x_6 \geq 1$ is a facet of $\mathrm{conv}(X(\overline{K})) \equiv X \cap \{x \in (0,1)^8 : x_2 = 1,\ x_8 = 1\}$.

We now consider the association between canonical facets and the inequalities (1.5) of GUBKP.

Proposition 1.7 If for some $H \subseteq N$, the inequality $\sum_{j \in H} x_j \geq 1$ is a facet of GUBKP, then there exists a GUB cover K ($H \subseteq K$) such that within K, H is a minimal set satisfying $\sum_{i \in M_H} \max_{j \in N_i - H}(a_j) + \sum_{i \in M_{\overline{K}}} a_{j(i)} < b$.

Proof. Since $\sum_{j \in H} x_j \geq 1$ is valid for GUBKP, we have that $\sum_{j \in \bar{H}_+} a_j \leq b - 1$, which can be restated as $\sum_{i \in M - M_H} a_{j(i)} + \sum_{i \in M_H} \max_{j \in (N_i - H)} (a_j) \leq b - 1$. Since $\sum_{i \in M - M_H} a_{j(i)} \leq b - 1$, $K = \bigcup_{i \in M_H} N_i$ is a GUB cover set that contains H and satisfies the condition of the proposition. If H is not minimal, then there exists an $s \in H$ such that $H' = H - s$ satisfies the condition of the proposition. Then, by Proposition 1.5, $\sum_{j \in H'} x_j \geq 1$ is valid for GUBKP. Since $\sum_{j \in H'} x_j \geq 1$ strictly dominates $\sum_{j \in H} x_j \geq 1$, we have a contradiction, and this completes the proof. ■

1.3. Sequentially Lifted Facets from Minimal GUB Covers

We now consider a polynomial-time strengthening procedure which sequentially lifts a given minimal GUB cover inequality. This sequential lifting procedure actually conducts a simultaneous lifting of the minimal GUB cover inequality using each GUB set of variables N_p, in turn, for $p \in M_{\bar{K}}$, where K is a minimal GUB cover of X. Noting that the minimal GUB cover inequality (1.2) is valid for $\text{conv}(X(\bar{K}))$, we obtain a tighter valid inequality. In particular, if the condition of Proposition 1.6 holds, then we show that the resulting inequality is a facet of GUBKP.

Let us begin by defining some notation. For a given $p \in M_{\bar{K}}$, define η^t, for $t \equiv j(p)$, as

$$\begin{aligned} \eta^t &\equiv \min\{\sum_{j \in K} x_j : x \in X, \quad x_j = 0 \ \forall\, j \in N_p\} \\ &\equiv \min\{\sum_{i \in M_K} x_{j(i)} : \sum_{i \in M_K} a_{j(i)} x_{j(i)} \geq b - \sum_{i \in (M_{\bar{K}} - p)} a_{j(i)},\ x_{j(i)} \in (0,1) \ \forall\, i \in M_K\}, \end{aligned} \tag{1.6}$$

and define ζ^s, for each $s \in (N_p - j(p))$, as

$$\begin{aligned}\zeta^{s} &\equiv \min\{\sum_{j \in K} x_j : x \in X, \quad x_s = 1\} \\ &\equiv \min\{\sum_{i \in M_K} x_{j(i)} : \sum_{i \in M_K} a_{j(i)} x_{j(i)} \geq b - a_s - \sum_{i \in (M_{\bar{K}} - p)} a_{j(i)}, \; x_{j(i)} \in (0,1) \;\; \forall\, i \in M_K\}.\end{aligned} \tag{1.7}$$

Since K is a minimal GUB cover of X, and since GUBKP is full dimensional, and by (1.1), note that $1 \leq \eta^t \leq |M_K|$ and $1 \leq \zeta^s \leq \eta^t \;\forall\, s \in (N_p - t)$.

Proposition 1.8 (i) For a given minimal GUB cover K of X, the following inequality, defined for some $p \in M_{\bar{K}}$ as

$$\sum_{j \in K} x_j + \sum_{s \in (N_p - t)} \alpha_s x_s + \alpha_t x_t \geq 1 + \alpha_t \tag{1.8}$$

where $t \equiv j(p)$, is a valid inequality for GUBKP for any $\alpha_t \leq \eta^t - 1$ and for any $\alpha_s \geq -\zeta^s + \alpha_t + 1 \;\; \forall\, s \in (N_p - t)$.

(ii) Moreover, if $\alpha_t = \eta^t - 1$, $\alpha_s = \eta^t - \zeta^s \;\; \forall\, s \in (N_p - t)$, and $\min_{j \in K}(a_j) + \sum_{i \in M_{\bar{K}}} a_{j(i)} \geq b$, then the inequality (1.8) is a facet of $\mathrm{conv}(X(\overline{K} - N_p))$.

Proof. (i) Since the minimal GUB cover inequality (1.2) is valid for GUBKP, and since (1.8) coincides with (1.2) when $x_t = 1$ and $x_s = 0$ for $s \in (N_p - t)$, it is sufficient to show for establishing the validity of (1.8) that if $\bar{x} \in X$ and has $\bar{x}_s = 1$ for any $s \in (N_p - t)$, or if $\tilde{x} \in X$ and has $\tilde{x}_j = 0 \;\forall\, j \in N_p$, then such $\bar{x}$ and $\tilde{x}$ satisfy (1.8). That is, $\sum_{j \in K} \bar{x}_j + \alpha_s \geq 1 + \alpha_t$, and $\sum_{j \in K} \tilde{x}_j \geq 1 + \alpha_t$. By the definition of the quantities α_t and α_s, in the first case, we have, using (1.7), that $\sum_{j \in K} \bar{x}_j + \alpha_s \geq \zeta^s - \zeta^s + \alpha_t + 1 \geq 1 + \alpha_t$, and in the second case, we have, using (1.6), that $\sum_{j \in K} \tilde{x}_j \geq \eta^t \geq 1 + \alpha_t$. Hence, the inequality (1.8) is valid for GUBKP.

(ii) Since the minimal GUB cover inequality (1.2) is a facet of conv(X($\overline{K}$)) by Proposition 1.6, and since conv(X($\overline{K}$)) is of full dimension |K| by Corollary 1.1, there exist |K| linearly independent vertices of conv(X($\overline{K}$)), indexed by x^j, $j = 1,\ldots,|K|$, which satisfy the inequality (1.2) as an equality. Since $x^j_{j(i)} = 1$, for $i \in M_{\overline{K}}$, $j = 1,\ldots,|K|$, these vertices also satisfy the inequality (1.8) as an equality and moreover, these vertices belong to X($\overline{K} - N_p$). Now, for each $s \in (N_p - t)$, let $\hat{x}^s$ be a solution of (1.7) such that $\zeta^s = \sum_{j \in K} \hat{x}^s_j$. Note that $\hat{x}^s \in X(\overline{K} - N_p)$. Also, for $t = j(p)$, let $\hat{x}^t$ be a solution of (1.6) such that $\eta^t = \sum_{j \in K} \hat{x}^t_j$, and note that $\hat{x}^t \in X(\overline{K} - N_p)$. Moreover, $\hat{x}^s$ and $\hat{x}^t$ satisfy the inequality (1.8) as an equality when $\alpha_s = \eta^t - \zeta^s$, and $\alpha_t = \eta^t - 1$.

By Corollary 1.1, we have that dim(conv(X($\overline{K} - N_p$))) = $|K \cup N_p|$. Let $\tilde{X} \equiv \{$ $(x^j, 1)$ for $j = 1,\ldots, |K|$, $(\hat{x}^t, 1)$, $(\hat{x}^s, 1)$ for $s \in (N_p - t)$ $\}$ be the set of vectors obtained by adding a new component having value 1 to each vector x^j, $\hat{x}^t$, and $\hat{x}^s$ in the collection as shown. Let us show that the vectors in $\tilde{X}$ are linearly independent. On the contrary, suppose that these vectors are linearly dependent. Then, there exists a set of multipliers $\{ (\lambda_j$, for $j = 1,\ldots, |K|)$, δ_t, $(\mu_s$, for $s \in (N_p - t))\} \neq 0$ such that

$$\sum_{j=1}^{|K|} \lambda_j x^j + \delta_t \hat{x}^t + \sum_{s \in (N_p - t)} \mu_s \hat{x}^s = 0 \quad \text{and} \quad \sum_{j=1}^{|K|} \lambda_j + \delta_t + \sum_{s \in (N_p - t)} \mu_s = 0. \qquad (1.9)$$

Since for each $s \in (N_p - t)$, $\hat{x}^t_s = 0$, $x^j_s = 0$ for $j = 1,\ldots, |K|$, and $\hat{x}^s_s = 1$, it follows that $\mu_s = 0$ $\forall$ $s \in (N_p - t)$. Now, if $\delta_t = 0$, then by the linear independence of x^j, $j = 1,\ldots,|K|$, we would have $\lambda_j = 0$, $\forall$ $j = 1,\ldots,|K|$, a contradiction. Hence, without loss of generality, suppose that $\delta_t = -1$, so that (1.9) becomes $\hat{x}^t = \sum_{j=1}^{|K|} \lambda_j x^j$ and $\sum_{j=1}^{|K|} \lambda_j = 1$. Now, since $\hat{x}^t_t = 0$, $x^j_t = 1$ for $j = 1,\ldots, |K|$, it follows that $\sum_{j=1}^{|K|} \lambda_j = 0$, which is a contradiction. Hence, the vectors $\{(x^j,$ for $j = 1,\ldots, |K|)$, $\hat{x}^t$, $(\hat{x}^s$ for $s \in (N_p - t))\}$ are affinely independent. This completes the proof. ■

Proposition 1.9 Let K be a minimal GUB cover such that the corresponding minimal GUB cover inequality (1.2) gives a facet of $\mathrm{conv}(X(\overline{K}))$. Let $M_{\overline{K}} \equiv \{i_1, \ldots, i_k\}$ be arbitrarily ordered, where $k = |M_{\overline{K}}|$. Let $M(q) = \{i_1, \ldots, i_q\} \subseteq M_{\overline{K}}$, and let $N(q) = \bigcup_{i \in M(q)} N_i$ for $q = 1,\ldots,k$. For $q = 0$, let $M(q) = N(q) = \phi$. Let $q \in \{0,\ldots,k-1\}$ and suppose that $\sum_{j \in K \cup N(q)} \alpha_j x_j \geq \alpha_0$ is valid for GUBKP and is a facet of $\mathrm{conv}(X(\overline{K} - N(q)))$. Consider i_{q+1}. Denote $t = j(i_{q+1})$, and compute

$$\eta^t(q) = \min\{ \sum_{j \in K \cup N(q)} \alpha_j x_j : x \in X,\ x_j = 0\ \forall j \in N_{i_{q+1}} \}. \tag{1.10}$$

Also, for each $s \in N(i_{q+1}) - t$, compute

$$\zeta^s(q) = \min\{ \sum_{j \in K \cup N(q)} \alpha_j x_j : x \in X,\ x_s = 1 \}. \tag{1.11}$$

Then,

$$\sum_{j \in K \cup N(q)} \alpha_j x_j + \sum_{s \in \{N_{i_{q+1}} - t\}} (\eta^t(q) - \zeta^s(q)) x_s + (\eta^t(q) - \alpha_0) x_t \geq \eta^t(q) \tag{1.12}$$

is (i) valid for GUBKP, and (ii) is a facet of $\mathrm{conv}(X(\overline{K} - N(q+1)))$. ■

The proof of Proposition 1.9 follows the same argument used in the proof of Proposition 1.8. Note that if $q = 0$, then $\eta^t(q) = \eta^t$ and $\zeta^s(q) = \zeta^s$, as given by (1.6) and (1.7), respectively.

Remark 1.1 By Proposition 1.9, we can inductively construct a class of facets for GUBKP that can be obtained from the minimal GUB cover inequality via the above sequential lifting procedure. For the case of the ordinary knapsack polytope, Balas and Zemel (1978) exhibit that the sequential lifting procedure of Padberg (1975) that lifts one variable at a time, obtains a facet when applied to a minimal cover inequality which is a facet of the lower dimensional polytope. However, as shown in Proposition 1.8, in the presence of GUB constraints, we need to lift all the variables in each GUB constraint simultaneously in order to obtain a lifted inequality. Also, note that the sequence in which the GUB constraints are considered determines the coefficients of the lifted facet. In this sense, the lifting procedure of Proposition 1.9 remains *a sequential lifting procedure.*

Now, suppose that a sequentially lifted inequality obtained from a minimal GUB cover inequality is given by $\sum_{j\in K} x_j + \sum_{j\in \bar{K}_-} \alpha_j x_j - \sum_{j\in \bar{K}_+} \alpha_j(1-x_j) \geq 1$. This is of the form

$$\sum_{j\in K} x_j + \sum_{j\in \bar{K}_-} \alpha_j x_j + \sum_{j\in \bar{K}_+} \alpha_j x_j \geq 1 + \sum_{j\in \bar{K}_+} \alpha_j. \tag{1.13}$$

The derivation of the coefficients α_j of this lifted inequality requires the solution of a sequence of GUB constrained 0-1 knapsack problems. Furthermore, the values of the coefficients depend on the sequence in which the indices $i\in M_{\bar{K}}$ are considered. For each $i\in M_{\bar{K}}$, let α_j' denote the value of α_j when $i = i_1$, i.e., when i is taken to be the first index in $M_{\bar{K}}$. In other words, $\alpha_t' = \eta^i - 1$ for $t = j(i)$, and $\alpha_s' = \eta^i - \zeta^s \; \forall s\in(N_i - t)$. The subproblem that determines these initial coefficients has a simpler structure than the subsequent GUB constrained knapsack problems that have to be solved to find the other coefficients of the sequence, and due to this structure, the value of η^t and ζ^s can be easily obtained, as shown in the following propositions.

Proposition 1.10 Let K_h be the index set of the h largest $a_{j(i)}$ for $i \in M_K$. For a minimal GUB cover K and all $t \in \overline{K}_+$, $\eta^t = h$, where h is defined by

$$\sum_{k \in K_h} a_k \geq b - \sum_{i \in (M_{\overline{K}} - M_t)} a_{j(i)} > \sum_{k \in K_{h-1}} a_k.$$

Proof. The solution $\bar{x}$ to problem (1.6), given by $\bar{x} \equiv \{ \bar{x}_j = 1 \ \forall\, j \in K_h,\ \bar{x}_j = 0 \ \forall\, j \in K - K_h\}$, is feasible by the definition of h, and is readily seen to yield $\eta^t = h$. ■

We have the following result analogous to Proposition 1.10.

Proposition 1.11 Let K_h be the index set of the h largest $a_{j(i)}$, for $i \in M_K$. For a minimal GUB cover K and any $s \in \overline{K}_-$, $\zeta^s = h$, where h is defined by

$$\sum_{k \in K_h} a_k \geq b - a_s - \sum_{i \in (M_{\overline{K}} - M_s)} a_{j(i)} > \sum_{k \in K_{h-1}} a_k. \ ■$$

Note that for a given $p \in M_{\overline{K}}$, $\zeta^{j1} \geq \zeta^{j2}$ whenever $a_{j1} \leq a_{j2}$, for j1, j2 $\in N_p \cap \overline{K}_-$.

Corollary 1.2 For a given $p \in M_{\overline{K}}$, if $\eta^t = 1$, then $\zeta^s = 1 \ \forall\, s \in (N_p - t)$, where $t \equiv j(p)$.

Proof. Follows from the fact that $1 \leq \zeta^s \leq \eta^t \ \forall s \in N_p - t$. ■

Corollary 1.3 Let $t_1 \in \underset{i \in M_{\overline{K}}}{\operatorname{argmax}}(a_{j(i)})$. If $\eta^{t_1} = 1$, then $\zeta^{s} = 1 \ \forall \ s \in (N_i - j(i))$, $i \in M_{\overline{K}}$, and $\eta^{j(i)} = 1 \ \forall \ i \in M_{\overline{K}}$.

Proof. From (1.6), it follows that $1 \le \eta^{j(i)} \le \eta^{t_1} \ \forall i \in M_{\overline{K}}$. Using this fact along with Corollary 1.2 establishes the required result. ■

The above two corollaries facilitate to identify a class of facets for GUBKP as shown in the following proposition.

Proposition 1.12 If the minimal GUB cover inequality (1.2) is a facet of conv(X($\overline{K}$)), and if $\max_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}} - M_{t_1}} a_{j(i)} \ge b$, where $t_1 \in \underset{i \in M_{\overline{K}}}{\operatorname{argmax}}(a_{j(i)})$, then the minimal GUB cover inequality is a facet of GUBKP.

Proof. Since $\max_{j \in K}(a_j) + \sum_{i \in M_{\overline{K}} - M_{t_1}} a_{j(i)} \ge b$, we have that $\eta^{t_1} = 1$. By Corollary 1.3, all the coefficients in the lifted inequality of the form (1.8) are zeros. Examining (1.10), (1.11), and (1.12), we continue to obtain zeros for the lifted coefficients in Proposition 1.8, and so the minimal GUB cover inequality (1.2) is a facet of GUBKP. ■

Example 1.2 Consider the following constraints of a GUB constrained knapsack problem, where $X = \{ x \in (0, 1)^7 : \ 2x_1 + 4x_2 + x_3 + 2x_4 + x_5 + 2x_6 + x_7 \ge 4, \ x_1 + x_2 \le 1, \ x_3 + x_4 \le 1, \ x_5 + x_6 \le 1, \ x_7 \le 1 \}$. For a minimal GUB cover $K = \{1, 2, 3, 4\}$, the minimal GUB cover inequality is $x_1 + x_2 + x_3 + x_4 \ge 1$, which is a facet of conv(X($\overline{K}$)). Since $t_1 = 6$ and $\eta^6 = 1$, by Proposition 1.12, $x_1 + x_2 + x_3 + x_4 \ge 1$ is a facet of conv(X). Note that K is a strong GUB cover.

Finally, we show that the readily obtained coefficients α'_j, $j \in \overline{K}$, provides bounds on the coefficients β_j $j \in \overline{K}$, of arbitrary valid inequalities (not necessarily sequentially lifted inequalities) that have unit coefficients for all $j \in K$.

Proposition 1.13 If $\sum_{j \in K} x_j + \sum_{j \in \overline{K}_-} \beta_j x_j - \sum_{j \in \overline{K}_+} \beta_j (1 - x_j) \geq 1$ is valid for GUBKP, then $\beta_j \leq \alpha_j'$ $\forall j \in \overline{K}_+$, and $\beta_j \geq \alpha'_j - (\alpha'_t - \beta_t)$, where $t = j(p)$ such that $j \in N_p \cap \overline{K}_-$, $\forall p \in M_{\overline{K}}$.

Proof. Assume that for some $t \in \overline{K}_+$, $\beta_t > \alpha_t'$. From Proposition 1.8, we have that $\alpha_t' = \eta^t - 1$. Let $\overline{x}$ be an optimal solution of (1.6) such that $\eta^t = \sum_{j \in K} \overline{x}_j$. Then, $\overline{x} \in X$, but

$$\sum_{j \in K} \overline{x}_j + \sum_{j \in \overline{K}_-} \beta_j \overline{x}_j - \sum_{j \in \overline{K}_+} \beta_j (1 - \overline{x}_j) = \eta^t - \beta_t = \alpha_t' + 1 - \beta_t < 1,$$

and so, the inequality in the proposition is violated. Hence, $\beta_t \leq \alpha_t'$, $\forall t \in \overline{K}_+$. For the case of β_j, $j \in \overline{K}_-$, suppose that for some $p \in M_{\overline{K}}$, and $s \in N_p$, we have $\beta_s < \alpha'_s - (\alpha'_t - \beta_t)$ where $t \equiv j(p)$. Consider problem (1.7), and let $\overline{x} \in X$ solve this problem. Then, since $\alpha'_s - \alpha'_t = 1 - \zeta^s$, we have that

$$\sum_{j \in K} \overline{x}_j + \sum_{j \in \overline{K}_-} \beta_j \overline{x}_j - \sum_{j \in \overline{K}_+} \beta_j (1 - \overline{x}_j) = \zeta^s + \beta_s - \beta_t < \zeta^s + \alpha'_s - \alpha'_t = 1,$$

and so the inequality in the proposition is again violated. This completes the proof. ∎

We now consider the task of efficiently computing the coefficients α_j, $j \in \overline{K}$, of a sequentially lifted facet (1.13). As mentioned earlier, the task of computing α_j for $j \in \overline{K}$

involves the solution of a set of GUB constrained 0-1 knapsack problems. However, due to the special structure of these problems, we can easily obtain the coefficients α_j for all $j \in \overline{K}$ within a time complexity of $O(n|M_K|)$ by adapting the procedure due to Zemel (1989) which was proposed for the ordinary knapsack problem. Toward this end, consider the following proposition.

Proposition 1.14 Suppose that $\bar{x}$ is an optimal solution to the following problem with $z = \bar{z}$, and with all data integer valued.

$$w(z) = \max\{\sum_{j \in N} a_j x_j : \sum_{j \in N} c_j x_j \leq z, \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M, \ x_j \in (0,1) \ \forall j \in N\}.$$

Then, $\bar{x}$ is an optimal solution to the following GUB constrained knapsack problem

$$P(b): \ v(b) = \min\{\sum_{j \in N} c_j x_j : \sum_{j \in N} a_j x_j \geq b, \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M, \ x_j \in (0,1) \ \forall j \in N\}$$

for all b satisfying $w(\bar{z} - 1) < b \leq w(\bar{z})$.

Proof. Suppose that $w(\bar{z} - 1) < b \leq w(\bar{z})$. Then, $\bar{x}$ is feasible to problem P(b) since $\sum_{j \in N} a_j \bar{x}_j = w(\bar{z}) \geq b$. Now suppose that $\bar{x}$ is not optimal to P(b). Hence, there exists a feasible solution $\hat{x} \neq \bar{x}$ such that $\sum_{j \in N} c_j \hat{x}_j \leq \sum_{j \in N} c_j \bar{x}_j - 1$. Consequently, $\sum_{j \in N} c_j \hat{x}_j \leq \bar{z} - 1$, and $\sum_{j \in N} a_j \hat{x}_j \geq b$, and $\sum_{j \in N_i} \hat{x}_j \leq 1 \ \forall i \in M$, which means that $w(\bar{z} - 1) \geq b$. This contradicts the assumption that $w(\bar{z} - 1) < b$, and completes the proof. ■

Proposition 1.15 Let the function w(z) and the Problem P(b) be as defined in Proposition 1.14, for any integers z and b. Then, $v(b) = \min\{z: w(z) \geq b\}$.

Proof. Problem P(b) seeks the value v(b) = min{z : $\sum_{j\in N} c_j x_j \le z$, $\sum_{j\in N} a_j x_j \ge b$, $\sum_{j\in N_i} x_j \le 1$ $\forall$ i∈M, $x_j \in (0,1)$ $\forall$ j∈N} . Equivalently, v(b) = min { z : there exists x satisfying $\sum_{j\in N} c_j x_j \le z$, $\sum_{j\in N} a_j x_j \ge b$, $\sum_{j\in N_i} x_j \le 1$ $\forall$ i∈M, $x_j \in (0,1)$ $\forall$j∈N} . Hence, we have that v(b) = min{z : w(z) ≥ b}. ■

Suppose that we have a (partially) lifted inequality of the form

$$\sum_{j\in K} x_j + \sum_{j\in T_-} \alpha_j x_j - \sum_{j\in T_+} \alpha_j (1 - x_j) \ge 1. \tag{1.14}$$

We want to find a lifting of (1.14) with respect to the variables x_j, $\forall j \in N_p$, for some $p \in (M_{\bar{K}} - M_T)$. By Proposition 1.9, we have that for $t = j(p)$,

$$1 + \alpha_t = \min \{ \sum_{j\in K} x_j + \sum_{j\in T_-} \alpha_j x_j - \sum_{j\in T_+} \alpha_j (1 - x_j) : \sum_{j\in(K \cup T)} a_j x_j \ge b - \sum_{i\in(M_{\bar{K}} - M_T - p)} a_{j(i)},$$
$$\sum_{j\in N_i} x_j \le 1 \ \forall i\in M,\ x_j \in (0,1) \ \forall j \in N \} \ ,$$

and for each $s \in (N_p - t)$, we have,

$$1 + \alpha_t - \alpha_s = \min \{ \sum_{j\in K} x_j + \sum_{j\in T_-} \alpha_j x_j - \sum_{j\in T_+} \alpha_j (1 - x_j) : \sum_{j\in(K \cup T)} a_j x_j \ge b - a_s - \sum_{i\in(M_{\bar{K}} - M_T - p)} a_{j(i)},$$
$$\sum_{j\in N_i} x_j \le 1 \ \forall i\in M,\ x_j \in (0,1) \ \forall j \in N \} \ .$$

Let $L \equiv K \cup T$, $\bar{L} \equiv N - L$, and define

$$w_L(z) = \max \{\sum_{j\in L} a_j x_j : \sum_{j\in K} x_j + \sum_{j\in T_-} \alpha_j x_j - \sum_{j\in T_+} \alpha_j(1 - x_j) \le z, \quad \sum_{j\in N_i} x_j \le 1 \ \forall i\in M,\ x_j\in(0,1)\ \forall j\in N\}. \tag{1.15}$$

By Proposition 1.15, we have that

$$1 + \alpha_t = \min\{z : w_L(z) \ge b - \sum_{i\in(M_L^- - p)} a_{j(i)}\}, \text{ and}$$

$$1 + \alpha_t - \alpha_s = \min\{z : w_L(z) \ge b - a_s - \sum_{i\in(M_L^- - p)} a_{j(i)}\}.$$

Hence, we can efficiently obtain the coefficients α_j for $j\in N_p$, by computing $w_L(z)$ efficiently for different pertinent values of z. Toward this end, consider the following recursive equation for computing the function $w_L(z)$. Note that

$$\begin{aligned} w_{L\cup N_p}(z) = \max \ \{&\sum_{j\in L} a_j x_j + \sum_{j\in N_p} a_j x_j : \\ &\sum_{j\in K} x_j + \sum_{j\in T_-} \alpha_j x_j - \sum_{j\in T_+} \alpha_j(1 - x_j) + \sum_{s\in(N_p - t)} \alpha_s x_s - \alpha_t(1 - x_t) \le z, \\ &\sum_{j\in N_i} x_j \le 1 \ \forall i\in M,\ x_j\in(0,1)\ \forall j\in N\} \ . \end{aligned}$$

Equivalently, this yields,

$$w_{L\cup N_p}(z) = \max\{\max_{s\in(N_p - t)} [a_s + w_L(z + \alpha_t - \alpha_s)]\, ,\ [a_t + w_L(z)]\}. \tag{1.16}$$

Hence, we can compute the coefficients α_j for $j\in\overline{K}$ by using the recursive equation (1.16). Consider the problem (1.15). Let $\hat{z}$ be the smallest among the alternative optimal solutions of $\max_z(\, w_L(z)\,)$. Then, it follows that $w_L(z) = w_L(\hat{z})$ $\forall z \geq \hat{z}$. For any minimal GUB cover K, to begin with, since $w_K(z) = w_K(|M_K|)$ for all $z \geq |M_K|$, we only need to compute $w_K(z)$ for $z=1,\ldots,|M_K|$. Moreover, since $\alpha_t - \alpha_s \geq 0$ $\forall s\in(N_p - t)$, we have recursively that the value in (1.16), for each L and N_p thereafter, also remains a constant for $z \geq |M_K|$. Therefore, each function $w_L(z)$ needs to be evaluated (recursively) via (1.16) only for $z = 1, \ldots, |M_K|$. Hence, the time complexity of computing the lifted coefficients α_j, for $j\in\overline{K}$, is $O(n|M_K|)$.

1.4. Simultaneously Lifted Facets from Minimal GUB Covers

We now consider an implementation of the Reformulation-Linearization-Technique (RLT) (see Sherali and Adams, 1990) to characterize a class of valid inequalities (facets) of GUBKP, obtainable via a simultaneous lifting of minimal GUB cover inequalities. Toward this end, define a set F corresponding to feasible solutions for GKP as

$$F \equiv \{\, J\subseteq N : \sum_{j\in J} a_j \geq b, \quad |J \cap N_i| \leq 1 \quad \forall\, i \in M\} \text{ and let } \quad \overline{F} \equiv \{J\subseteq N : J\notin F\}$$

Then, as in Sherali and Adams (1990), we can write

$$\begin{aligned} \text{GUBKP} \equiv \text{conv}(X) = \{\, x : \quad & x_j = \sum_{J \ni j \in J} y_J \;\forall\, j \in N, \\ & \sum_{J\subseteq N} y_J = 1, \\ & y_J \geq 0, \;\; \forall\, J \in F,\, y_J \equiv 0 \;\forall \in \overline{F} \;\} \end{aligned} \tag{1.17}$$

Using the standard projection operation, we obtain

$$\text{GUBKP} \equiv \{\, x : \sum_{j\in N} \pi_j^k x_j \ge \pi_0^k$$

$$\text{where } (\pi_j^k, \pi_0^k),\ k = 1,\ldots,K, \text{ are the extreme directions of } \Pi\}$$

where $\Pi \equiv \{(\pi, \pi_0) : \sum_{j\in J} \pi_j - \pi_0 \ge 0 \quad \forall\ J \in F\}$.

We now consider the characterization of a family of valid inequalities of GUBKP obtainable via a simultaneous lifting of the minimal GUB cover inequality. Recall that the minimal GUB cover inequality $\sum_{j\in K} x_j \ge 1$ is a valid inequality for conv(X($\overline{K}$)), and a facet for conv(X($\overline{K}$)) if $\min_{j\in K}(a_j) + \sum_{l\in M_{\overline{K}}} a_{j(l)} \ge b$. We are interested in finding a lifted inequality, which is a facet of GUBKP, and is of the form $\sum_{j\in K} x_j + \sum_{j\in \overline{K}_-} \pi_j x_j - \sum_{j\in \overline{K}_+} \pi_j \overline{x}_j \ge 1$ where $\overline{x}_j \equiv (1 - x_j)\ \forall\ j \in N$, and $\pi_j\ \forall\ j \in \overline{K}$ are unrestricted in sign. This is of the form

$$\sum_{j\in K} x_j + \sum_{j\in \overline{K}_-} \pi_j x_j + \sum_{j\in \overline{K}_+} \pi_j x_j \ge 1 + \sum_{j\in \overline{K}_+} \pi_j. \qquad (1.18)$$

Consider a polyhedral set $\Pi_{\overline{K}}$, where $\Pi_{\overline{K}} \equiv \{\pi_j\ j \in \overline{K} : (\pi, \pi_0) \in \Pi,\ \pi_j = 1\ \forall\ j \in K,\ \pi_0 = 1 + \sum_{j\in \overline{K}_+} \pi_j\}$. $\Pi_{\overline{K}}$ can be represented as follows, where $\pi_{\overline{K}} \equiv \{\pi_j : j \in \overline{K}\}$

$$\Pi_{\overline{K}} \equiv \{\, \pi_{\overline{K}} : \sum_{j\in J\cap \overline{K}} \pi_j \ge 1 + \sum_{j\in \overline{K}_+} \pi_j - |J \cap K|, \quad \forall\ J \in F\}. \qquad (1.19)$$

Equivalently, we have,

$$\Pi_{\overline{K}} \equiv \{\ \pi_{\overline{K}} : \sum_{j \in (\overline{K}_+ - J)} \pi_j - \sum_{j \in J \cap \overline{K}_-} \pi_j \le |J \cap K| - 1,\ \forall\ J \in F\}.$$

Note that we need to consider only those $J \in F$ above, for which $\overline{KJ} \equiv (J \cap \overline{K}_-) \cup (\overline{K}_+ - J) \neq \phi$. Hence, we have that

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}} : \sum_{j \in \overline{K}_+ - J} \pi_j - \sum_{j \in J \cap \overline{K}_-} \pi_j \le |J \cap K| - 1, \quad \forall\ J \in F \ni \overline{KJ} \neq \phi\}. \tag{1.20}$$

Furthermore, note that we need to examine only the most restrictive constraints in (1.20). Toward this end, we define $\mathrm{GUB}(\overline{K}) = \{T \subseteq \overline{K} : |T \cap N_i| \le 1\ \forall\ i \in M_{\overline{K}}\}$. Now for any $T \in \mathrm{GUB}(\overline{K})$, 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 $\overline{KT} = (T \cap \overline{K}_-) \cup (\overline{K}_+ - T)$. Accordingly, define

$$F_K = \{T \in \mathrm{GUB}(\overline{K})\ :\ J(T) \neq \phi, \text{ and } \overline{KT} \neq \phi\}.$$

Then, we can restate (1.20) as follows.

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}} : \sum_{j \in \overline{K}_+ - T} \pi_j \ - \sum_{j \in T \cap \overline{K}_-} \pi_j \le \ \min\ \{|J \cap K| :\ J \in J(T)\} - 1\ \ \forall\ \ T \in F_K\} \tag{1.21}$$

We now consider an explicit representation of the minimization problem in (1.21). Note that $T \in F_K$ if and only if (i) $|T \cap N_i| \ \le 1$ for $i \in M_{\overline{K}}$, i.e., $T \in \mathrm{GUB}(\overline{K})$, (ii) $\sum_{j \in K_+} a_j + \sum_{j \in T} a_j \ \ge b$, 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 (1.21), denoted by AGUBKP(T), as follows.

$$\text{AGUBKP(T)}: \quad \text{Minimize}\{\sum_{j\in K_+} y_j : \sum_{j\in K_+} a_j y_j \geq b - \sum_{j\in T} a_j, \quad y_j \in (0,1) \ \forall j \in K_+\}$$

Note that $1 \leq \nu(\text{AGUBKP(T)}) \leq |M_K|$, where $\nu(P)$ 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 $\nu(\text{AGUBKP(T)})$ for each $T \in F_K$, using a greedy procedure. Let $\bar{b} = b - \sum_{j\in T} a_j$. If $\bar{b}$ is less than or equal to $\max_{j\in K_+}(a_j)$, then $\nu(\text{AGUBKP(T)}) = 1$. Otherwise, if $\bar{b}$ is less than or equal to the sum of the first two largest a_j's for $j \in K_+$, then $\nu(\text{AGUBKP(T)}) = 2$, and so on. Hence the time complexity of solving AGUBKP is $O(|M|\log|M|)$. Let $N(T) = \nu(\text{AGUBKP(T)}) - 1$. Then, we have

$$\Pi_{\bar{K}} \equiv \{\pi_{\bar{K}} : \sum_{j\in \bar{K}_+ - T} \pi_j - \sum_{j\in T\cap \bar{K}_-} \pi_j \leq N(T) \quad \forall \ T \in F_K\}. \tag{1.22}$$

Proposition 1.16 For a minimal GUB cover K, the inequality (1.18) is a valid inequality for GUBKP if and only if $\pi_{\bar{K}} \in \Pi_{\bar{K}}$, where $\pi_{\bar{K}} = \{\pi_j : j \in \bar{K}\}$, and $\Pi_{\bar{K}}$ is given by (1.22).

Proof. $\pi x \geq \pi_0$ is valid for GUBKP if and only if $\min\{\pi x : x \in X\} \geq \pi_0$. Equivalently, by (1.17), $\min\{\pi x : x \in X\} \geq \pi_0$ holds if and only if

$$\min\{\sum_{j\in N} \pi_j \ (\sum_{J \ni j\in J} y_J) : \sum_{J\subseteq N} y_J = 1, \ y_J \geq 0 \ \forall \ J \in F, y_J = 0 \ \forall \ J \in \bar{F}\} \geq \pi_0.$$

The above condition is equivalent to requiring that

$$\min\{\sum_{J\in F} y_J \ (\sum_{j\in J} \pi_j) : \sum_{J\in F} y_J = 1, \ y_J \geq 0 \ \forall \ J \in F\} \geq \pi_0,$$

which in turn is equivalent to

$$\sum_{j\in J}\pi_j \geq \pi_0\,, \;\; \forall\, J \in F.$$

Consequently, $\pi x \geq \pi_0$ is valid for GUBKP if and only if $(\pi, \pi_0) \in \Pi$. Hence, noting the form of (1.18) and the derivation of (1.22), we have that (1.18) is valid for GUBKP if and only if $\pi_{\bar K} \in \Pi_{\bar K}$, where $\Pi_{\bar K}$ is given by (1.22). This completes the proof. ■

Proposition 1.17 Let K be a minimal GUB cover such that $\min_{j\in K}(a_j) + \sum_{l\in M_{\bar K}} a_{j(l)} \geq b$, and let $\Pi_{\bar K}$ be given by (1.22). Then, the inequality (1.18) having $1 + \sum_{j\in\bar K_+}\pi_j > 0$ is a facet of GUBKP if and only if $(\pi_j,\ j \in \bar K)$ is a vertex of $\Pi_{\bar K}$ with $1 + \sum_{j\in\bar K_+}\pi_j > 0$.

Proof. From Sherali and Adams (1990), we know that $\pi x \geq 1$ is a facet of GUBKP if and only if π is an extreme point of Π_1, where

$$\Pi_1 = \{\pi : \sum_{j\in J}\pi_j \;\geq\; 1, \;\; \forall J \in F\}. \tag{1.23}$$

Hence, (1.18) with $\pi_j = \bar\pi_j$ for $j \in \bar K$ is a facet of GUBKP if and only if the scaled partitioned vector $\hat\pi$, where

$$\hat\pi = \{(\,\hat\pi_j = \frac{1}{(1+\sum_{j\in\bar K_+}\bar\pi_j)}\,,\; j \in K)\,,\;\; (\,\hat\pi_j = \frac{\bar\pi_j}{(1+\sum_{j\in\bar K_+}\bar\pi_j)}\,,\; j \in \bar K\,)\} \tag{1.24}$$

is an extreme point of (1.23).

Now, for each j $\in$K, the set $J(j) \equiv \{j\} \bigcup \overline{K}_+ \in F$ by the hypothesis of the theorem, and the corresponding constraints of (1.23) are linearly independent and are binding at the solution (1.24). The latter |K| linearly independent equality constraints appear as

$$\pi_j = 1 - \sum_{t \in \overline{K}_+} \pi_t \quad \text{for } j \in K \tag{1.25}$$

and so determine the π_j, j $\in$K, uniquely in terms of π_j, j $\in\overline{K}_+$. Now, (1.24) is a vertex of (1.23) if and only if it is feasible to (1.23), and there exist some $|\overline{K}|$ hyperplanes binding from (1.23) which are linearly independent in combination with (1.25). This happens if and only if $\{\hat{\pi}_j, j \in \overline{K}\}$ is an extreme point of the set obtained by imposing (1.25) on (1.23), i.e., the set

$$\{\pi_{\overline{K}} : \sum_{j \in J \bigcap \overline{K}} \pi_j \geq 1 + |J \bigcap K|(\sum_{t \in \overline{K}_+} \pi_t - 1) \ \forall \ J \in F\}. \tag{1.26}$$

This holds if and only if $(\hat{\pi}_j, j \in \overline{K})$ is feasible to (1.26), 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 (1.26) requires from (1.24) that

$$\sum_{j \in J \bigcap \overline{K}} \frac{\overline{\pi}_j}{(1 + \sum_{t \in \overline{K}_+} \overline{\pi}_t)} \geq 1 + |J \bigcap K| \{ \frac{\sum_{t \in \overline{K}_+} \overline{\pi}_t}{(1 + \sum_{t \in \overline{K}_+} \overline{\pi}_t)} - 1\} \ \forall \ J \in F.$$

That is, we must have

$$\sum_{j\in J\cap\overline{K}} \overline{\pi}_j \geq 1 + \sum_{t\in\overline{K}_+} \overline{\pi}_t - |J\cap K| \quad \forall J\in F. \tag{1.27}$$

Note by (1.19) that (1.27) is equivalent to requiring that $\overline{\pi}_{\overline{K}}$ belongs to $\Pi_{\overline{K}}$. Moreover, an inequality in (1.26) is binding at $\hat{\pi}$ if and only if the corresponding inequality in (1.27) is binding. Also, a collection of $|\overline{K}|$ linearly independent equations from (1.26) give $\hat{\pi}$ as the unique solution if and only if the corresponding $|\overline{K}|$ equations from (1.27) give $\overline{\pi}$ as the unique solution, since from (1.24), there is a 1-1 correspondence between $\hat{\pi}$ and $\overline{\pi}$ according to

$$\{\hat{\pi}_j = \frac{\overline{\pi}_j}{(1+\sum\limits_{j\in\overline{K}_+}\overline{\pi}_j)} \quad \forall j\in\overline{K}\} \text{ and } \{\overline{\pi}_j = \frac{\hat{\pi}_j}{(1-\sum\limits_{j\in\overline{K}_+}\hat{\pi}_j)} \quad \forall j\in\overline{K}\}.$$

Hence, $\hat{\pi}_{\overline{K}}$ is an extreme point of (1.26) if and only if $\overline{\pi}_{\overline{K}}$ is an extreme point of $\Pi_{\overline{K}}$ with $1+\sum_{j\in\overline{K}_+}\overline{\pi}_j > 0$, and this completes the proof. ■

Example 1.3 Consider the following example to illustrate the above simultaneous lifting procedure. Let $X \equiv \{x \in (0,1)^9 : x_1+x_2+2x_3+x_4+x_5+2x_6+x_7+x_8+3x_9 \geq 4,\ x_1+x_2+x_3 \leq 1,\ x_4+x_5+x_6 \leq 1,\ x_7+x_8+x_9 \leq 1\}$. 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 $F_K = [\phi,\ \{7\}, \{8\}]$, thereby leading to the following computations.

T	$\overline{KT}$	$\sum_{j \in T} a_j$	ν(AGUBKP(T))	Inequalities of $\Pi_{\overline{K}}$ in (1.22)
ϕ	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, $x_1 + x_2 + x_3 + x_4 + x_5 + x_8 + x_9 \geq 2$ is the only facet obtainable from the minimal GUB cover inequality by the lifting procedure.

Lower and Upper Bounds on the Lifted Coefficients π_j for $j \in \overline{K}$ Under Proposition 1.17

We now derive lower and upper bounds for the coefficients of lifted facets (1.18), under the conditions assumed in Proposition 1.17.

To begin with, let us consider the lifted coefficients π_t, for $t \in \overline{K}_+$. Since GUBKP is a full dimensional polytope, it follows that for any $t \in \overline{K}_+$, we have $T = (\overline{K}_+ - t) \in F_K$, and from (1.22), we directly have that $\pi_t \leq N(\overline{K}_+ - t)$. Hence, from (1.6), an upper bound UB_t on π_t is given by

$$UB_t = N(\overline{K}_+ - t) = \eta^t - 1 \qquad \text{for each } t \in \overline{K}_+ .$$

Now, for given lower bounds LB_s on π_s $\forall s \in (N_p - t)$, where $p \equiv M_t$, let us derive a lower bound LB_t on π_t. Toward this end, examine any $T \in F_K$ such that $t \notin T$. Then, from (1.22), we have that

$$\sum_{j \in \overline{K}_+ - T} \pi_j - \sum_{j \in T \cap \overline{K}_-} \pi_j \leq N(T) \quad \forall\, T \in F_K \ni \; t \notin T. \tag{1.28}$$

From (1.28), since $t \in (\overline{K}_+ - T)$, we have that

$$\pi_t \leq N(T) - [\sum_{j \in \overline{K}_+ - T - t} \pi_j - \sum_{j \in T \cap \overline{K}_-} \pi_j] \;\; \forall T \in F_K \ni t \notin T. \tag{1.29}$$

Note that (1.29) is comprised of all the constraints of $\Pi_{\overline{K}}$ that contain π_t. Since at any vertex of $\Pi_{\overline{K}}$, at least one of (1.29) must be binding, we have that

$$\pi_t = \min \; \{N(T) - [\sum_{j \in \overline{K}_+ - T - t} \pi_j - \sum_{j \in T \cap \overline{K}_-} \pi_j \;] \; : \; T \in F_K \ni t \notin T\}. \tag{1.30}$$

For all $T \epsilon F_k \ni t \notin T$, define $v(T) \equiv \sum_{j \epsilon (\overline{K}_+ - T)} \pi_j - \sum_{j \epsilon T \cap \overline{K}_-} \pi_j$, $\overline{v}(T) \equiv \sum_{j \epsilon (\overline{K}_+ - T - t)} \pi_j - \sum_{j \epsilon T \cap \overline{K}_-} \pi_j$. Then equation (1.30) reads

$$\pi_t = \min\{N(T) - \overline{v}(T) \colon T \epsilon F_K,\; t \notin T\} \tag{1.31}$$

For any $T \epsilon F_K \ni t \notin T$, if $(T + t) \epsilon F_K$, then we have that $\overline{v}(T) = v(T + t) \leq N(T + t)$ by (1.28). On the other hand, if $(T + t) \notin F_K$, then, there exists some $s \epsilon T \cap N_p$ and $\overline{v}(T) = v(T + t - s) - \pi_s \leq N(T + t - s) - \pi_s$. Hence, from (1.31), we have

$$\pi_t = \min\{ \min_{T \epsilon T_1}[N(T) - N(T + t)], \min_{T \epsilon T_2}[N(T) - N(T + t - s) + \pi_s]\}$$

where $T_1 \equiv \{T\in F_K\text{: } t\notin T$ and $(T+t)\in F_K\}$, $T_2 \equiv \{T\in F_K\text{: } t\notin T$ and there exists some $s\in(N_p - t)\bigcap T$ where $p \equiv M_t\}$. Hence, for given lower bounds LB_s on π_s for $s\in(N_p - t)$, a lower bound LB_t on π_t, for $t\in\overline{K}_+$, is given by

$$LB_t = \min\{\min_{T\in T_1}[N(T) - N(T+t)], \min_{T\in T_2}[N(T) - N(T+t-s) + LB_s]\} \qquad (1.32)$$

Next, let us derive lower and upper bounds on the lifted coefficients π_s, for any $s\in\overline{K}_-$. From (1.22), we have that when $T = \{s\}\bigcup\{\overline{K}_+ - t\}$, where $t \equiv j(M_s)$, $\pi_t - \pi_s \leq N(T)$. Hence, we have $\pi_s \geq \pi_t$ - $N(T)$. Consequently, for a given lower bound LB_t on π_t, $t\in\overline{K}_+$, a lower bound LB_s for any $s\in\overline{K}_-$ is given by

$$LB_s = LB_t - N(\overline{K}_+ - t + s) \text{ where } t \equiv j(M_s). \qquad (1.33)$$

Remark 1.2 Note that the lower bounds (1.32) and (1.33) are conditional bounds, each being determined based on lower bounds on the other. These conditional bounds are useful if we restrict the class of facets to have prescribed lower bounds on the π_t or the π_s coefficients. Otherwise, we need to derive unconditional lower bounds LB_j $\forall j\in\overline{K}$. Toward this end, we derive an unconditional lower bound of zero on π_t, $\forall t\in\overline{K}_+$, as follows. From Johnson and Padberg (Proposition 2.1, 1981), since the inequality (1.18) is a facet of GUBKP and since $|M_K| > 1$ (otherwise, dim(GUBKP) $<$ n), it can be readily shown that the following inequality

$$\sum_{j\in K_+} z_j + \sum_{j\in\overline{K}_+}\pi_j z_j + \sum_{i\in M_{\overline{K}}}\ \sum_{j\in\{N_i - j(i)\}}(\pi_{j(i)} - \pi_j)z_j \leq |M_K| - 1 \qquad (1.34)$$

is a facet of the convex hull of the polytope.

$$Z \equiv \operatorname{conv}\{z \in (0,1)^n : \sum_{i \in M}\sum_{j \in N_i} \bar{a}_j z_j \leq \bar{b}, \ \sum_{j \in N_i} z_j \leq 1 \ \ \forall i \in M\}$$

where for all $i \in M$, $\bar{a}_{j(i)} = a_{j(i)}$, $\bar{a}_j = a_{j(i)} - a_j \ \forall \ j \in \{N_i - j(i)\}$, and $\bar{b} = b + \sum_{j \in N_+} a_j$. Since the constraints of the polytope Z have all nonnegative coefficients, we have that all the coefficients of the facet (1.34) are nonnegative (Hammer et al., 1975). Accordingly, for each $t \in \overline{K}_+$, we have

$$\pi_t \geq 0, \quad \text{and } \pi_t - \pi_s \geq 0, \ \ \forall s \in (N_p - t), \text{ where } p = M_t \qquad (1.35)$$

From (1.35), we have a valid lower bound of zero on π_t, $\forall t \in \overline{K}_+$. Consequently, from (1.33), we have that $LB_s = - N(\overline{K}_+ - t + s) \ \forall s \in (N_p - t)$, where $p \equiv M_t$. ■

Finally, let us derive an upper bound UB_s for any given $s \in \overline{K}_-$. Note from (1.22) that the collection of constraints defining $\Pi_{\overline{K}}$ that contain the coefficient π_s are given by

$$\pi_s \geq \sum_{j \in (\overline{K}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \overline{K}_-} \pi_j - N(T) \quad \forall \ T \in F_K \ni s \in T. \qquad (1.36)$$

Again, at any vertex of $\Pi_{\overline{K}}$, since at least one of (1.36) must be binding, we have that

$$\pi_s = \max\{\sum_{j \in (\overline{K}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \overline{K}_-} \pi_j - N(T) : \ \ T \in F_K \ni s \in T\}. \qquad (1.37)$$

Now, in (1.37), if $(T - s) \in F_K$, we have from (1.22) that

$$\sum_{j \in (\overline{K}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \overline{K}_-} \pi_j \leq N(T - s).$$

On the other hand, if $(T-s)\notin F_K$, then $N(T-s)\equiv\infty$, and so the foregoing inequality holds for all $T\in F_K \ni s\in T$. Furthermore, for any $T\in F_K \ni s\in T$, we also have $(T-s+t)\in F_K$, where $t\equiv j(M_s)$. Consequently, from (1.22), the corresponding constraint for $T'\equiv(T-s+t)\in F_K$ yields

$$[\sum_{j\in(\overline{K}_+-T)}\pi_j-\sum_{j\in(T-s)\bigcap\overline{K}_-}\pi_j]=[\sum_{j\in(\overline{K}_+-T')}\pi_j-\sum_{j\in T'\bigcap\overline{K}_-}\pi_j]+\pi_t\leq N(T')+\pi_t\leq N(T')+UB_t.$$

Note that if $\overline{KT'}\equiv(\overline{K}_+-T')\bigcup(T'\bigcap\overline{K}_-)=\phi$, we simply have $N(T')\equiv 0$ in that case. Combining the last two inequalities, we may write for any $T\in F_K\ni s\in T$, and $t\equiv j(M_s)$,

$$[\sum_{j\in(\overline{K}_+-T)}\pi_j-\sum_{j\in(T-s)\bigcap\overline{K}_-}\pi_j]\leq\ \min\{N(T-s),\ N(T-s+t)+UB_t\}.$$

Substituting this in (1.37) above, we obtain the following upper bound UB_s on π_s for any $s\in\overline{K}_-$.

$$UB_s=\max_{T\in F_K\ni s\in T}\ \{\min\{N(T-s),\ N(T-s+t)+UB_t\}-N(T)\},\qquad(1.38)$$

where $t\equiv j(M_s)$.

Example 1.4 Consider the following constraints of a GUB constrained knapsack problem, where $X\equiv\{x\in(0,1)^{12}$: $2x_1+5x_2+2x_3+3x_4+x_5+3x_6+x_7+3x_8+2x_9+2x_{10}+2x_{11}+2x_{12}\geq 16$, $x_1+x_2\leq 1$, $x_3+x_4\leq 1$, $x_5+x_6\leq 1$, $x_7+x_8\leq 1$, $x_9\leq 1$, $x_{10}\leq 1$, $x_{11}\leq 1$, $x_{12}\leq 1\}$. For a minimal GUB cover $K=\{9,\ 10,\ 11,\ 12\}$, the minimal GUB cover inequality is $x_9+x_{10}+x_{11}+x_{12}\geq 1$, which is a facet of $\text{conv}(X(\overline{K}))$ by Proposition 1.6. Note that $\overline{K}_+ = \{2, 4, 6, 8\}$ and $\overline{K}_- = \{1, 3, 5, 7\}$. We consider a facet of the form (1.18) with $\pi_j \geq 0 \ \forall j \in \overline{K}_-$.

We now derive lower and upper bounds on the coefficient π_2, where $\{2\} \in \overline{K}_+$. Since $\eta^2 = 4$, we have that $UB_2 = \eta^2 - 1 = 3$. Furthermore, the set $T_1 \equiv \{T: T \in F_K \ni \{2\} \notin T$ and $(T + \{2\}) \in F_K\}$, is given by $T_1 = \{(3, 6, 8), (4, 6, 8)\}$. Also the set $T_2 \equiv \{T: T \in F_K \ni \{2\} \notin T$ and $\{1\} \in T\}$ is given by $T_2 = \{ (1, 4, 6), (1, 4, 8), (1, 6, 8), (1, 3, 5, 8), (1, 3, 6, 7), (1, 3, 6, 8), (1, 4, 5, 8), (2, 4, 6, 7), (1, 4, 6, 8)\}$. Hence, by (1.32), conditioned on $LB_1 = 0$, a lower bound LB_2 can be computed as follows.

$$LB_2 = \min\{ \min_{T \in T_1}[N(T) - N(T + \{2\})], \ \min_{T \in T_2}[N(T) - N(T + \{2\} - \{1\}) + 0]\} = 1$$

Next, let us select $\{1\} \in \overline{K}_-$, and illustrate the computation of an upper bound on the coefficient π_1 in any lifted facet (1.18). Now, the set $T' \equiv \{T : T \in F_K \ni \{1\} \in T\}$ is given by $\{(1, 4, 6), (1, 4, 8), (1, 6, 8), (1, 3, 5, 8), (1, 3, 6, 7), (1, 3, 6, 8), (1, 4, 5, 8), (1, 4, 6, 7), (1, 4, 6, 8)\}$. Since $UB_2 = 3$, we have that by (1.38),

$$UB_1 = \max_{T \in T'}\{ \min\{N(T - \{1\}), \ N(T - \{1\} + \{2\}) + 3\} - N(T)\} = 2.$$

Hence, we have that $0 \leq \pi_1 \leq 2$ and $1 \leq \pi_2 \leq 3$ in any lifted facet (1.18) having $\pi_1 \geq 0$. Note that we can also compute an unconditional lower bound LB_1, taking $LB_2 = 0$. By (1.33), $LB_1 = 0 - N(\overline{K}_+ - \{2\} + \{1\}) = -2$. Hence, a set of unconditional bounds on π_1 and π_2 are given by $-2 \leq \pi_1 \leq 2$ and $0 \leq \pi_2 \leq 3$.

1.5. A Special Case: the Zero-One Knapsack Polytope

Consider a special case of GUBKP with $|N_i| = 1 \ \forall i \in M$, which represents the ordinary knapsack polytope, denoted by KP. That is, $KP \equiv \text{conv}\{x \in (0,1)^n : \sum_{j \in N} a_j x_j \geq b\}$, where the data are all integer, $N=\{1,\ldots,n\}$, $b>0$, $0<a_j\leq b$ $\forall j\in N$, and $\sum_{j\neq k} a_j \geq b$ for all $k\in N$. Recall that the minimal (GUB) cover inequality $\sum_{j\in K} x_j \geq 1$ is a facet of $\mathrm{conv}(KP(\overline{K}))$, where $KP(\overline{K}) = KP \bigcap \{x\in(0,1)^n : x_j = 1, \ \forall\ j\in\overline{K}\}$. Our interest is in characterizing a (simultaneously) lifted facet, as in Balas and Zemel (1978), of the form

$$\sum_{j\in K} x_j + \sum_{j\in\overline{K}} \pi_j x_j \geq 1 + \sum_{j\in\overline{K}} \pi_j, \tag{1.39}$$

where K is a minimal (GUB) cover of KP. Toward this end, in the spirit of (1.21) and Problem AGUBKP(T), we define

$$f(\theta) = \min\{\sum_{j\in K} y_j : \sum_{j\in K} a_j y_j \geq \overline{b} + \theta\} - 1,$$

where $\overline{b} = b - \sum_{j\in\overline{K}} a_j$.

By Proposition 1.17, we have that (1.39) is a facet of KP if and only if $(\pi_j,\ j\in\overline{K})$ is an extreme point of $\Pi_{\overline{K}}$ with $1 + \sum_{j\in\overline{K}} \pi_j > 0$, where

$$\Pi_{\overline{K}} \equiv \{\pi_{\overline{K}} : \sum_{j\in(\overline{K}-T)} \pi_j \leq f(\sum_{j\in(\overline{K}-T)} a_j)\ \forall T\in F_K\} \tag{1.40}$$

and where $F_K = \{T\subset\overline{K} : \sum_{j\in K\bigcup T} a_j \geq b\}$.

This is precisely Balas and Zemel's (1978) characterization of simultaneously lifted facets obtainable from minimal cover inequalities. We now derive upper and lower bounds on π_j, $j\in\overline{K}$, for such facets of KP.

Upper bound UB_t on π_t, $t\in\overline{K}$

From (1.40), by examining $T = \overline{K} - \{t\}\in F_K$, we directly have that $\pi_t \le f(a_t)$. Hence, an upper bound UB_t on π_t is given by

$$UB_t = f(a_t) \quad \text{for each } t\in\overline{K}.$$

Note that UB_t is the same as Balas and Zemel's upper bound on π_t, $t\in\overline{K}$.

Lower bound LB_t on π_t, $t\in\overline{K}$

Balas and Zemel (1978) derive the following lower bound, denoted by $LBBZ_t$:

$$LBBZ_t = h \ \ \forall t\in S_h,$$

where letting $E(K)$ denote the extension of K as before, we have $S_0 = \overline{E(K)}$, and $S_h = \{t\in(E(K) - K) \ : \sum_{j\in K_h} a_j \le a_t < \sum_{j\in K_{h+1}} a_j\}$, where K_h is the index set of the h largest a_j for $j\in K$. Note that $UB_t = LBBZ_t$ or $LBBZ_t + 1$.

We now construct a tighter lower bound on π_t. Consider any $t\in\overline{K}$, and let us examine any $T\in F_K$ such that $t\notin T$. From (1.40), we have that

$$\pi_t \le f(\sum_{j\in(\overline{K}-T)} a_j) - \sum_{j\in(\overline{K}-T-\{t\})} \pi_j \ \ \forall T\in F_K,\ \{t\}\notin T. \tag{1.41}$$

But since at an extreme point of $\Pi_{\overline{K}}$, at least one of (1.41) must be binding, we have that

$$\pi_t = \min\{f(\sum_{j\in(\overline{K}-T)} a_j) - \sum_{j\in(\overline{K}-T-\{t\})} \pi_j \ : \ T\in F_K \ni \{t\}\notin T\},$$

given other π_j values. Consequently, we get

$$LB_t = \min\{\, f(\sum_{j\in(\overline{K}-T)} a_j) - f(\sum_{j\in(\overline{K}-T-\{t\})} a_j) \ : \ T\in F_K \ni \{t\}\notin T\}. \qquad (1.42)$$

Proposition 1.18 $LB_t \geq LBBZ_t$, $\forall t\in\overline{K}$.

Proof. By the monotone increasing nature of the function f, it follows that $LB_t \geq 0$ $\forall t\in\overline{K}$. Since $LBBZ_t = 0$ for $t\in\overline{E(K)}$, the result holds trivially for this case. Hence, suppose that $t\in(E(K)-K)$. Consider any $T\in F_K$ with $t\notin T$, and examine two cases.

Case (i) : $a_t = \sum_{j\in K_h} a_j$. It follows that $LBBZ_t = h \equiv f(a_t - \overline{b}) + 1$. But we have, defining $\hat{b} = b - \sum_{j\in T} a_j - a_t > 0$, that

$$\begin{aligned} f(\sum_{j\in(\overline{K}-T)} a_j) - f(\sum_{j\in(\overline{K}-T-\{t\})} a_j) &= \min\{\sum_{j\in K} y_j : \sum_{j\in K} a_j y_j \geq \hat{b} + a_t\} - \min\{\sum_{j\in K} y_j : \sum_{j\in K} a_j y_j \geq \hat{b}\} \\ &\geq \min\{\sum_{j\in K} y_j : \sum_{j\in K} a_j y_j \ \geq \ a_t\} \ = \ f(a_t - \overline{b}) + 1. \end{aligned} \qquad (1.43)$$

This implies from (1.42) that $LB_t \geq LBBZ_t$.

Case (ii) : $\sum_{j\in K_h} a_j < a_t < \sum_{j\in K_{h+1}} a_j$. In this case, we have, $LBBZ_t = h \equiv f(a_t - \bar{b})$. Let Δ be the amount that needs to be subtracted from a_t so that $a_t - \Delta = \sum_{j\in K_h} a_j$. Then, using the monotone increasing nature of the function f and following (1.43), we have that

$$\begin{aligned} f(\sum_{j\in(\bar{K}-T)} a_j) - f(\sum_{j\in(\bar{K}-T-\{t\})} a_j) &\geq f(\sum_{j\in(\bar{K}-T)} a_j - \Delta) - f(\sum_{j\in(\bar{K}-T-\{t\})} a_j) \\ &\geq f(a_t - \Delta - \bar{b}) + 1 = f(a_t - \bar{b}). \end{aligned} \qquad (1.44)$$

Hence from (1.42) and (1.44), we have that $LB_t \geq LBBZ_t$. Therefore, the result holds for any $t\in(E(K) - K)$ as well, and this completes the proof. ■

Example 1.5 Consider $KP = \text{conv}\{x\in(0, 1)^4 : 3x_1 + 3x_2 + 3x_3 + 2x_4 \geq 7\}$. Let $K = \{1, 2\}$, so that $E(K) = \{1, 2, 3\}$ and $\bar{b} = 2$. Consider $t = \{4\}\in\overline{E(K)}$. Note that $\{T\in F_K : t\notin T\} = \{3\}$. Hence, from (1.42), we have that $LB_4 = f(a_4) - f(0) = f(a_4) = UB_4 = 1 > LBBZ_4 = 0$. Note, however, that K is not a strong cover, as evidenced by the (strong) minimal cover $K' = \{3, 4\}$. Otherwise, by the definition of a strong cover, we would have had for any $t\in\overline{E(K)}$, if it exists, that $UB_t = f(a_t) = 0$, and so $LB_t = 0 \equiv LBBZ_t$ as well.

Example 1.6 Consider $KP = \text{conv}\{x\in(0, 1)^4 : 3x_1 + 3x_2 + 3x_3 + 4x_4 \geq 7\}$. The minimal cover $K = \{1, 2, 3\}$ is a strong cover for KP. Since $\bar{b} = 3$, $\bar{K} = \{4\}$, and for $t = \{4\}\in(E(K) - K)$, we have that $\{T\in F_K : t\notin T\} = \phi$, this gives $LB_4 = f(a_4) - f(0) = f(a_4) \equiv UB_4 = 2$. However, $LBBZ_t = 1 < LB_4$.

1.6. A Generalization of the GUB Cover

Motivated by the strengthening procedure of Proposition 1.5, we can suggest an alternative, more general, definition of the GUB cover set in order to obtain a family of a valid inequalities for GUBKP.

A set $C \subseteq N$ is called a *cover set* of X if $\sum_{j \in \bar{C}_+} a_j \leq b - 1$, where $\bar{C} \equiv N - C$. A cover set C is called a *minimal cover set* of X if no proper subset of C is a cover set, i.e., for $k \in C$, $\sum_{j \in D_+} a_j \geq b$, where $D \equiv \bar{C} + \{k\}$. Accordingly, we define a *minimal cover inequality* as

$$\sum_{j \in C} x_j \geq 1 \qquad (1.45)$$

By the definition of the cover set, inequality (1.45) is valid for GUBKP. For a minimal cover set C, we define $R \equiv \{j \in \bar{C} : a_j \geq \max_{j \in C}\{a_j\}\}$. An *extension of a minimal cover set C* , denoted by E(C), is defined as $E(C) = C \bigcup S$, where $S = \bigcup_{i \in M_R} N_i$.

Remark 1.3 Observe that we can obtain a minimal cover C from a given minimal GUB cover K by the strengthening procedure of Proposition 1.5 . In addition, the definition of a cover set here is precisely equivalent to the definition of "GUB cover" in Wolsey (1990). However, Wolsey (1990) was interested in characterizing valid inequalities for GUB constrained problems, without in particular, studying the generation of facets for GUBKP.

Proposition 1.19 If C is a minimal cover set of X, then

$$\sum_{j \in E(C)} x_j \geq 1 + |M_R| \tag{1.46}$$

is a valid inequality for GUBKP, which dominates the minimal cover inequality whenever $R \neq \phi$.

Proof. Suppose that there exists a solution $\bar{x} \in X$ such that $\sum_{j \in E(C)} \bar{x}_j \leq |M_R|$. By the definition of E(C), the largest $|M_R|$ coefficients a_j among $\{a_j : j \in E(C)_+\}$ are the coefficients $\{a_j : j \in R_+\}$. But $\sum_{j \in \bar{C}_+} a_j < b$ implies that even with such a solution $\bar{x} \equiv \{\bar{x}_j = 1$ $\forall j \in R_+, \bar{x}_j = 1 \ \forall j \in (\bar{C} - S)_+\}$, we cannot obtain a feasible solution to X by the definition of a cover set C. Hence, the inequality (1.46) must always hold. Moreover, note that the inequality (1.46) can be represented as $\sum_{j \in C} x_j \geq 1 + \sum_{i \in M_R} (1 - \sum_{j \in (N_i - C)} x_j)$ Since $\sum_{j \in (N_i - C)} x_j \leq 1$ for all $i \in M_R$, the inequality (1.46) dominates the cover inequality (1.45), and this completes the proof. ■

We now consider the association between canonical facets and the minimal cover inequalities of GUBKP. A similar result for the ordinary knapsack problem appears in Balas (1975).

Proposition 1.20 If $\sum_{j \in H} x_j \geq 1$ is a facet of GUBKP, then H is a minimal cover of X.

Proof. Since $\sum_{j \in H} x_j \geq 1$ is valid for GUBKP, we have that $\sum_{j \in \bar{H}_+} a_j \leq b - 1$. This implies that H is a cover set. If H is not a minimal cover, then there exists some k∈H such that (H-k) is a cover, or that $\sum_{j \in (H-k)} x_j \geq 1$ is valid. Since this strictly dominates the given inequality, we have a contradiction, and this completes the proof. ■

Example 1.7 Consider the following example, where $X \equiv \{x \in (0,1)^8 : x_1 + 5x_2 + x_3 + 5x_4 + x_5 + 2x_6 + x_7 + 2x_8 \geq 9,\ x_1 + x_2 \leq 1,\ x_3 + x_4 \leq 1,\ x_5 + x_6 \leq 1,\ x_7 + x_8 \leq 1\}$. Since $\sum_{j \in (N - N_i)+} a_j \geq b$ for all $i \in M$, the convex hull of X is a full dimensional polytope. A cover C is given by $\{1, 2, 3, 4, 5, 6\}$, where $C_+ = \{2, 4, 6\}$, $C_- = \{1, 3, 5\}$, $\overline{C}_+ = \{8\}$, and $\overline{C}_- = \{7\}$. The corresponding cover inequality (1.45) is $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1$. A minimal cover C is given by $\{2, 4\}$, and the corresponding minimal cover inequality is $x_2 + x_4 \geq 1$. Note that this cover has no extension. However, among minimal covers of X, a minimal cover $C = \{4, 5, 6\}$ has an extension $E(C) = \{1, 2, 4, 5, 6\}$, and the corresponding extended inequality (1.46) is $x_1 + x_2 + x_4 + x_5 + x_6 \geq 2$.

1.7. Summary and Recommendations for Further Research

In this chapter, we have presented a new class of valid inequalities for problem GKP obtained by an extension of minimal cover inequalities defined for the ordinary knapsack polytope. We have also developed a necessary and sufficient condition for such an inequality to define a facet of a lower dimensional polytope. Subsequently, we have developed a sequential lifting procedure in order to obtain a family of facets. The sequential lifting procedure developed herein computes lifted coefficients of the variables in each GUB set simultaneously, in contrast with the usual sequential lifting procedure that lifts only one variable at a time. Moreover, we have shown that this sequential lifting procedure can be implemented in polynomial time of complexity O(mn), where n is the number of variables and m ($\leq$ n) is the number of GUB constraints. Following this, we have used the RLT procedure to easily characterize facets obtainable through a simultaneous lifting procedure. This characterization enables us to derive lower and upper bounds on the lifted coefficients. Particularly, for the special case of the ordinary knapsack polytope, we have used this analysis to further tighten a known lower bound on the coefficients of lifted facets derived from minimal covers. Finally, we have developed a generalization of the GUB cover inequality and discussed its implementation.

Although we have shown that our proposed lower bound dominates the lower bound of Balas and Zemel (1979) for the case of the ordinary knapsack polytope, its rudimentary derivation requires an enumeration of all the feasible subsets, a task that involves too much effort in the worst case. Consequently, a further extension of this research might be the development of an efficient procedure for computing such bounds, which continue to dominate the bounds of Balas and Zemel for the case of the ordinary knapsack polytope.

2. The Polyhedral Structure of the Set Partitioning Polytope

2.1. Introduction

The set partitioning problem can be stated as follows.

$$\textbf{SP}: \quad \text{Minimize } \{cx : Ax = e, \ x_j = 0 \text{ or } 1 \ \forall j \in N\}$$

where $A = (a_{ij})$ is an mxn matrix of 0's and 1's, e is an m vector of 1's, $N = \{1,...,n\}$, and $M = \{1,...,m\}$. We will denote by a_j the jth column of A, and assume that A has no zero rows or columns. We assume that rank(A) = m < n and that SP is feasible.

Problem SP has been extensively investigated by several researchers for the last 30 years because of its simple combinatorial structure and its numerous practical applications. Among the applications described in the literature are crew scheduling, truck scheduling, information retrieval, circuit design, capacity balancing, capital investment, facility location, political districting, and radio communication planning. (See Balas and Padberg, 1976 for a survey.) Other applications of problem SP are given by Garfinkel and Nemhauser (1972), Salkin (1975), Fisher and Kedia (1990), and Chan and Yano (1992). Two well-known approaches to problem SP are implicit enumeration and simplex based cutting plane methods. (See Balas and Padberg, 1976 for a survey of these algorithms.) Particularly, as observed by Chan and Yano (1992), SETPAR (Marsten et al., 1979), a linear programming based branch-and-bound code, is still the most popular tool of solving problem SP among practitioners. However, in this research effort, we are mainly concerned with the polyhedral structure of the convex hull of feasible solutions to problem SP, denoted by SPP, and the development of new classes of cutting planes.

Denoting the convex hull by "conv," define

$$\text{SPP} = \text{conv}\{x \in R^n: Ax = e, \ x_j = 0 \text{ or } 1 \ \ \forall j \in N\}$$

as the *set partitioning polytope*, and denote the continuous (linear programming) relaxation of SPP by

$$\overline{\text{SPP}} = \{x \in R^n: Ax = e, \ x_j \geq 0 \ \ \forall j \in N\}.$$

The following is an outline of this chapter. Recently, Sherali and Adams (1990) have proposed a new reformulation-linearization-technique (RLT), for generating a hierarchy of relaxations for linear and polynomial zero-one programming problems, spanning the spectrum from the continuous relaxation to the convex hull representation. By specializing the application of this technique to the set partitioning polytope, we are able to derive polyhedral representations for this problem. Similar to the pure zero-one programming case, we multiply the problem constraints with d-degree polynomial factors composed of the n binary variables and their complements, for some fixed $d \in \{0,...,n\}$, where the zero-degree factors are taken as unity. We then linearize the resulting polynomial program through a suitable redefinition of variables. By exploiting the set partitioning structure, namely, zero-one coefficients of the constraint matrix A with unit right hand sides, in Section 2, we can obtain an explicit representation of the polyhedral structure of various levels of the above hierarchy for the SPP. Using this result, we show in Section 3 that all of Balas' valid inequalities for the set partitioning polytope (1977) can be readily obtained. More importantly, in Section 4, we derive a family of cutting planes, which delete any obtained fractional basic feasible solution to the underlying linear programming relaxation $\overline{SPP}$. Finally, Section 5 concludes this chapter.

2.2. A Hierarchy of Relaxations for the Set Partitioning Polytope

We define some notation.

$M_k = \{i \in M : a_{ik} = 1\}$, $\overline{M_k} = M - M_k$, $k \in N$.

$N_i = \{k \in N : a_{ik} = 1\}$, $\overline{N_i} = N - N_i$, $i \in M$.

Using this notation, problem SP can be restated as follows.

SP: Minimize $\sum_{j \in N} c_j x_j$

subject to

$$\sum_{j \in N_i} x_j = 1 \quad \forall\, i \in M \tag{2.1}$$

$$x_j \text{ binary } \forall\ j \in N \tag{2.2}$$

Now, for any $d \in \{0,...,n\}$, define the (nonnegative) polynomial *factors of degree d* as

$$F_d(J_1, J_2) = \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j) \text{ for each } J_1, J_2 \subseteq N \text{ such that } J_1 \cap J_2 = \phi,\ |J_1 \cup J_2| = d. \tag{2.3}$$

Any (J_1, J_2) satisfying the condition in (2.3) is said to be of *order d* . For example, for $n=2$ and $d=2$, these factors are x_1x_2, $x_1(1 - x_2)$, $(1 - x_1)x_2$, $(1 - x_1)(1 - x_2)$. For convenience, we will take $F_0(\phi, \phi) = 1$, and accordingly, assume products over null sets to be unity. For a given $d \in \{0,\ldots,n\}$, we define $F_d(J) \equiv F_d(J, \phi)$.

With these factors, let us construct a relaxation SPP_δ of SPP, for any given $\delta \in \{0,\ldots,n\}$, using the following two steps.

STEP 1. For all $d \in \{0,\ldots, \delta\}$, multiply each of the equalities (2.1), by each of the factors $F_d(J)$ of degree d. Include the constraints representing nonnegativity of all possible factors $F_d(J_1, J_2)$ of degree d, $d = 1, \ldots, \min(\delta + 1, n)$. 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 constraints.

$$[\, |N_i \cap J| F_d(J) + \sum_{j \in (N_i - J)} F_{d+1}(J + j) \;=\; F_d(J) \;\; \forall i \in M \,], \;\; \forall J \subseteq N \ni |J| = d, \;\; d = 0,\ldots, \delta \quad (2.4)$$

$$F_d(J_1, J_2) \geq 0, \;\; (J_1, J_2) \text{ of order d}, \;\; d = 1,\ldots, \min(\delta + 1, n) \quad (2.5)$$

STEP 2. Linearize the constraints in (2.4) and (2.5) by substituting the following variables for the corresponding nonlinear terms for each $J \subseteq N$, $w_J \equiv \prod_{j \in J} x_j$, where we assume the notation that $w_j \equiv x_j$ for $j = 1,\ldots,n$, and $w_\phi \equiv 1$. Furthermore, denoting by $f_d(J)$ and $f_d(J_1, J_2)$ the respective linearized forms of the polynomial expressions $F_d(J)$ and $F_d(J_1, J_2)$ under such a substitution, we obtain the following polyhedral set SPP_δ, where

$SPP_\delta = \{ (x,w) :$

$$[\,|N_i \cap J| f_d(J) + \sum_{j \in (N_i - J)} f_{d+1}(J + j) = f_d(J) \;\; \forall i \in M\,], \;\; \forall J \subseteq N \ni |J| = d, \; d = 0,\ldots,\delta \qquad (2.6)$$

$$f_d(J_1, J_2) \geq 0, \quad (J_1, J_2) \text{ of order } d, \quad d = 1,\ldots, \min(\delta + 1, n) \qquad \}. \qquad (2.7)$$

Remark 2.1 Note that for the case $\delta = 0$, using the fact that $f_0(\phi, \phi) \equiv 1$, and that $f_1(j, \phi) \equiv x_j$ and $f_1(\phi, j) \equiv (1 - x_j)$ for $j = 1,\ldots,n$, it follows that SPP_0 given by (2.6) and (2.7) is $\overline{SPP}$. Also note that for any $d \in \{1,.., \delta\}$,

$$f_d(J_1, J_2) \equiv f_{d+1}(J_1 + k, J_2) + f_{d+1}(J_1, J_2 + k)$$

where $k \in N - (J_1 \bigcup J_2)$. Hence, for any $d \in \{1,\ldots,n-1\}$, the (d+1)th-order nonnegativity constraints of (2.7) imply the nonnegativity of the (d)th-order constraints.

Remark 2.2 For a given $\delta \in \{0,\ldots,n\}$, denote by

$$SPP_{P\delta} = \{x : (x,w) \in SPP_\delta\}$$

the projection of the set SPP_δ onto the space of the original variables x. According to Sherali and Adams (1990), we can easily show that for $\delta = 0,\ldots,n$, the sets $SPP_{P\delta}$ represent a sequence of nested, valid relaxations leading up to the convex hull representation. That is,

$$SPP \equiv SPP_{Pn} \subseteq SPP_{P(n-1)} \subseteq \ldots \subseteq SPP_{P1} \subseteq SPP_0 \equiv \overline{SPP}.$$

Due to the structure of the set partitioning polytope, we can further reduce the size of the constraints (2.7) of SPP_δ, $\delta \in \{0,\ldots,n\}$. Toward this end, consider the following set of constraints associated with SPP_δ, $\delta \in \{0,\ldots,n\}$.

$$f_d(J) \geq 0, \quad \forall J \subseteq N \ni |J| = d, \quad d = 1,\ldots, \min(\delta + 1, n) \tag{2.8}$$

Proposition 2.1 For any $\delta \in N$, the constraints (2.6) and (2.8) imply the constraints (2.7) in SPP_δ.

Proof. Consider problem SPP_δ for $\delta \in N$. We will use induction on $|J_2|$ to prove the theorem. Accordingly, consider any $d \in \{1,\ldots, \min(\delta + 1, n)\}$.

(a) If $|J_2| = 0$, then $f_d(J_1, J_2) \equiv f_d(J_1)$ is implied by (2.8). Next, suppose that $|J_2| = 1$, say, $J_2 = \{k\} \in N$. Then

$$f_d(J_1, J_2) \equiv \left[\prod_{j \in J_1} x_j \, (1 - x_k) \right]_L \equiv f_{d-1}(J_1) - f_d(J_1 + \{k\}) \tag{2.9}$$

where $[\, F(x) \,]_L$ denotes the linearized term of the polynomial $F(x)$ as defined at Step 2 of the RLT procedure. For some $i \in M_k$, we have a constraint in (2.6) as follows. $[F_{d-1}(J_1) \;\; (x_k + \sum_{j \in N_i - \{k\}} x_j) - F_{d-1}(J_1)]_L = 0$ or,

$$f_d(J_1 + \{k\}) + \left[\sum_{j \in (N_i - \{k\})} x_j \, F_{d-1}(J_1) \right]_L = f_{d-1}(J_1). \tag{2.10}$$

From (2.8), (2.9) and (2.10), it follows that

$$\begin{aligned} f_d(J_1, J_2) &\equiv [\sum_{j\in(N_i-\{k\})} x_j F_{d-1}(J_1)]_L \\ &\equiv \sum_{j\in(N_i-\{k\}-J_1)} f_d(J_1+\{j\}) + \sum_{j\in(N_i-\{k\})\bigcap J_1} f_{d-1}(J_1) \geq 0. \end{aligned}$$

Hence, it follows that $f_d(J_1, J_2) \geq 0$.

(b) Assume that $f_d(J_1, J_2) \geq 0$ is implied by the constraints (2.6) and (2.8) for $|J_2| = 1,\ldots,(p-1)$. Consider the case of $|J_2| = p$, where $p \geq 2$, and suppose that $k\in J_2$. Hence, we can write

$$f_d(J_1, J_2) \equiv [\prod_{j\in J_1} x_j \prod_{j\in(J_2-\{k\})} (1-x_j)\ (1-x_k)\ ]_L.$$

For some $i\in M_k$, we have the set partitioning constraint, $x_k + \sum_{j\in(N_i-\{k\})} x_j = 1$. Since (2.6) includes constraints obtained by multiplying the foregoing constraint with all factors $F_d(J)$, $|J| = 0,1,\ldots,\delta$, and since $\prod_{j\in J_1} x_j \prod_{j\in(J_2-\{k\})} (1-x_j)$ is a linear combination of such factors, it follows that (2.6) gives

$$[(\sum_{j\in(N_i-\{k\})} x_j) \prod_{j\in J_1} x_j \prod_{j\in(J_2-\{k\})} (1-x_j)]_L \equiv [(1-x_k) \prod_{j\in J_1} x_j \prod_{j\in(J_2-\{k\})} (1-x_j)]_L \equiv f_d(J_1, J_2).$$

Letting $J'_2 = (J_2 - \{k\})$, the left-hand-side of the above equation is comprised of terms of the type $f_{\min(d+1,n)}(J_1 \bigcup \{j\}, J'_2)$ for $j\in N_i - \{k\} \ni j\notin J_1 \bigcup J'_2$, and of the type $f_d(J_1, J'_2)$ for $j\in N_i - \{k\} \ni j\in J_1$, and zeroes in case $j\in(N_i - \{k\}) \bigcap J'_2$. 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. ■

Remark 2.3 By Remark 2.1, note that we only need $f_D(J_1, J_2) \geq 0 \ \forall (J_1, J_2)$ of order $D = \min(\delta + 1, n)$ in (2.7) to obtain the same representation of SPP_δ. Hence, the constraints (2.6) and (2.7) can be further reduced by deleting the remaining (redundant) constraints. Then, as evident in the proof of Proposition 2.1, we only need $f_D \geq 0$ and $f_{D-1}(J) \geq 0$, where $D = \min(\delta + 1, n)$, in (2.8) for obtaining the same representation of SPP_δ. However, for the simplicity of analysis, we include the redundant nonnegativity constraints, $f_d(J) \geq 0$, for $d \in (1,\ldots, D - 2)$, as well in the formulation, since these constraints will be treated implicitly.

By Proposition 2.1, the relaxation SPP_δ, $\delta \in \{0,\ldots,n\}$, can be restated as follows.

$SPP_\delta = \{ (x,w) :$

$$[\, |N_i \cap J| f_d(J) + \sum_{j \in (N_i - J)} f_{d+1}(J + j) = f_d(J) \ \forall i \in M \,], \quad \forall J \subseteq N \ni |J| = d, \ d = 0,\ldots, \delta \qquad (2.11)$$

$$f_d(J) \geq 0 \ \forall J \subseteq N \ni |J| = d, \quad d = 1,\ldots, \min(\delta + 1, n) \qquad \}. \qquad (2.12)$$

The special structure of SPP_δ stated in this form lends itself to a further reduction in size by deleting certain null variables and the resulting trivial constraints. Consider the following Proposition to explicitly identify such null variables.

Proposition 2.2 Consider any $J \subseteq N$, $|J| = d \in \{1,\ldots, \delta\}$, and define $M_J \equiv \bigcup_{k \in J} M_k$. Then $f_{d+1}(J + j) = 0$ in (2.11) $\forall j \in N_i - J \ \forall i \in M_J$. Moreover, if $|J \cap N_i| > 1$ for any $i \in M_J$, then $f_d(J) = 0$ also.

Proof. Consider the constraint (2.11) for any $i \in M_J$, and any $J \subseteq N$, $|J| = d \in \{0,\ldots, \delta\}$. Since $|J \cap J_i| \geq 1$, the nonnegativity constraints (2.12) imply that $f_{d+1}(J + j) = 0$, $\forall j \in N_i - J \ \forall i \in M_J$. Moreover, if $|J \cap N_i| > 1$, then we also have form (2.11) and (2.12) that $f_d(J) = 0$. This completes the proof. ■

Hence, we can delete these null variables along with any resulting trivial constraints. In order to explicitly represent these zero variables and the resulting trivial constraints, we define some notation. For any $J \subseteq N$, let

$$\tilde{M}_J = M - \bigcup_{j \in J} M_J \equiv \bigcap_{j \in J} \overline{M}_j \equiv \{i \in M : N_i \bigcap J = \phi\}$$

$$N_{i,k} = \{j \in N_i : a_j^T a_k = 0\}, \quad i \in \overline{M}_k .$$

$$\tilde{N}_{i,J} = \bigcap_{j \in J} N_{i,j}, \quad i \in \overline{M}_k .$$

Hence, $\tilde{M}_J$ represents the set of rows in M not containing any variables from the set J, and $\tilde{N}_{i,J}$ represents the variables in row i which do not simultaneously appear with any variables from J in any row of SP. Then, by Proposition 2.2, we have that

$SPP_\delta = \{ \, (x,w) :$

$$\sum_{j \in N_i} x_j = 1 \quad \forall i \in M \tag{2.13}$$

$$[\sum_{t \in \tilde{N}_{i,J}} w_{J \bigcup \{t\}} = w_J \quad \forall i \in \tilde{M}_J \,], \quad \forall J \subseteq N \ni |J| = d, \; d = 1, \ldots, \delta \tag{2.14}$$

$$[w_J \geq 0, \; w_{J \bigcup \{t\}} \geq 0 \;\; \forall t \in \tilde{N}_{i,J}, \;\; \forall i \in \tilde{M}_J], \;\; \forall J \subseteq N \ni |J| = d, \, d = 1, \ldots, \delta \,\} . \tag{2.15}$$

where w_J and $w_{J \bigcup \{t\}}$ are the linearized terms for the products $\prod_{j \in J} x_j$ and $\prod_{j \in J \bigcup \{t\}} x_j$, respectively.

Note that SPP_n can be obtained by multiplying the exponential number of polynomial factors with the constraints, and subsequently linearizing these resulting constraints.

Hence, it may be impractical to obtain an explicit representation for SPP_n. However, using the structure of SPP, we can further reduce the size of the constraints in SPP_n.

Let G be an *intersection graph* of SPP. The vertex set of G is N and vertices i and j are connected by the edge (i, j) if and only if $a_i^T a_j \neq 0$. Without loss of generality, we assume that G is connected. Otherwise, the following results are valid for each component of G. We also assume that G is not a complete graph. (If G is a complete graph, then the problem SP is trivial.) A subset V of N is called an *independent set* of G if no two vertices of V are adjacent in G. An independent set is *maximal* if G has no independent set V′ with $|V'| > |V|$. The number of vertices in a maximal independent set of G is called the *independence number* of G, denoted by $\alpha(G)$. Then, we have $1 < \alpha(G) < n$.

Since $\sum_{j \in N} x_j \leq \alpha(G) \ \forall x \in SPP$, we have that $\prod_{j \in J} x_j = 0 \ \forall J \subseteq N \ni |J| > \alpha(G)$, i.e., $f_p(J) = 0$, $p = \alpha(G) + 1, \ldots, n$. Hence, we have the following result.

Proposition 2.3 Let $\alpha(G)$ be the independence number of the intersection graph of SPP. Then, $SPP \equiv SPP_{P\alpha(G)}$.

Remark 2.4 In a GUB constrained zero-one programming problem, where all the variables appear in the union of the GUB constraints, we can obtain a compact representation of the convex hull representation by constructing the relaxation $SPP_{|M_G|}$, where $|M_G|$ is the total number of GUB constraints. However, this might also be prohibitively expensive. One practical way of implementing the hierarchy of relaxations is to build a family of cutting planes. We can easily obtain a family of cutting planes from the first-order relaxation SPP_1. Then, the higher order RLT formulations can be used to strengthen this class of inequalities. This is addressed next.

2.3. A Family of Valid Inequalities for the Set Partitioning Polytope

Consider the first-order RLT formulation SPP_1 of SPP, given by (2.13) - (2.15), as follows. Note that in this relaxation, w_{jk} is the linearized term for the product x_jx_k. We will denote $w_{(jk)}$ to be w_{jk} if $j<k$ and w_{kj} otherwise.

$SPP_1 = \{ (x,w) :$

$$\sum_{j\in N_i} x_j = 1 \quad \forall\, i \in M \tag{2.16}$$

$$\sum_{j\in N_{i,k}} w_{(jk)} = x_k \quad \forall\, i \in \overline{M}_k, \quad \forall\, k \in N \tag{2.17}$$

$$[x_k \geq 0, \;\; w_{(jk)} \geq 0 \;\; \forall\, j \in N_{i,k}, \;\; \forall i \in \overline{M}_k], \; \forall k \in N \qquad \}. \tag{2.18}$$

Note that (2.17) are the constraints of type (2.14), and (2.18) is the set of nonnegativity constraints of type (2.15). Since every feasible solution x of SPP_{P1} is feasible to $\overline{SPP}$, we have that $SPP_{P1} \subseteq \overline{SPP}$. Hence, letting $\nu(P)$ denote the optimal objective function value for any problem P, $\nu(SP_1) \geq \nu(\overline{SP})$, where SP_1 : Minimize { cx : (x,w) satisfies (2.16), (2.17), (2.18) }, and where $\overline{SP}$ is the linear programming relaxation of SP.

Similarly, by (2.13) and (2.14), we can write the second-order ($\delta = 2$) RLT formulation SPP_2 of SPP as follows. Note here that w_{jkl} is the linearized term for the product $x_jx_kx_l$, for $j<k<l$, and where the indices are not necessarily so arranged, we simply write this product term as $w_{(jkl)}$.

$SPP_2 = \{ (x,w) :$

$$\sum_{j\in N_i} x_j = 1 \quad \forall\, i \in M \tag{2.19}$$

$$\sum_{j \in N_{i,k}} w_{(jk)} = x_k \quad \forall\, i \in \overline{M}_k, \quad \forall\, k \in N \tag{2.20}$$

$$\sum_{j \in (N_{i,k} \cap N_{i,l})} w_{(jkl)} = w_{(kl)} \quad \forall\, i \in \overline{M}_k \cap \overline{M}_l, \quad \forall\, k < l \in N \tag{2.21}$$

$$[x_l \geq 0,\ w_{(kl)} \geq 0,\ w_{(jkl)} \geq 0\ \forall\, j \in (N_{i,k} \cap N_{i,l}),\ \forall i \in (\overline{M}_k \cap \overline{M}_l)\,],\ \forall k < l \in N \quad \}. \tag{2.22}$$

Here again, (2.20) and (2.21) are the constraints (2.14) written for $|J| = 1$ and 2 respectively, and (2.22) are the nonnegativity constraints (2.15).

We now construct a class of valid inequalities implied by the constraints (2.17) as follows.

Proposition 2.4 For every $k \in N$ and $i \in \overline{M}_k$, the inequalities

$$x_k - \sum_{j \in N_{i,k}} x_j \leq 0 \tag{2.23}$$

are satisfied by all $x \in SPP$.

Proof. For each $k \in N$, consider the constraint (2.17). By the nonnegativity constraint (2.18), we have that $w_{(jk)} \leq x_k\ \forall\, j \in N_{i,k}$. Moreover, given any $\tilde{j} \in N_{i,k}$ for $i \in \overline{M}_k$, consider the constraint (2.17). Since $\tilde{j} \in N_{i,k}$, we have that $k \in N_{\iota,\tilde{j}}$ for some $\iota \in \overline{M}_{\tilde{j}}$. Hence, we have that $w_{(jk)} \leq x_j$ all $j \in N_{i,k}$ for all $i \in \overline{M}_k$. Hence, the constraint (2.17) implies that $x_k = \sum_{j \in N_{i,k}} w_{(jk)} \leq \sum_{j \in N_{i,k}} x_j$. Hence, it follows that $x_k - \sum_{j \in N_{i,k}} x_j \leq 0$. This completes the proof. ■

Remark 2.5 The inequality (2.23) asserts that when $x_k = 1$ at least one of the variables x_j, $j \in N_{i,k}$, must be one. Note that the inequality (2.23) is called an *elementary inequality* by Balas (1977) for the set partitioning problem. This class of elementary inequalities is building blocks for constructing certain types of strong valid inequalities. Balas develops strengthening procedures and composition rules to generate some valid inequalities from elementary inequalities. We will show that these strengthening procedures are implied by SPP_1 and SPP_2. Consequently, we can generalize Balas' approach to build an extended family of strong valid inequalities, which can be obtained from the first-order RLT formulation of SPP, and then successively strengthened by higher order RLT formulations of SPP. All of Balas' strong valid inequalities can be shown to be subsumed by such a family of strong valid inequalities.

We now consider a strengthening procedure for tightening the formulation of SPP_1, which consequently tightens the elementary inequality (2.23). Toward this end, for each $k \in N$, define $L(k)$ to be the index set of those columns orthogonal to a_k, and by $\overline{L}(k)$ its complement, i.e. ,

$$L(k) = \{j \in N : a_j^T a_k \ = \ 0\} \equiv \bigcup_{i \in \overline{M}_k} N_{i,k}\,, \ \ \overline{L}(k) = N - L(k).$$

By Proposition 2.2, we have that $w_{(jk)} = 0$ for $j \in \overline{L}(k)$ for each $k \in N$. Moreover, the structure of the set partitioning problem implies that there may exist some $j \in L(k)$ for which $x_k = 1$ implies that $x_j = 0$, and vice versa. We want to systematically find indices $j \in L(k)$ such that $x_k + x_j \leq 1$ for any vertex x of SPP. From SPP_1, we have that $w_{(jk)} = 0$ for all feasible solutions if and only if $x_k + x_j \leq 1$ for any vertex x of SPP. For a given $k \in N$, let $N_k^0 \ = \ \{\ j \in L(k)\text{:}\ \ x_j = 0,\ \forall\ x \in \{Ax = e,\ x_k = 1,\ x_j \in (0,1)\ \forall j \in N\}$. As might be expected, finding the set N_k^0 is as hard as finding an optimal solution to SP. However, we can easily construct a subset of N_k^0 for some $k \in N$ as follows.

Recall the following constraints (2.21) of SPP_2:

$$w_{(kl)} \;=\; \sum_{j \in (N_{i,k} \cap N_{i,l})} w_{(jkl)} \qquad \forall\, i \in \overline{M}_k \cap \overline{M}_l, \qquad \forall\, k < l \in N.$$

The logical structure of the constraints (2.21) can be used to find a subset of the set N_k^0. For a given $k \in N$, we define $z_{(kl)}$ for some $l \in L(k)$ as

$$z_{(kl)} = \text{maximum}\{\, w_{(kl)} : \quad w_{(kl)} = \sum_{j \in (N_{i,k} \cap N_{i,l})} w_{(jkl)},\ \forall i \in \overline{M}_k \cap \overline{M}_l, \quad w_{(jkl)} \geq 0\ \forall (jkl),\ w_{(kl)} \leq 1\ \ \forall (kl) \quad \}. \tag{2.24}$$

By its structure, $z_{(kl)} = 1$ or 0 because if $w_{(kl)} > 0$, then $w_{(kl)}$ can be assigned to be 1. Hence, if $z_{(kl)} = 0$, then $w_{(kl)} = 0$, i.e., $l \in N_k^0$. Therefore, we can delete $w_{(kl)}$ from the first-order RLT formulation.

The above procedure for detecting a set of zero $w_{(kl)}$'s generalizes Balas' two procedures for strengthening elementary inequalities. We show below that these two procedures are trivial sufficient conditions for the optimal solution of (2.24) to be zero. Toward this end, consider the following proposition.

Proposition 2.5 For $k \in N$, $i \in \overline{M}_k$, and $Q \subseteq N_{i,k}$, the following inequality

$$x_k - \sum_{j \in Q} x_j \leq 0 \tag{2.25}$$

is valid for SPP if and only if $w_{(jk)} = 0 \ \forall j \in N_{i,k} - Q$.

Proof. Suppose that (2.25) is valid for SPP. Then, the inequality $x_k - \sum_{j \in Q} w_{(jk)} \leq 0$ is valid for SPP$_1$. From equation (2.17), we have that $w_{(jk)} = 0$ for all $j \in (N_{i,k} - Q)$ because $w_{(jk)} \geq 0 \ \forall j, k$. On the other hand, if $w_{(jk)} = 0$ for all $j \in (N_{i,k} - Q)$, then the equality (2.17) becomes $\sum_{j \in Q} w_{(jk)} = x_k$. This implies that $x_k - \sum_{j \in Q} x_j \leq 0$ is valid for SPP and this completes the proof. ■

We now consider Balas' strengthening procedures.

Proposition 2.6 (Proposition 3.1, Balas, 1977) For some $k \in N$, let the index sets $Q_i \subseteq N_{i,k}$, $i \in \overline{M}_k$, be such that the inequalities $x_k - \sum_{j \in Q_i} x_j \leq 0$, $i \in \overline{M}_k$ are satisfied by all $x \in P$. For each $j \in \bigcup_{h \in \overline{M}_k} Q_h$, define $Q(j) = \bigcup_{h \in \overline{M}_k \ni j \in Q_h} Q_h \setminus \{j\}$ and for $i \in \overline{M}_k$, let $T_i = \{j \in Q_i : Q_h \subseteq Q(j)$ for some $h \in \overline{M}_k\}$. Then, the inequalities $x_k - \sum_{j \in Q_i \setminus T_i} x_j \leq 0 \ \forall i \in \overline{M}_k$ are satisfied by all $x \in$SPP. ■

By Proposition 2.5, we have that $w_{jk} = 0 \ \forall \ j \in N_{i,k} - Q_i$. Hence, for the simplicity of analysis, let us suppose that $Q_i = N_{i,k}$. Since for any $l \in T_i$, $i \in \overline{M}_k$, we have $Q_h \subseteq Q(l)$ for some $h \in \overline{M}_k$, we then have that $N_{h,k} \cap N_{h,l} = \phi$, which implies that $w_{(kl)} = \sum_{j \in N_{h,k} \cap N_{h,l}} w_{(jkl)} = 0$. Hence, Proposition 2.6 is a trivial sufficient condition to guarantee that $z_{(kl)} = 0$ in (2.24).

Example 2.1 (Example 3.1 in Balas, 1977) Consider the following numerical example, i.e., the set partitioning polytope with coefficient matrix A (where the blanks are zeros.)

$$
\begin{array}{c|ccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & 1 & 1 & \\
2 & 1 & 1 & & & & & & & & 1 & 1 & & & & 1 \\
3 & & 1 & 1 & & & 1 & 1 & 1 & & & & 1 & & & \\
4 & & 1 & 1 & 1 & & & & 1 & 1 & & 1 & & & & \\
5 & & & 1 & 1 & 1 & & 1 & & & 1 & & 1 & 1 & &
\end{array}
$$

Consider the strengthening procedure of Proposition 2.6 to strengthen the inequality $x_1 - x_3 - x_{12} \leq 0$ associated with $N_{3,1}$, where for $k=1$, $\overline{M}_1 = \{ 3,4,5\}$, and $N_{3,1} = \{ 3,12 \}$, $N_{4,1} = \{3,4\}$, and $N_{5,1} = \{3,4,5,12\}$. We have that $Q(3) = \{4,5,12\}$, $Q(12)=\{3,4,5\}$, and we find that $N_{4,1} \subseteq Q(12)$. Hence, $T_3 = \{12\}$, and the above inequality can be replaced by $x_1 - x_3 \leq 0$.

We now consider the equality of type (2.20) for $k=1$.

$$
\begin{aligned}
x_1 &= w_{1,3} + w_{1,12} && \text{for } i=3 \\
x_1 &= w_{1,3} + w_{1,4} && \text{for } i=4 \\
x_1 &= w_{1,3} + w_{1,4} + w_{1,5} + w_{1,12} && \text{for } i=5
\end{aligned}
$$

Consider the form of equality (2.21) for $l=12$ in the corresponding second-order RLT formulation. Since $\overline{M}_{12} = \{1,2,4\}$, we have that $\overline{M}_1 \cap \overline{M}_{12} = \{4\}$ and $N_{4,12} = \{1, 9, 11, 13, 14\}$, $N_{4,4} = \{3, 4\}$. Therefore, $N_{4,12} \cap N_{4,4} = \phi$. Hence, the inequality of type (2.21) for $k=1$ and $l=12$ is

$$w_{1,12} = \sum_{l \in N_{4,4} \cap N_{4,12}} w_{1,12,l} = 0 . \blacksquare$$

Remark 2.6 For notational convenience, we can rewrite the system (2.24) as $Fw = 0$. If $\mathrm{rank}(F) = |\overline{M}_k \cap \overline{M}_l| = |\bigcup_{i \in (\overline{M}_k \cap \overline{M}_l)} (N_{i,k} \cap N_{i,l})| + 1$, then the unique solution of the LP (2.24) is zero. Hence, $w_{(kl)} = 0$. The following example illustrates this case.

Example 2.2 (Example 3.2 in Balas, 1977) Consider the following numerical example.

	1	2	3	4	5	6	7	8	9	10
1	1								1	1
2	1									1
3		1			1		1		1	
4		1				1		1		
5			1		1			1		1
6			1	1		1			1	
7				1	1		1			

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 $N_{7,1} = \{4, 5, 7\}$.

Any attempt to apply Balas' first strengthening procedure fails to strengthen any of the elementary inequalities associated with $k = 1$. On the other hand, we can apply the rule mentioned in Remark 2.6. The constraints (2.20) can be stated as follows:

$$x_1 = w_{1,2} + w_{1,5} + w_{1,7}$$

$$x_1 = w_{1,2} + w_{1,6} + w_{1,8}$$

$$x_1 = w_{1,3} + w_{1,5} + w_{1,8}$$

$$x_1 = w_{1,3} + w_{1,4} + w_{1,6}$$

$$x_1 = w_{1,4} + w_{1,5} + w_{1,7}.$$

Consider the second-order constraints of type (2.21) for $l=2$ and $k=1$, where $\overline{M}_2$ = $\{1, 2, 5, 6, 7\}$, so that $\overline{M}_1 \cap \overline{M}_2 = \{5, 6, 7\}$ and

$$N_{5,1} \cap N_{5,2} = \{3\}$$
$$N_{6,1} \cap N_{6,2} = \{3, 4\}$$
$$N_{7,1} \cap N_{7,2} = \{4\}\ .$$

Hence, the constraints of type (2.21) for $k=1$ and $l=2$ are

$$w_{1,2} = w_{1,2,3}$$
$$w_{1,2} = w_{1,2,3} + w_{1,2,4}$$
$$w_{1,2} = w_{1,2,4}\ .$$

Since the rank of the corresponding matrix F in Remark 2.6 is 3, it follows that $w_{1,2} = w_{1,2,3} = w_{1,2,4} = 0$. Hence, the elementary inequality associated with $N_{3,1}$, $x_1 - x_2 - x_5 - x_7 \leq 0$ can be strengthened to $x_1 - x_5 - x_7 \leq 0$. ■

We now consider Balas' second strengthening procedure.

Proposition 2.7 (Proposition 3.2, Balas, 1977) Let the index sets $Q_{ik} \subseteq N_{ik}$, $i \in \overline{M}_k$, $k \in N$, be such that the inequalities $x_k - \sum_{j \in Q_{ik}} x_j \leq 0$, $i \in \overline{M}_k$, $k \in N$ are satisfied by all $x \in SPP$. For $i \in \overline{M}_k$, $k \in N$, define

$$U_{ik} = \{j \in Q_{ik} : Q_{hk} \bigcap Q_{hj} = \phi \text{ for some } h \in \overline{M}_k \bigcap \overline{M}_j\}. \qquad (2.26)$$

Then, the inequalities $x_k - \sum_{j \in Q_{ik} \setminus U_{ik}} x_j \leq 0$, $i \in \overline{M}_k$, $k \in N$ are satisfied by all $x \in SPP$. ∎

The foregoing strengthening procedure of Proposition 2.7 can be easily obtained because condition (2.26) in Proposition 2.7 is a trivial sufficient condition for ensuring that the corresponding linear program (2.24) has an objective value zero, since it includes the trivial constraints $w_{(jk)} = 0$ for any $j \in U_{ik}$.

Remark 2.7 We have shown that Balas' strengthening procedures for the elementary inequalities are an evident consequence of the second-order RLT formulation of SPP. Of course, higher order RLT formulations can be used for further strengthening the valid inequalities obtained from the first-order RLT formulation of SPP. For example, in the second-order RLT formulation of SPP, suppose that $w_{(jk)} = w_{(jkl)}$ for some l and that $w_{(jkl)} = \sum_{t \in N_{i,k} \bigcap N_{i,l} \bigcap N_{i,j}} w_{tjkl}$, for $i \in \overline{M}_j \bigcap \overline{M}_k \bigcap \overline{M}_l$. If $N_{i,k} \bigcap N_{i,l} \bigcap N_{i,j} = \phi$, then $w_{(jkl)} = 0$ and consequently $w_{(jk)} = 0$. In other words, for a given δ, we can tighten the formulation of SPP_δ by using the logical implications of the structure of $SPP_{\delta+1}$ for $\delta = 1, \ldots, \alpha(G) - 1$. 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 two composition rules that can be used to combine inequalities in a certain class (which contains as a subclass the elementary inequalities) into a new inequality belonging to the same class and stronger that the sum of the inequalities from which it was obtained. The first type of composite inequality is of the form

$$x_k - \sum_{j \in S} x_j \leq 0 \tag{2.27}$$

where $a_k^T a_j = 0 \quad \forall\ j \in S$. This subclass of elementary inequalities is distinguished by the additional property that $S \subseteq N_{i,k}$ for some $i \in \overline{M}_k$. The first composition rule, given in the next proposition, generates a new inequality (2.27) from a pair of inequalities of type (2.23). This process can be sequentially applied to a pair of inequalities from the combined set (2.23) and (2.27) thus generated, provided that such a pair satisfies the requirement of the proposition. After stating Balas' results, we show that the inequalities of type (2.27) are implied by the second-order RLT formulation of SPP.

Proposition 2.8 (Proposition 5.1, Balas, 1977) For k, $h \in N$, let $S_r \subseteq L(r)$, $r = k, h$, be such that $h \in S_k$, $k \notin S_h$, and that the inequalities

$$x_r - \sum_{j \in S_r} x_j \leq 0, \quad r = k,h \tag{2.28}$$

are satisfied by all $x \in P$. Then all $x \in SPP$ satisfy the inequality

$$x_k - \sum_{j \in S} x_j \leq 0 \tag{2.29}$$

where $S = (S_k \backslash \{h\}) \bigcup [S_h \bigcap L(k)]$. Furthermore, (2.29) is stronger than the sum of the two inequalities (2.28) if and only if $S_h \bigcap [S_k \bigcup \overline{L}(k)] \neq \phi$. ■

We want to show that the composite inequality of type (2.29) is implied by the constraints of SPP_2. (The statement regarding the relative strength of (2.28) is readily evident.) Since the inequalities of type (2.28) are valid for SPP, by Proposition 2.5, we have that $w_{jk} = 0$ for $j \in N_{ik} - S_k$. We then have the following first-order RLT formulation associated with the constraints (2.28).

$$x_k = \sum_{j \in S_k} w_{(jk)} = w_{(kh)} + \sum_{j \in S_k \backslash \{h\}} w_{(jk)} \quad (\text{ since } h \in S_k \;) \tag{2.30}$$

$$x_h = \sum_{j \in S_h} w_{(jh)} \tag{2.31}$$

Hence, we have that

$$w_{(jkh)} = 0 \quad \forall j \in S_k \backslash \{h\}. \tag{2.32}$$

The second-order RLT constraints of type (2.21) corresponding to (2.31) are

$$w_{(kh)} = \sum_{j \in S_h} w_{(jkh)} \,.$$

Since $w_{(jkh)} = 0$ for $j \notin L(k)$ and $w_{(jkh)} = 0$ for $j \in (S_k \backslash \{h\})$ by (2.32), it follows that

$$w_{(kh)} = \sum_{j \in (S_h \cap L(k) - [S_k \backslash \{h\}])} w_{(jkh)} \,.$$

By substituting $w_{(kh)}$ in (2.30), we have that

$$x_k = \sum_{j \in (S_h \cap L(k) - [S_k \setminus \{h\}])} w_{(jkh)} + \sum_{j \in (S_k \setminus \{h\})} w_{(jk)} \ . \qquad (2.33)$$

Since $w_{(jkh)} \leq x_j$ and $w_{(jk)} \leq x_j$, (2.33) implies that

$$x_k \leq \sum_{j \in (S_h \cap L(k) - [S_k \setminus \{h\}])} x_j + \sum_{j \in (S_k \setminus \{h\})} x_j \ .$$

Equivalently,

$$x_k - \sum_{j \in S} x_j \leq 0$$

where $S = (S_k \setminus \{h\}) \cup (S_h \cap L(k))$.

Hence, the valid inequality (2.29) is implied if we implement the second-order RLT formulation.

Next, we state Balas' second composition rule, which can be used to obtain all valid inequalities of type (2.29) for a certain index $k \in N$ from the set of elementary inequalities (2.23) corresponding to the same index k.

Proposition 2.9 (Proposition 5.2, Balas, 1977) For some $k \in N$, let the sets $Q_i \subseteq N_{i,k}$, $i \in \overline{M}_k$, be such that the inequalities

$$x_k - \sum_{j \in Q_i} x_j \leq 0,\ i \in \overline{M}_k \tag{2.34}$$

are satisfied by all $x \in SPP$, and let $Q_0 = \bigcup_{i \in \overline{M}_k} Q_i$. Then the inequality

$$x_k - \sum_{j \in S} x_j \leq 0, \tag{2.35}$$

where $S \subseteq N - \{k\}$, is satisfied by all $x \in SPP$, if and only if

$$\sum_{j \in Q} a_j \neq e - a_k, \quad \forall Q \subseteq Q_0 \backslash S\ .\ \blacksquare \tag{2.36}$$

Since (2.34) is valid for SPP, by Proposition 2.5, it follows that $w_{(jk)} = 0\ \forall j \in N_{i,k} - Q_i$ $\forall i \in \overline{M}_k$. Hence, we may assume that $Q_i = N_{i,k}$ for the convenience of analysis. Note that since $x_j \geq 0\ \forall j \in S$, the inequality (2.35) is trivially valid for SPP whenever $x_k = 0$. If $x_k = 1$, the inequality (2.35) is valid to SPP if and only if

$$\text{maximum}\{1 - \sum_{j \in S} x_j\ :\ \sum_{j \in N_{i,k}} x_j = 1,\ \forall i \in \overline{M}_k\ \ x_j \in (0,1)\} \leq 0. \tag{2.37}$$

Note that the condition (2.37) is equivalent to the condition (2.36) which states that there does not exist any feasible solution to SPP if $x_k = 1$ and $x_j = 0\ \forall j \in S$. Hence, we need to establish (2.37) with respect to an integer programming problem in order to construct a valid inequality of type (2.35) from the inequalities (2.34), a potentially exponential task. However, we would like to conjecture that a class of valid inequality of the form (2.35) that is derived from (2.34) is implied by SPP_1.

In SPP_1, we have the following (implied) constraints for some $k \in N$.

$$x_k = \sum_{j\in Q_i} w_{(jk)} \quad \forall i\in\overline{M}_k \tag{2.38}$$

$$x_j \geq w_{(jk)} \quad \forall j\in Q_0 \tag{2.39}$$

We want to show that a valid inequality of type (2.35) can be obtained by surrogating the constraints (2.38) and (2.39). Let μ_i and β_j be the surrogate multipliers corresponding to each constraint (2.38) for $i\in\overline{M}_k$, and each constraint (2.39) for $j\in Q_0$, respectively. Using these multipliers, we have the following surrogate inequality.

$$\sum_{i\in\overline{M}_k}\mu_i x_k - \sum_{i\in\overline{M}_k}\mu_i\sum_{j\in Q_i} w_{(jk)} = 0$$

$$-\sum_{j\in Q_0}\beta_j x_j + \sum_{j\in Q_0}\beta_j w_{(jk)} \leq 0$$

Hence, it follows that

$$(\sum_{i\in\overline{M}_k}\mu_i)x_k - \sum_{j\in Q_0}\beta_j x_j + \sum_{j\in Q_0}(\beta_j - \sum_{i\in M_j\cap\overline{M}_k}\mu_i)w_{(jk)} \leq 0 \tag{2.40}$$

Suppose that $(\overline{\mu}, \overline{\beta})$ is a feasible solution satisfying the following linear system.

$$\beta_j - \sum_{i\in M_j\cap\overline{M}_k}\mu_i = 0 \quad \forall j\in Q_0, \tag{2.41}$$

$$\beta_j \geq 0, \quad \mu_i : \text{unrestricted } \forall i\in\overline{M}_k \tag{2.42}$$

Conjecture: In order to show that the inequality of type (2.35) can be constructed by this surrogation procedure, it is sufficient to show that there is some $S\subseteq N\backslash\{k\}$ such that there exists a feasible solution (μ, β) of (2.41) and (2.42) such that $\beta_j = 1$ for $j \in S$ and $\beta_j = 0$ for $j \in (Q_0 - S)$, and $\sum_{i \in \overline{M}_k} \mu_i = 1$. Alternatively, using a surrogate of (2.38) and (2.39), we want to claim that there exists a feasible solution to the following system M.

$$M \equiv \{\ \mu_i \text{ unrestricted } \forall i \in \overline{M}_k :\ \sum_{i \in (\overline{M}_k \cap \overline{M}_j)} \mu_i = 0\ \forall j \in S,\ \ \sum_{i \in (M_j \cap \overline{M}_k)} \mu_i = 0\ \forall j \in (Q_0 - S),\ \sum_{i \in \overline{M}_k} \mu_i = 1\ \}.$$

The above conjecture is illustrated by the following example.

Example 2.3 Consider the following set-partitioning problem.

SP : Maximize $\{\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 :\ x_1 + x_2 + x_3 + x_7 = 1,\ x_1 + x_3 + x_4 + x_8 = 1,$ $x_1 + x_4 + x_5 + x_9 = 1,\ x_1 + x_5 + x_6 + x_{10} = 1,\ x_1 + x_2 + x_6 + x_{11} = 1,\ x_j \in (0,1)\ j = 1,\dots,11\ \}$.
The optimal solution $\overline{x}$ of $\overline{SP}$ is

$$(\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4, \overline{x}_5, \overline{x}_6, \overline{x}_7, \overline{x}_8, \overline{x}_9, \overline{x}_{10}, \overline{x}_{11}) = (0,\ 1/2,\ 1/2,\ 1/2,\ 1/2,\ 1/2,\ 1/2,\ 0,\ 0,\ 0,\ 0,\ 0).$$

Note that no elementary inequality cuts off this optimal basic feasible solution $\overline{x}$. However, the above procedure generates a valid inequality which cuts off $\overline{x}$. We can construct SP_1 with respect to x_2 by adding the constraints (2.38) and (2.39) to SP. That is,

$SP_1(x_2)$: Maximize $x_1 + x_2 + x_3 + x_4 + x_5 + x_6$

subject to

$$x_1 + x_2 + x_3 + x_7 = 1$$
$$x_1 + x_3 + x_4 + x_8 = 1$$
$$x_1 + x_4 + x_5 + x_9 = 1$$
$$x_1 + x_5 + x_6 + x_{10} = 1$$
$$x_1 + x_2 + x_6 + x_{11} = 1$$

$$x_2 = w_{2,4} + w_{2,8}$$
$$x_2 = w_{2,4} + w_{2,5} + w_{2,9}$$
$$x_2 = w_{2,5} + w_{2,10}$$

$$w_{2,4} \leq x_4$$
$$w_{2,5} \leq x_5$$
$$w_{2,8} \leq x_8$$
$$w_{2,9} \leq x_9$$
$$w_{2,10} \leq x_{10}$$
$$w_{2,k} \geq 0 \quad k = 4,5,8,9,10$$
$$x_j \in (0,1), \quad j \in \{1,\ldots,11\} .$$

The optimal solution of $\overline{SP}_1(x_2)$ is all integer. For example, using the optimal dual vector, we can construct the following surrogate inequality of type (2.35):

$$x_2 - x_8 - x_{10} \leq 0 \tag{2.43}$$

Moreover, we can construct an equivalent nonhomogeneous inequality 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 (2.43)

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2. \tag{2.44}$$

It is easily shown that the inequality (2.44) is a lifting of the odd hole inequality for the underlying intersection graph G of SPP. ■

Remark 2.8 The above procedure of finding a class of valid inequalities can be easily extended for finding valid inequalities of the type $\sum_{j \in K} x_j - \sum_{j \in S} x_j \leq 0$, which can be obtained by surrogating the constraints of SPP_1.

Remark 2.9 We remark here that an inequality of the type (2.35) can be obtained by using a surrogate inequality from SPP_1 and the integrality condition. If $\sum_{i \in \overline{M}_k} \overline{\mu}_i \leq 0$, then the inequality (2.40) is implied by the nonnegativity restrictions. Hence, suppose that $\sum_{i \in \overline{M}_k} \overline{\mu}_i > 0$. For each $j \in Q_0$, define $\alpha_j = \overline{\beta}_j / \sum_{i \in \overline{M}_k} \overline{\mu}_i$. Since $x_j \leq 1$ $\forall j \in N$, all $\alpha_j > 1$ can be replaced by 1 without cutting off any $x \in SPP$. Hence we have a valid inequality for SPP as follows.

$$x_k - \sum_{j \in Q_F} \alpha_j x_j - \sum_{j \in Q_I} x_j \leq 0 \tag{2.45}$$

where $Q_F = \{j \in Q_0 \colon 0 < \alpha_j \leq 1\}$, and $Q_I = \{j \in Q_0 \colon \alpha_j > 1\}$.

This further implies trivially an inequality of the form (2.35) as follows.

$$x_k - \sum_{j \in Q_F} x_j - \sum_{j \in Q_I} x_j \leq 0 \tag{2.46}$$

Evidently, a necessary condition for a valid inequality (2.35) or (2.46) to be maximal is that the set S is minimal ($Q_F \bigcup Q_I$ is minimal). Finding a maximal inequality of the form (2.46) has a combinatorial nature. Hence, we want to find a cutting plane of the form (2.45) that cuts off a given fractional optimal basic feasible solution of $\overline{SP}$. This issue will be addressed in the next Section.

2.4. A Family of Cutting Planes for the Set Partitioning Polytope

An inequality

$$\pi x \leq \pi_0 \tag{2.47}$$

satisfied by all $x \in SPP$ is called a *valid inequality* for SPP. A valid inequality (2.47) is a *cut*, or *cutting plane* if it deletes some $x \in \overline{SPP} \backslash SPP$.

Proposition 2.10 Let $\bar{x} = (\bar{x}_F, \bar{x}_{\bar{F}})$, where $F = \{ j \in N : \bar{x}_j$ is fractional $\}$ and $\bar{F} = N - F$, be a basic feasible solution of $\overline{SPP}$ with $F \neq \phi$. Then SPP_1 cuts off $\bar{x}$.

Proof. Suppose on the contrary that $\bar{x}$ remains feasible to SPP_1 , along with some associated value $\bar{w}$ of w. Since $\bar{x}$ is a basic feasible solution, there exists a basis B such that $B\bar{x}_B = e$, where x_B is the vector of basic variables corresponding to B. Hence, for any basic variable x_k, $k \in F$, $(B^{-1}e)_k \neq 1$ where $(B^{-1}e)_k$ is the kth element of $B^{-1}e$.

Since $(\bar{x}, \bar{w})$ must satisfy (2.17), and since $x_j = 0$ implies that $w_{(jk)} = 0 \; \forall j$, we have $B\overline{W}_k = e\bar{x}_k \; \forall k \in B$, where $W_k = (w_{(jk)}, j \in B)$, and where $j \in B$ means that x_j is basic. For each $k \in B \bigcap F$, since $\bar{w}_{kk} \equiv \bar{x}_k \neq 0$, and since $\overline{W}_k = (B^{-1}e)\bar{x}_k$, the kth component of

$B^{-1}e$ is equal to one, i.e., $(B^{-1}e)_k = 1 \ \forall k \in B \cap F \neq \phi$, a contradiction. This completes the proof. ■

Proposition 2.10 shows that there exists a family of cutting planes inherent in SPP_1. We consider a procedure to generate a class of cutting planes, which cut off the fractional basic feasible solutions of $\overline{SPP}$. For this purpose, we may write (2.17) and (2.18) as follows.

$$Dw = Ex, \quad w \geq 0, \quad x \geq 0. \tag{2.48}$$

Associating Lagrangian multiplier vectors π with respect to (2.48), we have by linear programming duality that $x \in SPP_1(x) = \{ x : (x,w) \text{ is feasible to } SPP_1 \}$ if and only if (2.16) holds, $x \geq 0$, and $0 = \min\{0w: \ Dw = Ex, \ w \geq 0\}$, i.e.,

$$0 = \max \{ \pi Ex: \quad \pi \in PC \} \tag{2.49}$$

where PC is a polyhedral cone defined as

$$PC = \{\pi: \pi D \leq 0, \quad \pi \text{ unrestricted} \}$$

Hence, we have the following result.

Proposition 2.11 Suppose that D has full row rank so that PC is a pointed cone with vertex at the origin. Let $\overline{\pi}_l$, $l = 1,\ldots,L$ be the set of generators of the polyhedral cone PC. Then (2.49) holds if and only if

$$\overline{\pi}_l Ex \leq 0 \quad \forall \ l = 1,\ldots,L . \tag{2.50}$$

In particular, for any $\overline{\pi} \in PC$, the inequality

$$\bar{\pi} E\, x \leq 0 \qquad (2.51)$$

is satisfied by all x∈ SPP. ■

Consequently, any inequality (2.51) is a valid inequality for SPP. Note that the inequality (2.23) can be obtained by selecting a particular $\bar{\pi}$ that belongs to the polyhedral cone PC. Of course, a total enumeration of the inequalities (2.50) is prohibitive. But we can construct a separation problem that generates a strong valid inequality that cuts off $\bar{x} \in \overline{SPP} \setminus SPP_1$.

Separation Problem : Given a fractional basic feasible solution $\bar{x} = (\bar{x}_F, \bar{x}_{\bar{F}}) \in \overline{SPP}$, find an inequality of type (2.50) that cuts off $\bar{x}$.

We consider a linear programming problem to solve this separation problem, which attempts to generate a normalization inequality (2.51) with the most negative slack variables, and so in this sense, a deepest valid cut of this type.

$$SPP_{projection}: \text{ maximize } \{\, (\pi^+ - \pi^-)E\bar{x}: \; (\pi^+ - \pi^-)D \leq 0,$$

$$\sum_j (\pi_j^+ + \pi_j^-) \leq 1,$$

$$\pi^+ \geq 0, \quad \pi^- \geq 0 \,\}$$

Let $(\tilde{\pi}^+, \tilde{\pi}^-)$ be an optimal solution of $SPP_{projection}$ with a positive objective value. Then, we have the following cutting plane, which cuts off $\bar{x}$ where $\tilde{\pi} = \tilde{\pi}^+ - \tilde{\pi}^-$,

$$\tilde{\pi} E x \leq 0 . \qquad (2.52)$$

Conversely, if there exists a valid inequality $\pi Ex \leq 0$ that cuts off $\bar{x}$, then the inequality (2.52) generated from an optimal solution to $SPP_{Projection}$ will also cut off $\bar{x}$. Since the size of problem SPP_1 can be large, the construction of the above separation problem can be an ineffective procedure for finding a strong valid inequality of type (2.52). Hence, we need to devise a scheme to reduce the size of the problem SPP_1, while maintaining the properties of the strong formulation. This issue will be addressed next.

We now consider another RLT formulation of SPP. Let $\bar{x}$ be an optimal basic feasible solution to $\overline{SPP}$. The motivation of the revised RLT formulation is based on the observation that we can imagine as if the fractional variables are being treated as binary variables and the others as continuous, in an attempt to tighten the formulation. Consider the set of constraints generated via products with factors involving only the fractional variables x_j $j\in F$. Denote $w_{ii} \equiv x_i$, $w_{(jk)} = w_{jk}$ if $j < k$ and w_{kj} otherwise, $\forall j, k \in F$ and $v_{jk} = x_j x_k$ $\forall j \in \bar{F}$, $k \in F$. Then a first-order RLT formulation with this restricted set of products can be described as follows.

$$\sum_{j \in N_i} x_j = 1 \quad \forall \; i \in M \tag{2.53}$$

$$\sum_{j \in N_{i,k} \cap F} w_{(jk)} + \sum_{j \in N_{i,k} \cap \bar{F}} v_{jk} = x_k \quad \forall \; i \in \overline{M}_k, \quad \forall \; k \in F \tag{2.54}$$

$$w_{(jk)} \geq 0 \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F \tag{2.55}$$

$$w_{(jk)} \leq x_j \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F \tag{2.56}$$

$$w_{(jk)} \leq x_k \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F \tag{2.57}$$

$$w_{(jk)} \geq x_j + x_k - 1 \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F \tag{2.58}$$

$$v_{jk} \geq 0 \quad \forall j \in \tilde{N}_k \cap \overline{F}, \quad k \in F \tag{2.59}$$

$$v_{jk} \leq x_j \quad \forall j \in \tilde{N}_k \cap \overline{F}, \quad k \in F \tag{2.60}$$

$$v_{jk} \leq x_k \quad \forall j \in \tilde{N}_k \cap \overline{F}, \quad k \in F \tag{2.61}$$

$$v_{jk} \geq x_j + x_k - 1 \quad \forall j \in \tilde{N}_k \cap \overline{F}, \quad k \in F \tag{2.62}$$

$$x_j \geq 0, \ \forall j \in N \tag{2.63}$$

where $\tilde{N}_k \equiv \bigcup_{i \in \overline{M}_k} N_{i,k}$.

Proposition 2.12 The constraints (2.56), (2.57), (2.58), (2.61), and (2.62) are implied by (2.53), (2.54), (2.55), (2.59), (2.60) and (2.63).

Proof. The constraints (2.56), (2.57) and (2.61) are implied by (2.54) and the nonnegativity constraints (2.55), (2.59).

Next, consider (2.58) for $j \in \tilde{N}_k \cap F, k \in F$. For any $k \in F$, we have that for any $i \in \overline{M}_k \ni j \in N_{ik}$, by (2.54),

$$w_{(jk)} + \sum_{t \in N_{i,k} \cap F - j} w_{(tk)} + \sum_{s \in N_{i,k} \cap \overline{F}} v_{sk} = x_k. \tag{2.64}$$

Since $w_{(tk)} \leq x_t$ and $v_{sk} \leq x_s$, (2.64) implies that

$$w_{(jk)} + \sum_{t\in N_{i,k}\cap F - j} x_t + \sum_{s\in N_{i,k}\cap \bar{F}} x_s \;\geq\; x_k \,. \tag{2.65}$$

By (2.53) and the nonnegativity constraints (2.63), we have that

$$1 = \sum_{l\in N_i} x_l \geq x_j + \sum_{t\in(N_{i,k}\cap F - \{j\})} x_t + \sum_{s\in N_{i,k}\cap \bar{F}} x_s$$

Using this in (2.65), $w_{jk} \geq x_j + x_k - 1 \;\forall j\in \tilde{N}_k \cap F,\; k\in F$ and so, (2.58) is implied. In the same way, it can be shown that $v_{jk} \geq x_j + x_k - 1 \;\forall j\in \tilde{N}_k \cap \bar{F},\; k\in F$. This completes the proof. ∎

By Proposition 2.12, we can delete the redundant inequalities, (2.56), (2.57), (2.58), (2.61), and (2.62). Hence, the revised first-order RLT formulation of SPP is to minimize $\{ \sum_{j\in N} c_j x_j$: (x,w) satisfies (2.53), (2.54), (2.55), (2.59), (2.60), (2.63) $\}$.

Remark 2.10 As it turns out, in order to generate a tighter reformulation of SPP that cuts off the linear programming solution $\bar{x}$, we can further restrict the multiplication of each fractional variable x_k, $k\in F$, to a subset of the constraints in $\overline{M}_k$. This is shown below.

Suppose that we fix all the variables x_j in $\bar{F}$ and examine the separable sets of constraints produced thereby. Let $S_1, \ldots, S_p$ be the partition of F corresponding to the, say, p separable sets, 1,...,p, produced in this manner. Then we can multiply each

$x_j \in S_t$ with each of the constraints in set t alone, and then construct all bound factor products (2.55), ..., (2.62) for all created product terms. The motivation is that the separable linear programs for each t produce fractional solutions, and so RLT can be used to eliminate these solutions.

Suppose that a basic feasible solution $\bar{x} = (\bar{x}_F, \ \bar{x}_{\bar{F}})$ solves $\overline{SP}$. If $\bar{x}_{\bar{F}}$ is fixed as above, the resulting surviving (nontrivial) constraint sets are of the type

$$[\sum_{j \in S_t \cap N_i} x_j = 1 \ \text{ for } i \ni \ S_t \cap N_i \neq \phi \ ], \ \text{ for } \ t = 1,\ldots,p\,. \qquad (2.66)$$

Applying RLT on SPP by multiplying each x_j, $j \in S_t$, with each constraint in set t, and then using Proposition 2.12 to eliminate redundant constraints, we obtain the following modified RLT formulation of SPP.

$$
\begin{aligned}
&\sum_{j \in N_i} x_j = 1 \quad \forall \ i \in M \\
&[(\sum_{j \in N_{i,k} \cap F} w_{(jk)} + \sum_{j \in N_{i,k} \cap \bar{F}} v_{jk} = x_k \ \forall \ k \in S_t) \ , \ \forall \ i \in \overline{M}_k \ \ni \ S_t \cap N_i \neq \phi \ ], \ \forall \ t = 1,\ldots,p \\
&w_{(jk)} \geq 0 \ \text{ for } j \in \tilde{N}_k \cap F, \ \ k \in S_t, \ \ t = 1,\ldots,p \\
&v_{jk} \leq x_j \ \text{ for } j \in \tilde{N}_k \cap \bar{F}, \ \ k \in S_t, \ \ t = 1,\ldots,p \qquad\qquad (2.67) \\
&v_{jk} \geq 0 \ \text{ for } j \in \tilde{N}_k \cap \bar{F}, \ \ k \in S_t, \ \ t = 1,\ldots,p \\
&x_j \geq 0, \ \ \forall j \in N
\end{aligned}
$$

where $\tilde{N}_k = \bigcup_{i \in \overline{M}_k \ \ni \ N_{i,k} \cap S_t \neq \phi} N_{i,k} \ \ \forall k \in S_t, \ t = 1,\ldots,p.$

Proposition 2.13 If $\bar{x} = (\bar{x}_F, \bar{x}_{\bar{F}})$ is an optimal basic feasible solution to $\overline{SP}$ with $F \neq \phi$, then SPP_{RLT} cuts off $\bar{x}$.

Proof. Suppose on the contrary that $\bar{x}$ remains feasible to SPP_{RLT}. By Proposition 2.12, $\overline{w}_{(ij)} = 0\ \forall i \in F$ and $\bar{v}_{ij} = 0\ \forall i \in \bar{F}$ if $\bar{x}_j = 0$, and $\overline{w}_{(ij)} = \bar{x}_i\ \forall i \in F$, and $v_{ij} = \bar{x}_i\ \forall i \in \bar{F}$. if $\bar{x}_j = 1\ \ \forall\ j \in \bar{F}$, after fixing $x_j = \bar{x}_j\ \forall j \in \bar{F}$, this becomes equivalent to applying the RLT formulation to the reduced system (2.66) itself.

Now consider the system (2.66) and let us examine the separable block of constraints for any $t \in \{1,\ldots,p\}$. Denote $x_{F_t} = \{x_j : j \in S_t\}$. Since $\bar{x}$ is a basic feasible solution, this system is of the form $B_t\bar{x}_{F_t} = e$ where $e = (1,\ldots,1)^t$, and B_t is nonsingular. Letting $W_i = (w_{(ij)}, j \in S_t)$, $\forall i \in S_t$, we have by RLT that $B_t\overline{W}_i = e\,\bar{x}_i$ for all $i \in S_t$. Hence, $\overline{W}_i = B_t^{-1}e\bar{x}_i$ for all $i \in S_t$.

But for each $i \in S_t$, $\overline{W}_{ii} = \bar{x}_i > 0$, implies that the corresponding component of $B_t^{-1}e$ is equal to one. Hence, $B_t^{-1}e = e$, a contradiction to the fact that $\bar{x}_{F_t} = B_t^{-1}e$ is fractional. This completes the proof. ■

Corollary 2.14 If $\bar{x}$ is a unique optimal fractional basic feasible solution to $\overline{SPP}$, then $\nu(SPP_{RLT}) > \nu(\overline{SPP})$, and so SPP_{RLT} generates a tighter lower bound.

Proposition 2.13 shows that the modified RLT is a strong formulation that cuts off the obtained fractional optimal basic feasible solution to $\overline{SPP}$, but the size of SPP_{RLT} is greatly reduced as compared with that of SPP_1. Of course, we can explicitly obtain a family of cutting planes through the corresponding separation problem, $SPP_{projection}$, as before. In this manner, we can generate a strong valid inequality, which cuts off the fractional optimal basic feasible solutions to $\overline{\text{SPP}}$.

2.5. Summary and Recommendations for Further Research

In this chapter, we have developed a hierarchy of relaxations for the set partitioning polytope, ranging from the linear programming relaxation to the convex hull representation. This hierarchy of relaxations can be used to obtain a sequence of tight lower bounds. That is, for a given problem SP, letting $v(\text{SP}_\delta) = \min \{ cx : x \in \text{SPP}_{P\delta} \}$, where $\text{SPP}_{P\delta}$ is a projection of the δth-order relaxation SPP_δ onto the original x space, the lower bound $v(\text{SP}_\delta)$ approaches the optimal objective value of the problem SP when δ approaches n. Moreover, for a fixed δ, $v(\text{SP}_\delta)$ can (in principle) be computed in polynomial time, as the optimum of some linear programming problem. However, the computational effort of evaluating $v(\text{SP}_\delta)$ grows exponentially with δ. Furthermore, we have developed a new family of cutting planes for the set partitioning polytope which delete any fractional basic feasible solutions to its underlying linear programming relaxation. We have also shown that all the known valid inequalities belong to this family of cutting planes, and hence, this provides a unifying framework for a broad class of such inequalities. Hence, a natural extension of this research is leading to the development of an effective computational procedure to incorporate these cutting planes in the context of a branch-and-cut algorithm.

3. A New Class of Lifted Facets for the Boolean Quadric Polytope

3.1. Introduction

We consider the unconstrained pseudo-boolean quadratic zero-one problem in n variables

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

where all the data is rational and $N = \{1,..,n\}$.

Problem QP can be used for the analysis of various combinatorial optimization problems. As shown in Hammer (1979) and Hansen (1979), a large number of combinatorial optimization problems can be reformulated as problem QP such as vertex packing, maximum cut, quadratic assignment, and set partitioning problems, to name a few. It is also well known that the maximization of any pseudo-boolean polynomial function can be reduced to a quadratic case. Recent research on problem QP includes the work of Adams and Dearing (1988), Adams et al. (1991), Barahona et al. (1989), Billionnet and Jaumard (1989), Boros et al. (1989), and Hammer et al. (1984), as well as that of Sherali and Adams (1986) for the constrained zero-one quadratic programming problem, and of Balas and Mazzola (1984.a, 1984.b) for general nonlinear zero-one programming problems. However, we are mainly concerned in this chapter with the polyhedral structure of the convex hull of feasible solutions to problem QP.

Toward this end, we consider a linearization of problem QP. (See Hansen, 1979, for a survey on linearization techniques.) We linearize the quadratic terms x_ix_j by introducing new variables w_{ij} that are zero-one valued and satisfy the following well-known set of constraints.

$$w_{ij} - x_i \leq 0 \;\; \forall\; 1 \leq i < j \leq n \tag{3.1}$$

$$w_{ij} - x_j \leq 0 \;\; \forall\; 1 \leq i < j \leq n \tag{3.2}$$

$$-w_{ij} + x_i + x_j \leq 1 \;\; \forall\; 1 \leq i < j \leq n \tag{3.3}$$

$$x_i \in (0, 1), \;\; \forall i \in N, \;\; w_{ij} \geq 0, \;\; \forall\; 1 \leq i < j \leq n. \tag{3.4}$$

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

$$\text{Maximize } \{\sum_{j \in N} q_j x_j + \sum_{1 \leq i < j \leq n} q_{ij} w_{ij} : \text{(x, w) satisfies (3.1), (3.2), (3.3), and (3.4)}\}.$$

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

$$QP^n = \text{conv}\{ (x, w) \in R^{n_0} : \text{(x, w) satisfies (3.1), (3.2), (3.3), and (3.4)} \}$$

as the *boolean quadric polytope*, where $n_0 = n(n + 1)/2$.

Recent research on the polyhedral structure of QP^n includes that of Padberg (1989) and Boros et al.(1990). More importantly, as observed by Simone (1989), results for the polyhedral structure of QP^n can be directly translated into those of the cut polytope defined by Barahona and Mahjoub (1986), and vice versa. For the case of the cut polytope (cut cone), there is an extensive literature on several classes of valid inequalities and facets. (See Deza and Laurent, 1988, 1989a, 1989b, Deza et al., 1989, Simone, 1989, 1990, and Simone et al. 1989.) However, motivated by Padberg's results on the boolean quadric polytope, we develop a new class of lifted facets for the boolean quadric polytope.

The following is an outline of this chapter. In Section 2, we implement the reformulation-linearization-technique (RLT) of Sherali and Adams (1990) in order to characterize the polyhedral structure of QP^n. This structure plays a key role in obtaining a class of valid inequalities and facets. In Section 3, we develop a class of valid inequalities and facets, denoted by a product-form inequality, and show that the known classes of Padberg's (1989) facets for QP^n, the clique inequalities, the cut inequalities, and the generalized cut inequalities, all belong to this product-form class of valid inequalities. In Section 4, we present a new explicit class of valid inequalities and facets obtained through the simultaneous lifting procedure applied to this foregoing class of inequalities. Section 5 concludes this chapter with some recommendations for further research.

3.2. Polyhedral Structure of the Boolean Quadric Polytope

In this section, we consider an implementation of the reformulation-linearization-technique (RLT) of Sherali and Adams (1990) for characterizing the polyhedral structure of the quadric polytope. Toward this end, consider the n-th order nonlinear reformulation of the constraints (3.1), (3.2), (3.3), and (3.4).

$$NQP(n) = \{x \in R^n : F_n(J, \bar{J}) \geq 0 \ \forall J \subseteq N, \ \bar{J} \equiv N - J\} \tag{3.5}$$

where $F_n(J, \bar{J}) = \prod_{t\in J} x_t \prod_{s\in \bar{J}}(1 - x_s)$.

In order to explicitly represent the quadratic terms, let us partition the constraints in (3.5) into the sets corresponding to $\{J = \phi, |J| = 1, |J| = 2\}$, and the remaining constraints written for $|J| = 3,\ldots,n$, stated in this order below.

$$1 - \sum_{j\in N} x_j + \sum_{1\le i<j\le n} x_i x_j \ge \sum_{r=3}^{n} \sum_{|J|=r}^{n} \frac{(r-1)(r-2)}{2} \prod_{t\in J} x_t \prod_{s\in \bar{J}} (1 - x_s)$$

$$x_j - \sum_{1\le i<j\le n} x_i x_j \ge \sum_{r=3}^{n} \sum_{|J|=r \ni j\in J}^{n} (2 - r) \prod_{t\in J} x_t \prod_{s\in \bar{J}} (1 - x_s) \quad \text{for } j = 1,\ldots, n$$

$$x_i x_j \ge \sum_{r=3}^{n} \sum_{|J|=r \ni j\in J}^{n} \prod_{t\in J} x_t \prod_{s\in \bar{J}} (1 - x_s) \quad \text{for } 1 \le i < j \le n$$

$$\prod_{t\in J} x_t \prod_{s\in \bar{J}} (1 - x_s) \ge 0 \quad \text{for } J \subseteq N, |J| = 3,\ldots, n.$$

Next, let us consider a linearization of the n-th order polynomial reformulation NQP(n). Let $f_n(J, \bar{J})$ be a linearized expression of a nonnegative polynomial factor $F_n(J, \bar{J}) = \prod_{t\in J} x_t \prod_{s\in \bar{J}}(1 - x_s)$, in which each polynomial term of the type $\prod_{t\in J} x_t$ has been replaced by a single variable w_J, $\forall J \subseteq N$. For example, a polynomial term $x_1x_2x_3$ is replaced by w_{123}.

Now, let us use the transformation $y_J = f_n(J, \bar{J})$, $\forall J \subseteq N$, $|J| = 3,\dots,n$, whose inverse is given by $w_J = \sum_{J' \subseteq \bar{J}} y_{J \cup J'}$, $\forall J \subseteq N$, $|J| = 3,\dots,n$. Writing the linearized set of constraints for NQP(n), and substituting out w_J, for $3 \le |J| \le n$, in terms of y_J, we obtain an equivalent linearized reformulation of (3.5), given by QP(n) $= \{(x,w) \in R^{n_0} : (x,w,y) \in Z^n\}$, where the constraints of Z^n are as follows.

$$1 - \sum_{j \in N} x_j + \sum_{1 \le i < j \le n} w_{ij} \ge \sum_{r=3}^{n} \sum_{|J|=r} \frac{(r-1)(r-2)}{2} y_J \qquad (3.6)$$

$$x_j - \sum_{1 \le i < j \le n} w_{ij} \ge \sum_{r=3}^{n} \sum_{|J|=r \ni j \in J}^{n} (2-r) y_J \quad \text{for } j = 1,\dots, n \qquad (3.7)$$

$$w_{ij} \ge \sum_{r=3}^{n} \sum_{|J|=r \ni j \in J}^{n} y_J \quad \text{for } 1 \le i < j \le n \qquad (3.8)$$

$$y_J \ge 0 \quad \text{for } J \subseteq N,\ |J| = 3,\dots, n.$$

Let π_0, π_j, and π_{ij} represent dual multipliers associated with constraint (3.6), (3.7), and (3.8), respectively. Since QP^n is a full dimensional polytope, using the projection operation (see, for example, Nemhauser and Wolsey, 1988 or Balas, 1992), we have

$$QP^n = \{(x, w) \in R^{n_0} : \sum_{j \in N} (\pi_0^k - \pi_j^k) x_j + \sum_{1 \le i < j \le n} (\pi_i^k + \pi_j^k - \pi_{ij}^k - \pi_0^k) w_{ij} \le \pi_0^k,\ \forall k \in K\} \qquad (3.9)$$

where π^k, $k \in K$, are extreme directions of the following polyhedral cone Π:

$$\Pi \equiv \{\pi \geq 0 : \frac{-(r-1)(r-2)}{2}\pi_0 + (r-2)\sum_{j\in J}\pi_j - \sum_{i<j\in J}\pi_{ij} \leq 0 \quad \forall\, J\subseteq N \ni |J| = r = 3,\ldots,n\},$$

where for $J\subseteq N$, $(i<j\in J) \equiv \{(i,j): i<j,\ i\in J, j\in J\}$.

For the simplicity of analysis, consider the following transformation:

$$\lambda_0 = \pi_0,\ \lambda_j = \pi_0 - \pi_j,\ \lambda_{ij} = \pi_i + \pi_j - \pi_{ij} - \pi_0,$$

with inverse $\pi_0 = \lambda_0,\ \pi_j = \lambda_0 - \lambda_j,\ \pi_{ij} = \lambda_0 - (\lambda_i + \lambda_j + \lambda_{ij})$.

Using the above transformation, we can restate (3.9) as follows.

$$QP^n = \{(x, w) \in R^{n_0} : \sum_{j\in N}\lambda_j^k x_j + \sum_{i<j\in N}\lambda_{ij}^k w_{ij} \leq \lambda_0^k,\ \forall k\in K\} \tag{3.10}$$

where λ^k, $k\in K$, are extreme directions of the polyhedral cone Λ, given by

$$\Lambda \equiv \{\lambda : \sum_{j\in J}\lambda_j + \sum_{i<j\in J}\lambda_{ij} \leq \lambda_0,\ \forall\, J\subseteq N\}. \tag{3.11}$$

Since QP^n is full dimensional, we have the following result due to Sherali and Adams (1990) concerning the polyhedral structure of QP^n.

Proposition 3.1 The inequality $\sum_{j\in N}\lambda_j x_j + \sum_{1\leq i<j\leq n}\lambda_{ij}w_{ij} \leq \lambda_0$ is valid for QP^n if and only if $\lambda \equiv (\lambda_j, \lambda_{ij}, \lambda_0)$ belongs to Λ, where Λ is a polyhedral cone given by (3.11). Furthermore, this valid inequality is facet defining for QP^n if and only if $\lambda \equiv (\lambda_j, \lambda_{ij}, \lambda_0)$ is an extreme direction of the polyhedral cone Λ. ■

Remark 3.1 Given $\lambda \equiv (\lambda_j, \lambda_{ij}, \lambda_0) \in \Lambda$ that defines a facet of QP^n, let the *support graph* $G(\lambda)$ have nodes $N(\lambda)$ corresponding to indices of nonzero coefficients λ_j, and have edges $A(\lambda)$ corresponding to indices of nonzero λ_{ij} coefficients. By Padberg (1989, Lemma 1), we know that $G(\lambda)$ is connected. More importantly, by Theorem 3 (Lifting Theorem) of Padberg (1989), any facet of $QP^{n'}$ $(n' < n)$ is a facet for QP^n, and conversely, if $|N(\lambda)| = n' < n$, then the given facet of QP^n is a facet for $QP^{n'}$. Hence, inductively, it is sufficient to study facets with $|N(\lambda)| = n$, i.e., facets having all the indices appearing among its nonzero coefficients. Also note that all the facets of QP^n for $n \leq 3$ are trivially identified (Padberg, 1989). We henceforth assume that $n \geq 4$.

3.3. A Product-Form Facet for the Boolean Quadric Polytope

In this section, we consider a class of facets for QP^n derived via nonpositive pseudo-boolean functions. A function of the form

$$f(x) = \sum_{j \in N} a_j x_j + \sum_{1 \leq i < j \leq n} b_{ij} x_i x_j + C$$

is called a *nonpositive quadratic pseudo-boolean function (NQPBF)* if maximum $\{f(x) : x \in (0, 1)^n\} = 0$, where a_j and b_{ij} are rational for all i,j and C is a constant. For any given NQPBF f, after linearizing the quadratic term $x_i x_j$ in the function f, let L[f] denote the following inequality

$$\sum_{j \in N} a_j x_j + \sum_{1 \leq i < j \leq n} b_{ij} w_{ij} + C \leq 0. \tag{3.12}$$

Clearly, the inequality (3.12) corresponding to a NQPBF is valid for QP^n. However, by Proposition 3.1, in order to show that the inequality (3.12) is facet defining, we need to prove that the coefficients of the inequality (3.12) correspond to an extreme direction of the pointed cone Λ, defined in (3.11). Although the computational burden of this task grows exponentially with n, there exist known classes of facets for specific nonpositive NQPBF's. We describe below a particular class of NQPBF's which will be used for deriving various classes of facet defining inequalities for QP^n.

Motivated by Remark 3.1, let $J_1 \subseteq N$ and $J_2 \equiv N - J_1$, and let $\alpha \in \{1, \ldots, n-2\}$, and consider the following quadratic function, where $\bar{x}_j \equiv (1 - x_j)\ \forall j \in N$.

$$f(x) = -\left(\sum_{j \in J_1} x_j + \sum_{j \in J_2} \bar{x}_j - \alpha\right)\left(\sum_{j \in J_1} x_j + \sum_{j \in J_2} \bar{x}_j - \alpha - 1\right) \qquad (3.13)$$

Then, the following proposition can be readily established.

Proposition 3.2 The quadratic function f(x) defined in (3.13) is a NQPBF.

We now consider the linearized inequality L[f] corresponding to the NQPBF (3.13), denoted by the *(nonhomogeneous) product-form inequality,*

$$(\alpha - t)\sum_{j \in J_1} x_j + (t - \alpha - 1)\sum_{j \in J_2} x_j - \sum_{i < j \in J_1} w_{ij} - \sum_{i < j \in J_2} w_{ij} + \sum_{(i,j) \in (J_1 : J_2)} w_{ij} \leq \frac{(t - \alpha)(t - \alpha - 1)}{2} \qquad (3.14)$$

where for any $J_1 \subseteq N$ and $J_2 \equiv N - J_1$, let $t = |J_2|$ and $(J_1 : J_2) \equiv \{(i,j) : i \in J_1,\ j \in J_2\}$. Note that the product-form inequality is valid for QP^n.

Remark 3.2 It is also known (Theorem 6 (Symmetry Theorem), Padberg, 1989) that the vertex figure of any vertex $(\overline{x}, \overline{w})$ of QP^n is the same as that of any other vertex, say, the origin, i.e., the facets binding at any vertex $(\overline{x}, \overline{w})$ are obtainable from those binding at the origin through a simple transformation. Hence, it is sufficient to characterize only the facets given by (3.10) that are binding at the origin, and as per Remark 3.1, have $|N(\lambda)| = n$.

From Remark 3.2, we are henceforth interested in finding facets of the type,

$$\sum_{j \in N} \lambda_j x_j + \sum_{1 \leq i < j \leq n} \lambda_{ij} w_{ij} \leq 0 \tag{3.15}$$

where λ is an extreme direction of Λ_0 given below, having all the indices in N accounted among the nonzero coefficients in (3.15).

$$\Lambda_0 \equiv \{\lambda : \sum_{j \in J} \lambda_j + \sum_{i < j \in J} \lambda_{ij} \leq 0 \quad \forall J \subseteq N\}. \tag{3.16}$$

Furthermore, according to Remark 3.2, consider the product-form inequalities that are binding at the origin and have $|N(\lambda)| = n$. These are *homogeneous product-form inequalities*, having $t = \alpha$ or $t = \alpha + 1$ in (3.14), and can be characterized as follows.

$$-\sum_{j \in J} x_j - \sum_{i < j \in J} w_{ij} - \sum_{i < j \in \bar{J}} w_{ij} + \sum_{(i,j) \in (J:\bar{J})} w_{ij} \leq 0 \quad \forall J \subseteq N, |J| = 1, \ldots, n-2, \bar{J} \equiv N - J \tag{3.17}$$

Proposition 3.3 For any $J_1 \subseteq N$ and $J_2 \equiv N - J_1$ with integer α, $1 \leq \alpha \leq n-2$, and $t = |J_2|$, the product-form inequality of type (3.14) is a facet-defining inequality for QP^n, denoted by a *product-form facet.* The number of distinct product-form facets equals $(n-3)3^{n-1} + 2n^2 - n + 1$.

Proof. (i) The first assertion can be shown to be a special case of the known class of facets by Boros and Hammer (1990) and Deza and Laurent (1988).

(ii) To count the number of such facets, let $\bar{f}_k$ be the number of product-form facets of QP^n which have some k particular indices represented among the nonzero coefficients, and let f_n be the total number of product-form facets QP^n. Then, we have

$$f_n = \binom{n}{2}\bar{f}_2 + \binom{n}{3}\bar{f}_3 + \ldots + \binom{n}{n}\bar{f}_n. \tag{3.18}$$

In order to avoid duplications in counting $\bar{f}_k$ for any $k \geq 3$, note that by interchanging J_1 and J_2, and replacing α by α' in the product-form facet, we will have duplicates whenever $\alpha - t = (k-t) - \alpha' - 1$, or $\alpha + \alpha' = k - 1$. Hence, when $k(\neq 2)$ is even, since we cannot then have $\alpha = \alpha'$ in this duplication, and $\alpha \in \{1, \ldots, k-2\}$, the number of distinct α values that need to be considered is (k-2)/2 for any partition (J_1, J_2) of the k indices. Similarly, when k is odd, we need $\alpha = 1, \ldots, (k-1)/2 - 1$, and when $\alpha = \alpha' = (k-1)/2$, we obtain duplicates. Hence, in either case, it is sufficient to consider $\alpha = 1, \ldots, \lfloor (k-1)/2 \rfloor$ for $k \geq 3$, but when k is odd, this will create one duplicate for any partition (J_1, J_2) upon interchanging J_1 and J_2. Moreover, noting that there are 2^k ways of composing partition J_1 and J_2 in the product-form facet for each α, we get

$$\bar{f}_k = \left(\frac{k}{2} - 1\right)2^k. \tag{3.19}$$

From (3.18) and (3.19) observing that $\bar{f}_2 = 4$, we obtain $f_n = \sum_{k=3}^{n}\binom{n}{k}2^k(k/2-1) + \binom{n}{2}4$ $= \sum_{k=0}^{n}\binom{n}{k}k2^{k-1} - \sum_{k=0}^{n}\binom{n}{k}2^k + (2n^2-n+1) = (n-3)3^{n-1} + 2n^2 - n + 1$, and this completes the proof. ■

We now recall three classes of facets from Padberg (1989). For any $S \subseteq N$ with $|S| \geq 2$ and integer β, $1 \leq \beta \leq |S|-1$, the *clique facet* is defined as

$$\sum_{j\in S}\beta x_j - \sum_{i<j\in S} w_{ij} \leq \frac{\beta(\beta+1)}{2}.$$

For any $S \subseteq N$ with $|S| \geq 1$ and $T \subseteq N - S$ with $|T| \geq 2$, the *cut facet* is defined as

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

For any $S \subseteq N$ with $s = |S| \geq 1$ and $T \subseteq V - S$ with $t = |T| \geq 2$, the *generalized cut facet* is given by

$$(s-t)\sum_{j\in S}x_j + (t-s-1)\sum_{j\in T}x_j - \sum_{i<j\in S} w_{ij} - \sum_{i<j\in T} w_{ij} + \sum_{(i,j)\in(S:T)} w_{ij} \leq \frac{(t-s)(t-s-1)}{2}.$$

Then, by Proposition 3.3, the following proposition is readily established.

Proposition 3.4 The clique facet for QP^n belongs to a class of the product-form facets of type (3.14) with $J_1 = \phi$, $J_2 = S$, $\alpha = |S| - \beta - 1$, where $\beta \in \{1, \ldots, |S|-2\}$, $S \subseteq N$, $|S| \geq 3$. The cut facet for QP^n belongs to the class of the product-form facets with $\{J_1 = S, J_2 = T, \alpha = |S| - 1\}$, or $\{J_1 = T, J_2 = S, \alpha = |S|\}$, where $S \subseteq N$, $|S| \geq 1$, $T \subseteq N - S$, $|T| \geq 2$.

The generalized cut facet for QP^n belongs to the class of the product-form facets with $J_1 = S$, $J_2 = T$, $\alpha = |S|$, where $S \subseteq N$, $|S| \geq 1$, $T \subseteq N - S$, $|T| \geq 2$. ■

3.4. Lifting of the Product-Form Facet

Consider a facet of the type (3.15) for the case of (n-1) variables x_j, $j \in N' \equiv \{1,\ldots,n-1\}$, of the form

$$\sum_{j \in N'} \bar{\lambda}_j x_j + \sum_{i < j \in N'} \bar{\lambda}_{ij} w_{ij} \leq 0. \tag{3.20}$$

By Proposition 3.1, note that $\bar{\lambda}(N') \equiv (\bar{\lambda}_j,\ j \in N',\ \bar{\lambda}_{ij},$ for $i < j \in N')$ is an extreme direction of the following polyhedral cone:

$$\{\lambda : \sum_{j \in J} \lambda_j + \sum_{i < j \in J} \lambda_{ij} \leq 0 \quad \forall J \subseteq N'\}. \tag{3.21}$$

Now, let us consider a lifting of the inequality (3.20) into a facet in N-space.

Proposition 3.5 Suppose that the inequality given by (3.20) is a facet for QP^{n-1}. Then, letting $N' \equiv \{1,\ldots,n-1\}$, the inequality

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

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

$$\{\lambda : \lambda_n + \sum_{j\in J} \lambda_{jn} \leq - \sum_{j\in J} \bar{\lambda}_j - \sum_{i<j\in J} \bar{\lambda}_{ij} \quad \forall J \subseteq N'\}. \tag{3.22}$$

Proof. Let us partition the constraints of Λ_0 as follows.

$$\text{N}' - \text{Block:} \quad \sum_{j\in J} \lambda_j + \sum_{I<j\in J} \lambda_{ij} \leq 0 \quad \forall J \subseteq N' \tag{3.23}$$

$$\text{n} - \text{Block:} \quad \lambda_n + \sum_{j\in J} \lambda_{jn} \leq - \sum_{j\in J} \lambda_j - \sum_{I<j\in J} \lambda_{ij} \quad \forall J \subseteq N' \tag{3.24}$$

From (3.20) and Proposition 3.1, we have that there are some n(n-1)/2 - 1 linearly independent hyperplanes binding from (3.21), and so from (3.23), at the solution $\bar{\lambda}(N')$. 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 (3.22), then it is uniquely determined by some n linearly independent binding inequalities from (3.22). Examining (3.22) and (3.24), 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 and has $[n(n-1)/2-1] + n = n(n+1)/2-1$ has linearly independent defining hyperplanes from (3.23) and (3.24) binding at this solution. Hence, $\bar{\lambda}(N)$ is an extreme direction of Λ_0, and so from Proposition 3.1, is a facet of QP^n. ■

Remark 3.3 In the light of Proposition 3.5, we need to characterize vertices of (3.24) with right-hand sides having values given by known facets for the (n-1) variable case. Note that this is a "simultaneous" lifting process, since the coefficients $\lambda(n) \equiv$

(λ_n, λ_{jn} for $j\in N'$) are being simultaneously determined, as opposed to being sequentially determined.

Consider the homogenous product-form facet of the form (3.17) in N'-space defined for some $J_1 \subseteq N'$, $|J_1| \in \{1,\dots, n-3\}$, $\bar{J}_1 = N' - J_1$.

$$-\sum_{j\in J_1} x_j - \sum_{i<j\in J_1} w_{ij} - \sum_{i<j\in \bar{J}_1} w_{ij} + \sum_{(i,j)\in(J_1:\bar{J}_1)} w_{ij} \le 0 \tag{3.25}$$

Let $\bar{\lambda}(N')$ denote that the coefficients of (3.25). Accordingly, denote the right-hand-side of (3.24) by b_J, for any $J\subseteq N'$, when $\lambda(N') \equiv \bar{\lambda}(N')$. Then, given any $J\subseteq N'$, $|J \cap J_1| = p$, $|J \cap \bar{J}_1| = q$, we have that

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

This gives the n-Block constraints (3.24) as

$$\lambda_n + \sum_{j\in J} \lambda_{jn} \le \frac{(p-q)(p-q+1)}{2} \quad \forall J\subseteq N', \text{ where } |J\cap J_1| = p,\ |J\cap \bar{J}_1| = q. \tag{3.26}$$

Proposition 3.6 The following are three vertices of (3.26):

(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 N′ −space facet itself, which is a facet for the N-space by Remark 3.1, (ii) corresponds to $J = J_1 \bigcup \{n\}$ and $\overline{J} = \overline{J}_1$, in (3.17), and (iii) corresponds to $J = J_1$ and $\overline{J} = \overline{J}_1 \bigcup \{n\}$ in (3.17).

Proof. First of all, note that the given values of (λ_n, λ_{jn} for j∈N′) in (i), (ii), and (iii), in combination with (3.25), yield inequalities of type (3.17), and so by Remark 3.1, they correspond to facets of QP^n. Consequently, by Proposition 3.5, the given three sets of values of (λ_n, λ_{jn} for j∈N′) must be vertices of (3.26). This completes the proof. ■.

Remark 3.4 The question we can raise here is whether Proposition 3.6 defines all the vertices of (3.26). By examining the resulting facets obtained for $n \leq 4$, with the known ones for this case as in Deza and Laurent (1988), this is true for $n \leq 4$. However, as the following example shows, for n=5, this is false. Indeed, for $n \geq 5$, not all facets are of the type (3.17) (see Deza and Laurent, 1988) and in fact, as we show below, vertices of (3.26) themselves can produce facets different from this class.

Example 3.1 Let n=5, $J_1 = \{1,2\}$, $\overline{J}_1 = \{3, 4\}$, $\overline{\lambda}_5 = -1$, $\overline{\lambda}_{j5} = 2$ $\forall j \in J_1$, $\overline{\lambda}_{j5} = -2$ $\forall j \in \overline{J}_1$. Let us verify that this $\overline{\lambda}(n)$ vector is a vertex of (3.26). To show feasibility, we must have $-1 + 2p-2q \leq (p-q)(p-q+1)/2$, i.e., $(p - q)^2 - 3(p - q) + 2 \geq 0$ p=0,1,2, q =0,1,2 . This expression is convex in (p-q) and is minimized when $(p - q) = 3/2$. Moreover, it is zero when (p-q)=1 or 2, and so by convexity, it is nonnegative for all integer values of (p-q). Moreover, the following five constraints from (3.26) are binding at $\overline{\lambda}(n)$, and so yield $\overline{\lambda}(n)$ as the unique solution.

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

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

Consequently, $\bar{\lambda}(n)$ is a vertex of (3.26). 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 (3.17), and it has been obtained by lifting a facet of the type (3.17) from the N′-space. In fact, as we will show, the above inequality is a member of an entire class of facets obtainable in this fashion. ■

Consider (3.26), and suppose that we restrict

$$\lambda_{jn} = \theta \ \ \forall j \in J_1 \ \text{ and } \lambda_{jn} = -\theta \ \ \forall j \in \bar{J}_1. \tag{3.27}$$

Then, system (3.26) becomes

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

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

$$\lambda_n + \theta r \leq \frac{r(r+1)}{2} \qquad \forall r = P + 1 - n,\dots, P. \tag{3.28}$$

Proposition 3.7 The vertices (λ_n, θ) of the two dimensional polytope (3.28) are all of the type

$$\lambda_n = \frac{-R(R-1)}{2} \ \text{ and } \theta = R, \ \text{ for } R = P + 2 - n,\dots, P. \tag{3.29}$$

Proof. Consider any defining hyperplane in (3.28) for $r = R$, say. Setting this as an equality gives $\lambda_n = R(R+1)/2 - \theta R$. The end points of the interval on this plane that are feasible to (3.28) will give vertices of (3.28) that lie on this plane. Repeating for all planes, will give all the vertices of (3.28). Toward this end, note that points in the above plane are feasible provided

$$R\frac{(R+1)}{2} - R\theta + r\theta \leq \frac{r(r+1)}{2}, \text{ i.e., } \theta(R-r) \geq \frac{(R-r)(R+r+1)}{2} \quad \forall r = P+1-n, \ldots, P.$$

In turn, this is equivalent to the following.

$$\theta \geq \frac{R+r+1}{2} \quad \forall r = P+1-n,\ldots, R-1 \text{ and } \theta \leq \frac{R+r+1}{2} \quad \forall r = R+1,\ldots,P$$

This implies that

$$\theta \geq R \quad \text{if } R \geq P+2-n \quad \text{and} \quad \theta \leq (R+1) \quad \text{if } R \leq P-1.$$

Hence, tracing vertices as R varies from $P+1\text{-}n$ to P, we get vertices of (3.28) as $\theta = R$ for $R = P+2\text{-}n,\ldots,P$, and $\lambda_n = R(R+1)/2 - R^2 = -R(R-1)/2$. This completes the proof. ■

The following proposition show that the vertices of (3.29) also correspond to the vertices of (3.26) via (3.27).

Proposition 3.8 The vertex $\bar{\lambda}(n) \equiv \{\bar{\lambda}_n = -R(R-1)/2, \bar{\lambda}_{jn} = R \ \forall j \in J_1, \bar{\lambda}_{jn} = -R \ \forall j \in \bar{J}_1\}$, for each $R = P+2\text{-}n,\ldots, P$, where $P = |J_1|$, defines a vertex of (3.26).

Proof. Given any $R \in \{P+2-n,\ldots P\}$, since $\theta = R$, and $\lambda_n = -R(R-1)/2$ are feasible to (3.28) by Proposition 3.7, we have from (3.27) that

$$\lambda_n = \frac{-R(R-1)}{2}, \quad \lambda_{jn} = R \ \forall j \in J_1, \quad \lambda_{jn} = -R \quad \forall j \in \bar{J}_1, \tag{3.30}$$

is feasible to (3.26) for all $R = P + 2\text{-}n, \ldots, P$, where $|J_1| \equiv P$. We need to show that this is a vertex of (3.26). For this solution (3.30), the left hand side of any constraint in (3.26) is given by $-R(R-1)/2 + R(p-q)$ and the right hand side is $(p\text{-}q)(p\text{-}q+1)/2$. Hence, the constraint is binding whenever these are equal, i.e., whenever $(p\text{-}q\text{-}R)(p\text{-}q\text{-}R+1) = 0$, i.e., $(p\text{-}q) = R$ or $(p\text{-}q) = R\text{-}1$.

To complete the proof, it is sufficient to show that the set of binding constraints yield (3.30) as the unique solution. Toward this end, let $s \in J_1$ and $t \in J_2$ 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 means that $\lambda_{jn} = 0$, $\forall j \in J_1$, and $\lambda_{jn} = -\theta$ $\forall j \in \bar{J}_1$ is implied by this system. But this reduces the system (3.26) to the system (3.28). Examining the constraints in (3.28) for $r = (p\text{-}q) = R$ and $r = (p\text{-}q) = R\text{-}1$, which are binding at the given solution (3.30) as noted above, we get $\lambda_n + \theta R = R(R+1)/2$ and $\lambda_n + \theta(R-1) = R(R-1)/2$. These equation uniquely yield $\lambda_n = -R(R-1)/2$ and $\theta = R$, and so this implies that (3.30) is a vertex of (3.26).

Hence, to show that $\lambda_{sn} = -\lambda_{tn}$, consider a set $J \subseteq N'$ such that $|J \cap J_1| = p$, $|J \cap \bar{J}_1| = q$, $\{s, t\} \cap J = \phi$, and $(p\text{-}q) = (R\text{-}1)$ if $R = P$, and $(p\text{-}q) = R$ 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 $(p\text{-}q) = R$.) Since the constraint for this 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.

Proposition 3.9 The following inequalities define a class of facets for QP^n:

$$-\sum_{j\in J_1} x_j - \sum_{i<j\in J_1} w_{ij} - \sum_{i<j\in \bar{J}_1} w_{ij} - \sum_{(i,\,j)\in(J_1:J_2)} w_{ij} - \frac{R(R-1)}{2} x_n + R\sum_{j\in J_1} w_{jn} - R\sum_{j\in \bar{J}_1} w_{jn} \le 0$$

where $J_1 \subseteq N'$, $|J_1| = P \ge 1$, $\bar{J}_1 = N' - J_1$, with $|\bar{J}_1| \ge 2$, and $R \in \{P + 2 - n, \ldots, P\}$.

Proof. Follows from Proposition 3.7 and 3.8. ■

3.5. Summary and Recommendations for Further Research

We have developed a class of facets obtained by lifting homogenous product-form facets of QP^n. Motivated by this result, we present some further research tasks:

1. Study the characterization of vertices of (3.26) for finding other classes of facets that can be generated by lifting the product-form facet (3.25).

2. Study the system of type (3.26) that is formulated via the problem of lifting the generalized product-form facet corresponding to a general NQPBF as specified by Boros and Hammer (1990).

Models and Algorithms for Passive Homing Device Operations in Battle-Group Anti-Air Warfare Scenarios

4. Problem Description

Consider the scenario of a naval battle-group defending itself against an air attack, being equipped with surface-to-air missiles (SAM's), sensors, and target illuminators (passive homing devices). The sensors track the approaching targets (airplanes, cruise missiles), and provide in-flight guidance for the SAM's. Each target must be illuminated during some last few seconds of the SAM trajectory in order for it to "home in" on the target. Because of the common theater that these battle-group ships operate within, this expands the battle space, and poses a cooperative engagement problem that calls for a coordinated use of anti-air defense resources among the ships. Also, the number of illuminators which the battle-group jointly possesses is very small compared to the number of SAM's, and compared to the number of potential incoming targets. In this resource management problem, the task of allocating the targets to the illuminators emerges as a critical problem. Additionally, each target must be illuminated during a given engageability duration that is specific to the location of each illuminator, and the illuminators may have certain specified unavailability durations due to previously scheduled commitments. The cooperative battle-group anti-air defense problem can then be stated as follows:

Given a collection of targets approaching on predicted trajectories, and given the engageability durations and the illumination times which are specific to each target and to the illuminators on each individual ship, allocate the targets among the illuminators and schedule them for illumination so as to minimize some appropriate merit or reward function, while satisfying the illuminator availability constraints, as well as the target engageability interval constraints. (See Boyer et al., 1990, for related command and control decision issues.)

To formulate the problem more precisely, consider the following mathematical description. Suppose that there are n targets, indexed $i \in N = \{1,\ldots,n\}$, and that there are m illuminators, indexed $j \in M = \{1,\ldots,m\}$. Each target i has a specified engageability duration $[r_{i,j}, d_{i,j}]$, which depends on the location of the illuminator j, and a missile can score a successful strike only during this interval. Moreover, if target i is assigned to illuminator j, then it must be illuminated during its corresponding engageability interval for a given duration $\Delta_{i,j}$, prior to its scheduled engagement time $t_{i,j}$, in order to score a successful strike. In addition, there may be certain specified blocks of duration for which each illuminator j might be unavailable, because of previously scheduled assignments for which the corresponding missile launch has already been committed. In this respect, note that we can suppose that the n targets under consideration in the present scheduling cycle are either carry-over targets from a previous cycle, but for which the assigned missile launches have not as yet been irrevocably committed, or they are new targets which have appeared on the radar track file since the previous cycle. Accordingly, a feasible schedule is comprised of a set of illuminator durations for each target, along with an assignment of this duration to a particular illuminator, so that the illumination time intervals over the set of targets assigned to each illuminator during the present as well as the previous cycle are all disjoint. Among such feasible schedules, we are interested in finding a schedule which minimizes the total weighted engagement time.

Observe that the above problem can be conceptualized as a production floor shop scheduling problem in which there are n jobs to be scheduled on m parallel machines. For machine $j \in \mathrm{M}$, each job $i \in \mathrm{N}$ has a release time $r_{i,j}$, a completion time (due-date or deadline) $d_{i,j}$, and a processing time $\Delta_{i,j}$. Furthermore, each of the machines has certain specified unavailability blocks of time. The problem then is to construct a feasible schedule for the jobs on the machines, without preemption or splitting of a job, so as to meet the job release time and due date restrictions, and satisfy the machine availability requirements, while minimizing the total weighted flow time.

This scheduling problem includes as special cases several multiprocessor scheduling problems with release times, deadlines, and time-window constraints (Simons, 1981). For example, it includes the single machine scheduling problem with release times and deadlines (Grabowski et al., 1986, Leon and Wu, 1992, Potts, 1980), the multiple machine scheduling problem without time-window constraints (Sarin et al., 1988), problems dealing with the scheduling of unit-time processing jobs with release times and deadlines (Garey et al., 1982, Simons, 1983, Simons and Warmuth, 1989), and two-processor scheduling problems with start times and deadlines (Garey and Johnson, 1979). Although there exists an extensive literature on these problems, to the best of our knowledge, this research is the first to consider a general multiple machine scheduling problem which involves jobs with machine-dependent release times, deadlines, and time-windows, along with machine-downtime side constraints, while minimizing the total weighted flow time.

The following is an outline of this Part II. In Chapter 5, we develop a polynomial-time algorithm for the case when all the job processing times are equal (and unity without loss of generality), and the data is all integer. In Chapter 6, we develop a strong formulation for the case of the general problem, denoted by problem GP, and compare this with two alternative formulations via their underlying linear programming relaxations. By exploiting inherent special structures of the strong formulation of problem GP, we develop some classes of strong valid inequalities which strengthen the initial formulation. In Chapter 7, we address the development of several effective approximate and exact algorithms. In particular, we develop a branch-and-bound algorithm to find an ε-approximate or an exact solution to problem GP. This branch-and-bound algorithm incorporates the Lagrangian dual procedure for obtaining a tight lower bound, and uses the enhanced Lagrangian dual heuristic (HGPLD) for providing a tight upper bound. A depth-first-search rule is implemented for selecting a branching node and the optimal dual solution of each unfathomed node is used to initialize the solution for the Lagrangian relaxation problems of immediate descendant nodes.

In Chapter 8, we present computational results for various heuristics, and for the branch-and-bound algorithm. These results show that the heuristic HGPLD generates a primal feasible solution whose quality is better than that of the solution obtained by the LP based heuristic (HGPLP), but the computational time of the heuristic HGPLD is reduced by a factor of five as compared with that of HGPLP. We also present computational results for the branch-and-bound algorithm. In Chapter 9, we present an extension of problem GP and a zero-one integer programming formulation. Finally, Chapter 10 concludes this Part II with some research tasks for future investigation.

5. Δ-Multiple Duration Model and Algorithm

Consider a special case in which the job processing times $\Delta_{i,j}$ are all the same and equal to Δ, say. Moreover, suppose that for each $i \in N$, $r_{i,j} = r_i$ and $d_{i,j} = d_i$, for all $j \in M$, where r_i and d_i, are all multiples of Δ, and that the start and end times of the unavailability intervals on each machine are all also multiples of Δ. We denote this problem a *Δ-multiple duration scheduling problem*. Note that this situation may have been achieved by conservatively perturbing the data, given roughly equal and small processing times Δ_i, for $i \in N$. In other words, we assume that the problem data has been perturbed so that each processing time is set at $\Delta = \underset{i \in N}{\text{maximum}}\{\Delta_i\}$, the values r_i, $i \in N$, have been increased and the values d_i, $i \in N$, have been decreased just enough to satisfy the stated assumptions, and similarly, the machine unavailability durations have also been suitably expanded. By scaling, we can therefore assume without loss of generality that $\Delta = 1$, and that r_i, d_i, $i \in N$, and the start and end times of machine unavailability durations are all nonnegative integers.

Under this assumption, letting $T \equiv \underset{i \in N}{\text{maximum}}\{d_i\}$ be the problem horizon given by the latest due-date, the interval [0,T] may be decomposed into slots of unit duration, distributed over all the machines. Let $k \in K = \{1,\ldots,K\}$ index all the available or unassigned slots over all the machines, and let B denote the number of blocked-off or unavailable slots due to previous assignments. Of course, we should have $K \geq n$, or else the problem is clearly infeasible. Let τ_k denote the upper interval end-point of slot k, for $k \in K$. We will assume that the slots are ordered so that $\tau_1 \leq \tau_2 \leq \ldots \leq \tau_K$. Note that by the nature of the objective function and because of our assumption, in any optimal solution, if one exists, all jobs will be assigned to be performed precisely during some unit slot interval. Accordingly, for each job i, let us construct the set of feasible slots $S_i = \{k \in K : [\tau_k - 1, \tau_k] \subseteq [r_i, d_i]\}$, and define the binary decision variables x_{ik} to be 1 for $k \in S_i$, $i \in N$, if job i is assigned to slot k, and 0 otherwise. Note that $x_{ik} = 1$ means that the completion (engagement) time t_i of job i equals τ_k. The corresponding cost with respect to the objective function of making an assignment x_{ik} is given by

$$c_{ik} = w_i \tau_k \quad \text{for } k \in S_i,\ i \in N \tag{5.1}$$

where w_i is some positive weight for job $i \in N$.

Note that the approach of solving an assignment problem to allocate the n jobs to the K slots using the decision variables x_{ik} with cost coefficients c_{ik} would only be pseudopolynomial in time complexity, since K is of the order mT. A polynomial-time algorithm for finding an optimal assignment of jobs to slots, given the foregoing data, can be constructed as follows.

For each $t = 0,\dots,T\text{-}1$, let us define $N(t) = \{i \in N: r_i \leq t\}$, $n(t) = |N(t)|$, and let $\{k = 1,\dots,a(t)\}$ be the slots available over all the machines in the time interval $[0, t+1]$. Consider the following result which effectively decomposes the problem into separable subproblems, whenever possible, and hence admits a polynomial-time algorithm.

Proposition 5.1 Define $\hat{t} = \min\{t: n(t) \leq a(t)\}$. If the problem is feasible, then it has an optimal solution in which the jobs in the set $N(\hat{t})$ are assigned to the slots $\{1,\dots,n(\hat{t})\}$.

Proof. Assuming feasibility, consider any optimal solution to the problem. If $\hat{t} = 0$, i.e., $n(0) \leq a(0)$, then the result is trivially true. Otherwise, some a(0) jobs in N(0) must occupy the first a(0) slots in the given optimal solution by the nature of the objective function and because no jobs in N-N(0) can occupy these slots. Let us flag the jobs in N(0) which are assigned to the slots 1,...,a(0) in the given optimal solution. Inductively, for any $1 \leq \bar{t} \leq \hat{t}$, suppose that all the slots $1,...,a(\bar{t} - 1)$ are occupied by jobs in $N(\bar{t} - 1)$ for the given optimal solution. Flag the jobs assigned to these slots. If $\hat{t} = \bar{t}$, then the unflagged jobs in $N(\bar{t})$ must occupy the slots $\{a(\bar{t} - 1) + 1,...,a(\bar{t})\}$, since $n(\bar{t}) \leq a(\bar{t})$, and so these jobs can be made to occupy the slots $\{a(\bar{t} - 1) + 1, ..., n(\bar{t})\}$ while preserving the optimal objective value. On the other hand, if $\hat{t} > \bar{t}$, then as before, some $a(\bar{t}) - a(\bar{t} - 1)$ jobs from the unflagged ones in $N(\bar{t})$ must occupy the slots $a(\bar{t} - 1) + 1, ..., a(\bar{t})$ in the given optimal solution. Flagging these jobs and continuing inductively until the time $\hat{t}$, we have the result holding, and this completes the proof. ■

Proposition 5.1 prompts the following algorithm for the Δ-multiple duration scheduling problem.

Algorithm ASP

STEP 1. Find $\hat{t} = \min\{t\colon\ n(t) \leq a(t)\}$, and solve the following assignment problem (AP), where for $i \in N$, the cost coefficient c_{ik} is given by (5.1) if $k \in S_i$, and is taken as ∞ otherwise.

AP: Minimize $\displaystyle\sum_{i \in N(\hat{t})} \sum_{k=1}^{n(\hat{t})} c_{ik} x_{ik}$

subject to

$$\sum_{i \in N(\hat{t})} x_{ik} = 1 \quad \text{for } k = 1, \ldots, n(\hat{t})$$

$$\sum_{k=1}^{n(\hat{t})} x_{ik} = 1 \quad \text{for } i \in N(\hat{t})$$

$$x_{ik} \in \{0,1\} \text{ for } i \in N(\hat{t}),\ k = 1, \ldots, n(\hat{t}).$$

Assign each job $i \in N(\hat{t})$ to that slot k for which $x_{ik} = 1$ at optimality in AP.

STEP 2. Replace N by $N - N(\hat{t})$. If $N = \phi$, stop. Otherwise, treating $r_{n(\hat{t})+1}$ as the shifted time origin, consider a new scheduling problem defined on the revised job set N from this time onward. Return to Step 1.

Proposition 5.2 Algorithm ASP can be implemented to find an optimal solution, if one exists, to the Δ-multiple duration scheduling problem in polynomial-time with complexity $O(B + n^4)$.

Proof. First of all, note that in order to determine $\hat{t}$ and the associated set of jobs $N(\hat{t})$ at Step 1 of the algorithm, we can proceed by simply assigning the jobs in the order 1,...,n to the available slots, taking groups of jobs with the same r_i value at a time, until the release time of the next group exceeds the start time of the job which was last assigned. The jobs thus allocated constitute the set $N(\hat{t})$. If B_1 blocked-off durations were encountered in this process, this effort is of complexity $O(B_1 + n(\hat{t}))$, and so over the entire algorithm, this part is of complexity $O(B + n)$.

Now, suppose that the algorithm performs q loops, considering assignment problems of sizes $n_1, \ldots, n_q$, where $\sum_{p=1}^{q} n_p = n$. Each assignment problem of size $n_p x n_p$ can be solved in time $O(n_p^4)$ (see, for example, Bazaraa et al., 1990). Since $\sum_{p=1}^{q} n_p^4 \leq (\sum_{p=1}^{q} n_p)^4 =$ n^4, the complexity of solving all the assignment problems is $O(n^4)$. All the other steps of the algorithm are readily verified to be bounded by this quantity, and the overall algorithm is of complexity $O(B + n^4)$. This completes the proof. ■

Remark 5.1 At Step 1 of Algorithm ASP, we have taken $c_{ik} = \infty$ for $k \notin S_i$, for each $i \in \mathrm{N}$. In effect, if the scheduling problem is infeasible, then some assignment problem AP being infeasible, will have an infinitely large objective value. In such a case, in order to obtain a best compromise solution in the practical sense, as well as to prescribe a finite, useable value for the penalty to be assigned to the dummy arcs in AP, we remark that one can adopt the valid penalty bounds derived by Sherali (1988) for this purpose.

6. *General Discrete Data Problem (Problem GP)*

We now consider a general case of unequal job processing times $\Delta_{i,j}$, for $i \in \mathrm{N}$, $j \in \mathrm{M}$. However, we assume that the processing times, and the job availability interval (time-window) endpoints $r_{i,j}$ and $d_{i,j}$, as well as the end points of each unavailability interval duration on each of the machines $j \in \mathrm{M}$, are all multiples of some quantity Δ. (Possibly, $\Delta = 1$.) Scaling by Δ, we can assume that we have nonnegative integer valued data. We denote this *general discrete data case problem* as *problem GP*.

From the perspective of computational complexity, determining whether or not a feasible schedule to problem GP exists is NP-complete in the strong sense (Simons and Warmuth, 1989). This complexity result can be proven by a simple reduction from the 3-PARTITION problem. Furthermore, if m is arbitrary, then there is a similar reduction showing that the problem in which all jobs are released at time 0 and have the same integer deadline is also NP-complete in the strong sense (Simons and Warmuth, 1989). This implies that the feasibility problem of problem GP belongs to the class of strongly NP-complete problems. Therefore, we may not expect polynomial-time or even pseudopolynomial-time algorithms for problem GP, unless P=NP. Moreover, the problem remains strongly NP-complete even for the single-machine case, and even if only two different integer values exist for the release times and the deadlines (Simons and Warmuth, 1989).

To formulate this problem, recalling that τ_k denotes the upper interval end-point of slot k, for $k \in K$, let us define for each job $i \in N$, a set

$$\begin{aligned} S_i \;=\; \{k \in K :\; & [\tau_k - \Delta_{i,\mu_k}, \tau_k] \subseteq [r_{i,\mu_k}, d_{i,\mu_k}] \text{ and} \\ & (\tau_k - \Delta_{i,\mu_k}, \tau_k) \bigcap [\text{unavailability intervals} \\ & \text{for the machine corresponding to slot } k\,] = \phi\} \end{aligned}$$

where μ_k is the index of the machine that corresponds to slot k. Note that the set S_i denotes the slots $k \in K$ for which job i can be feasibly scheduled (independently) to complete by time τ_k. Accordingly, we define the binary decision variables $x_{ik} = 1$ if job i is scheduled to complete at the end of slot k, and $x_{ik} = 0$ otherwise. Observe that each slot k corresponds to some machine $\mu_k \in M$, and hence if $x_{ik} = 1$, then job i is implicitly being assigned to the machine μ_k. The cost in the objective function for this assignment is $c_{ik} = w_{ik} \tau_k$ for $i \in N$, $k \in S_i$, where w_{ik}, $k \in S_i$, $i \in N$, are some positive weights that reflect the relative threats posed by the targets (see Chapter 8, for specific choices of weight functions). Furthermore, in order to ensure that the schedule of the n jobs do not overlap on any of the machines, let us define for each slot k∈K, the set $J_k = \{(i, \rho) : i \in \mathrm{N}, \ \rho \in S_i$, and $[\tau_k - 1, \ \tau_k] \subseteq [\tau_\rho - \Delta_{i,\mu_k}, \ \tau_\rho]\}$. Note that for each k∈K, J_k is the set of combinations (i,ρ) such that slot k will be occupied if $x_{i\rho} = 1$. Then, problem GP can be formulated as follows.

GP: Minimize $\sum_{i \in N} \sum_{k \in S_i} c_{ik} x_{ik}$

subject to

$$\sum_{k \in S_i} x_{ik} = 1 \quad \text{for } i \in \mathrm{N} \qquad (6.1)$$

$$\sum_{(i,\rho) \in J_k} x_{i\rho} \leq 1 \quad \text{for } k \in \mathrm{K} \qquad (6.2)$$

$$x_{ik} \in \{0,1\} \quad \text{for } k \in S_i, \ i \in \mathrm{N} \qquad (6.3)$$

6.1. Alternative Formulations of Problem GP

We now consider an alternative formulation of the general battle-group engagement problem which possesses a generalized network structure. Toward this end, define

$$y_{(i,\rho)k} = x_{i\rho} \quad \text{for each } (i, \rho) \in J_k, \ \ k \in \mathrm{K}. \qquad (6.4)$$

Then, aggregating (6.4) suitably, we derive the following model GPGN which has a generalized network structure, and which is readily verified to be equivalent to model GP in the discrete sense.

GPGN: Minimize $\sum_{i \in N} \sum_{k \in S_i} c_{ik} x_{ik}$

subject to

$$\sum_{k\in S_i} x_{ik} = 1 \quad \text{for } i\in \mathrm{N}$$

$$\sum_{k\in J_{(i,\rho)}} y_{(i,\rho)k} = \Delta_{i,\mu_\rho} x_{i\rho} \quad \text{for } \rho\in S_i,\ i\in \mathrm{N}$$

$$\sum_{(i,\rho)\in J_k} y_{(i,\rho)k} \le 1 \quad \text{for } k\in \mathrm{K}$$

$$x_{ik}\in\{0, 1\} \quad \text{for } k\in \mathrm{K},\ i\in \mathrm{N}$$

$$0 \le y_{(i,\rho)k} \le 1, \quad \text{for } (i, \rho)\in J_k,\ k\in \mathrm{K}$$

$$\text{where } J_{(i,\rho)} = \{\, k : (i, \rho)\in J_k \,\} .$$

Note, however, that if $\overline{\mathrm{P}}$ denotes the linear programming relaxation of problem P, then, because of the aggregation of (6.4), we get the following result, where ν(P) denotes the optimal objective function value of a given problem P.

Proposition 6.1 $\nu\,(\overline{\mathrm{GP}}) \ge \nu\,(\overline{\mathrm{GPGN}})$. ■

Hence, although the underlying linear programming relaxation of GPGN can be solved using special generalized network flow techniques (see Bazaraa et al., 1990, for example), we would need to contend with a relatively weaker formulation of the problem.

We now develop a second alternative formulation to problem GP, following an approach that has been adopted in the scheduling literature for similar problems. Toward this end, let us define,

z_{ik} = 1 if slot k is occupied by job i and 0 otherwise,

$R_k = \{ i: (i, \rho) \in J_k \text{ for some } \rho \geq k \} \equiv \{ i: z_{ik} \text{ is defined} \}$, for each $k \in K$,

$F_i = \{ k: (i, \rho) \in J_k \text{ for some } \rho \geq k \} \equiv \{ k: z_{ik} \text{ is defined} \}$ for each $i \in N$,

$P_{ik} = \{ \rho : \rho \in F_i$, and either ρ and k are on different machines, or $|\tau_\rho - \tau_k| \geq \Delta_{i,\mu_k} \} \equiv$ slots that job i cannot possibly occupy if $z_{ik} = 1$, and

$C_{ik} = \{ \rho$: slot ρ is on the same machine as slot k, and $\tau_\rho \leq \tau_k \leq \tau_\rho + \Delta_{i,\mu_k} - 1 \}$ $\equiv \Delta_{i,\mu_k}$ consecutive slots (where available) preceding and including slot k on the same machine.

The alternative formulation (GPAF) can then be stated as follows, and is readily verified to be again equivalent to problem GP in the discrete sense, but as we shall see in Proposition 6.2 below, it tends to be a weaker formulation in the continuous sense. Moreover, because of the additional advantage of inherent special structures possessed by model GP (see Section 6.2), we will prefer to use model GP.

GPAF: Minimize $\sum_{i \in N} \xi_i$

subject to

$$\xi_i \geq w_{ik} \tau_k z_{ik} \quad \text{for } k \in F_i, \; i \in N \tag{6.5}$$

$$\sum_{i \in R_k} z_{ik} \leq 1 \quad \text{for } k \in K \tag{6.6}$$

$$\sum_{j \in M} \frac{1}{\Delta_{i,j}} \sum_{k \in F_i \ni \mu_k \equiv j} z_{ik} = 1 \quad \text{for } i \in N \tag{6.7}$$

$$\sum_{\rho \in P_{ik}} z_{i\rho} \leq \max_{j \in M} \{\Delta_{i,j}\}(1 - z_{ik}) \quad \text{for } k \in F_i, \; i \in N \tag{6.8}$$

$$z_{ik} \in \{0,1\}, \quad \text{for } k \in K,\ i \in N$$

$$\xi_i \geq 0, \quad \text{for } i \in N$$

Proposition 6.2 Suppose that $\Delta_{i,j} = \Delta_i$ $\forall j \in M$, for each $i \in N$, and that $w_{ik} = w_{i\mu_k}$ $\forall k \in S_i$, $i \in N$. Then, for every feasible solution to $\overline{\text{GP}}$ having an objective function value v_1, there corresponds a feasible solution to $\overline{\text{GPAF}}$ having an objective function value v_2 that is less-than-or-equal-to v_1.

Proof. Suppose that $\bar{x}$ is a feasible solution to $\overline{\text{GP}}$. Define $Z_{ik} = \{\rho :\ (i, \rho) \in J_k\}$ and consider the following transformation: $\bar{z}_{ik} = \sum_{\rho \in Z_{ik}} \bar{x}_{i\rho}$ for $i \in N$, $k \in K$. Then, since the constraints (6.2) can be restated as $\sum_{i \in R_k} \sum_{\rho \in Z_{ik}} x_{i\rho} \leq 1$, for $k \in K$, we have, $\sum_{i \in R_k} \bar{z}_{ik} \leq 1$, for each $k \in K$. Therefore, $\bar{z}$ is feasible to (6.6), and we also have $0 \leq \bar{z}_{ik} \leq 1$ $\forall$ $k \in K$, $i \in N$. Moreover, for each $i \in N$, we have that for $j \in M$,

$$\sum_{k \in F_i \ni \mu_k \equiv j} \bar{z}_{ik} = \sum_{k \in F_i \ni \mu_k \equiv j} \sum_{\rho \in Z_{ik}} \bar{x}_{i\rho} = \Delta_{i,j} \sum_{\rho \in S_i \ni \mu_\rho \equiv j} \bar{x}_{i\rho}.$$

Hence, from (6.1), we get,

$$\sum_{j \in M} \frac{1}{\Delta_{i,j}} \sum_{k \in F_i \ni \mu_k \equiv j} \bar{z}_{ik} = \sum_{j \in M} \sum_{\rho \in S_i \ni \mu_\rho \equiv j} \bar{x}_{i\rho} = \sum_{\rho \in S_i} \bar{x}_{i\rho} = 1 \quad \forall i \in N,$$

and so, $\bar{z}$ is feasible to (6.7). Next, consider (6.8). Define for any $i \in N$ and $k \in F_i$,

$A_{ik} = \{ \ \rho :\ \rho$ and k are on the same machine with $|\ \tau_\rho - \tau_k| < \Delta_{i,\mu_k} \}$.

Hence, we obtain,

$$\sum_{q \in P_{ik}} \bar{z}_{iq} = \sum_{q \in F_i} \bar{z}_{iq} - \sum_{q \in A_{ik}} \bar{z}_{iq} = \sum_{q \in F_i} \bar{z}_{iq} - \sum_{q \in A_{ik}} \sum_{\rho \in Z_{iq}} \bar{x}_{i\rho}. \tag{6.9}$$

Using (6.7), we get,

$$\sum_{q \in F_i} \bar{z}_{iq} = \sum_{j \in M} \sum_{k \in F_i \ni \mu_k \equiv j} \bar{z}_{ik} \leq [\max_{j \in M}\{\Delta_{i,j}\}] \sum_{j \in M} \frac{1}{\Delta_{i,j}} \sum_{k \in F_i \ni \mu_k \equiv j} \bar{z}_{ik} = \max_{j \in M}\{\Delta_{i,j}\}. \tag{6.10}$$

Furthermore, since $\Delta_{i,j} = \Delta_i, \ \forall j \in M$

$$\sum_{q \in A_{ik}} \sum_{\rho \in Z_{iq}} \bar{x}_{i\rho} \geq \sum_{q \in A_{ik}} \sum_{\rho \in Z_{iq} \cap Z_{ik}} \bar{x}_{i\rho} = \sum_{\rho \in Z_{ik}} \sum_{q \in C_{i\rho}} \bar{x}_{i\rho} = \Delta_{i,\mu_k} \sum_{\rho \in Z_{ik}} \bar{x}_{i\rho} = \max_{j \in M}\{\Delta_{i,j}\} \bar{z}_{ik}. \tag{6.11}$$

Hence, using (6.10) and (6.11) in (6.9), we deduce that

$$\sum_{q \in P_{ik}} \bar{z}_{iq} \leq \max_{j \in M}\{\Delta_{i,j}\}(1 - \bar{z}_{ik}) \quad \text{for } k \in F_i, \ i \in N,$$

that is, $\bar{z}$ is feasible to (6.8) as well. Finally, consider the objective function values v_1 and v_2 corresponding to $\bar{x}$ and $\bar{z}$ in $\overline{\text{GP}}$ and $\overline{\text{GPAF}}$, respectively, where $\xi_i = \max\{w_{ik}\tau_k\bar{z}_{ik} : k \in F_i\} \geq 0$ for $i \in N$. Accordingly, we have that, $v_1 = \sum_{i \in N} \sum_{\rho \in S_i} w_{i\rho}\tau_\rho\bar{x}_{i\rho}$, and that

$$v_2 = \sum_{i \in N} \xi_i = \sum_{i \in N} \max\{w_{ik}\tau_k \sum_{\rho \in Z_{ik}} \bar{x}_{i\rho} : \quad k \in F_i\} \leq \sum_{i \in N} \max\{w_{ik} \sum_{\rho \in Z_{ik}} \tau_\rho \bar{x}_{i\rho} : k \in F_i\}.$$

Since for each $i \in \mathrm{N}$, the foregoing maximum is realized for some slot $k \in F_i$, and since the weights w_{il} are the same for all slots l on that machine μ_k, we have that $v_2 \leq \sum_{i \in \mathrm{N}} \sum_{\rho \in S_i} w_{i\rho} \tau_\rho \overline{x}_{i\rho} = v_1$. This completes the proof. ■

Example 6.1 The following scheduling problem illustrates the relative strength of model GP over model GPAF. The problem is to schedule four jobs on a single machine with time-window constraints. Release times, deadlines, and processing times are as follows.

$$(r_{1,1},\ d_{1,1},\ \Delta_{1,1},\ w_{1,1}) = (0,\ 5,\ 1,\ 1)$$
$$(r_{2,1},\ d_{2,1},\ \Delta_{2,1},\ w_{2,1}) = (0,\ 3,\ 1,\ 3)$$
$$(r_{3,1},\ d_{3,1},\ \Delta_{3,1},\ w_{3,1}) = (1,\ 4,\ 2,\ 3)$$
$$(r_{4,1},\ d_{4,1},\ \Delta_{4,1},\ w_{4,1}) = (3,\ 6,\ 1,\ 3)$$

A summary of the results is given below.

Model	Number of variables	Number of rows	$v(.)$	Remark
$\overline{\text{GP}}$	13	10	29.0	Integer solution
$\overline{\text{GPAF}}$	18	38	15.07	Non integer

Example 6.2 Note that the foregoing example satisfies the assumptions of Proposition 6.2. However, as the following example illustrates, a similar phenomenon was observed on instances that violate the assumptions of Proposition 6.2. Consider the following problem of scheduling two jobs on two machines having release times, deadlines, and processing times as follows.

$$(r_{1,1},\ d_{1,1},\ \Delta_{1,1},\ w_{1,1}) = (0,\ 5,\ 1,\ 1),\ (r_{1,2},\ d_{1,2},\ \Delta_{1,2},\ w_{1,2}) = (1,\ 4,\ 2,\ 3)$$
$$(r_{2,1},\ d_{2,1},\ \Delta_{2,1},\ w_{2,1}) = (0,\ 3,\ 1,\ 3),\ (r_{2,2},\ d_{2,2},\ \Delta_{2,2},\ w_{2,2}) = (0,\ 3,\ 2,\ 3)$$

A summary of the results is given below.

Model	Number of variables	Number of rows	$v(.)$	Remark
$\overline{\text{GP}}$	12	11	5.0	Integer solution
$\overline{\text{GPAF}}$	16	39	1.4967	Non integer

6.2. Special Structures of Model GP

We have seen that model GP dominates some related, plausible alternative formulations of problem GP. It also turns out that, in addition, model GP enjoys certain inherent special structures. These structures can be identified as follows: (i) a generalized upper bounding (GUB) structure of constraints (6.1), (ii) a block-diagonal (angular) structure of constraints (6.2), and also (iii) an interval matrix structure of the set packing constraints (6.2).

Note that the GUB structure can be exploited in solving linear programming relaxations, but more importantly, it can be used effectively to partition the search space in a branch-and-bound context, and provides a useful construct in various primal and dual approaches as we shall see later. Next, let us expose the block-diagonal or angular structure of model GP, by suitably rearranging its constraints. Let Q be the set of separate pieces of contiguous machine availability intervals, $q \in Q$, over all the machines, and over the time horizon [0,T]. Define the sets $T_q = \{r : \text{slot } r \text{ is part of the interval } q\}$, for each $q \in Q$. Note that each $T_q \subseteq K$ corresponds to slots that lie within the time interval q, for $q \in Q$. Then, the angular structured model GP, denoted by $GP_{angular}$, can be described as follows.

$$\text{GP}_{\text{angular}}: \quad \text{Minimize} \sum_{q \in Q} \sum_{i \in N} \sum_{k \in (T_q \cap S_i)} c_{ik} x_{ik}$$

subject to

$$\sum_{k \in S_i} x_{ik} = 1 \quad \text{for } i \in N \tag{6.12}$$

$$[\sum_{(i,\rho) \in J_k} x_{ik} \leq 1 \quad \text{for } k \in T_q], \quad \text{for each block } q \in Q \tag{6.13}$$

$$x_{ik} \in \{0,1\} \quad \text{for } k \in S_i,\ i \in N \tag{6.14}$$

While this angular structure can be exploited in a Dantzig-Wolfe decomposition or Lagrangian relaxation framework, it also turns out that these block diagonal packing constraints possess an interval matrix structure. This structure will play a central role in the development of heuristics as well as exact algorithms, because it reveals a hidden network structure corresponding to an acyclic directed graph (digraph). Consequently, problem GP can be interpreted as finding a shortest path satisfying the GUB constraints. Let us consider the following proposition that provides a mechanism for transforming the given interval matrix structure into an acyclic directed graph.

Proposition 6.3 The matrix of the set packing constraints (6.2) of model GP is an interval matrix, and the underlying graph is an acyclic digraph whose first node is a supply node with capacity 1 and its terminus node is a demand node with demand 1, while all other nodes are transshipment nodes.

Proof. If the set packing constraints (6.2) are permuted according to slots as they appear from left to right on each machine at a time, then each defined variable x_{ik}

appears in some Δ_{l,μ_k} consecutive constraints. Hence, the coefficient matrix of (6.2) is an interval matrix, and therefore also includes a hidden a network structure. To reveal this structure, we simply need to introduce slack variables s_k, $k \in K$, in (6.2) and then perform row operations which correspond to subtracting from each equation its preceding one, and then finally add a redundant equation equal to the negative sum of the resulting equations. This yields a node-arc incidence matrix of a graph G having a node set $V = \{1, ..., K+1\}$, and an arc set $E = E_s \bigcup E_x$, where $E_s = \{(i, i+1)$: for each slack variable s_i, $i=1, ..., K\}$ and $E_x = \{(p, q+1)$: for each column of the interval matrix whose first 1 is in row p and whose last 1 is in row q, for each variable x_{ik}, $k \in S_i$, $i \in N\}$. Also, by virtue of the row operations, the first node has a capacity of 1, the last node has a demand of 1, and all the other nodes are transshipment nodes. Since any arc (i, j) of G is directed from node i to node j only if $i < j$, the graph G is acyclic. Hence, any zero-one feasible solution that satisfies the set packing constraints is a simple (directed) path from the first node to the last node on the above digraph G. This completes the proof. ■

Remark 6.1 We remark here that problem GP can be interpreted as a shortest path problem on the acyclic digraph G from node 1 to node K+1, but the path must include exactly one arc from each of the disjoint arc sets E_1, E_2, ..., E_n, where $\bigcup_{i \in N} E_i \equiv E_x$, and where each E_i, $i \in N$, contains arcs corresponding to the variables x_{ik} for $k \in S_i$. Note that the acyclic digraph G may be a multigraph, i.e., the graph G may have multiple arcs between nodes i and j of G.

Example 6.3 Model GP and $\text{GP}_{\text{angular}}$ Consider the following problem that schedules three jobs on two machines with time-window constraints, as given below.

$(r_{1,1},\ d_{1,1},\ \Delta_{1,1},\ w_{1,1}) = (0, 4, 2, 1)$, $(r_{1,2},\ d_{1,2},\ \Delta_{1,2},\ w_{1,2}) = (0, 4, 2, 1)$

$(r_{2,1},\ d_{2,1},\ \Delta_{2,1},\ w_{2,1}) = (1, 4, 2, 2)$, $(r_{2,2},\ d_{2,2},\ \Delta_{2,2},\ w_{2,2}) = (1, 4, 2, 2)$

$(r_{3,1},\ d_{3,1},\ \Delta_{3,1},\ w_{3,1}) = (2, 3, 1, 16)$, $(r_{3,2},\ d_{3,2},\ \Delta_{3,2},\ w_{3,2}\ ) = (2, 3, 1, 16)$

blocked-off duration of machine 1 = [1,2],

blocked-off duration of machine 2 = ϕ.

Machine 1 slots	1		2	3	
Machine 2 slots	4	5	6	7	
Time units	0	1	2	3	4

Model GP for this example can be stated as follows.

Row	x_{13}	x_{15}	x_{16}	x_{17}	x_{23}	x_{26}	x_{27}	x_{32}	x_{36}	RHS
COST	4	2	3	4	8	6	8	48	48	MIN
1	1	1	1	1						$=1$
2					1	1	1			$=1$
3								1	1	$=1$
4	1				1			1		≤ 1
5	1				1					≤ 1
6		1								≤ 1
7		1	1			1				≤ 1
8			1	1		1	1		1	≤ 1
9				1			1			≤ 1

Consider an equivalent shortest path problem with side constraints corresponding to model GP. The arc set of the associated directed graph G, corresponding to the variables x_j and the slack variables in the constraints (6.2) is as follows. $E_1 = \{x_{13}, x_{15}, x_{16}, x_{17}\}$, $E_2 = \{x_{23}, x_{26}, x_{27}\}$, $E_3 = \{x_{32}, x_{36}\}$, and $E_S = \{s_4, s_5, s_6, s_7, s_8, s_9\}$, where s_i represents the slack variable in the ith row in the above table. For this example, we seek an optimal shortest path from node 1 to node 7, which must include exactly one arc from each arc set E_i for i = 1, 2, 3. Such an optimal solution is given by $x_{15} = 1$, $x_{27} = 1$, $x_{32} = 1$, and all the other variables are equal to zero. Consider the following description of $GP_{angular,}$ where Q = {1, 2} , and T_1 = [time blocks 3 and 4], T_2 = [time blocks 5, 6, 7, and 8]. Since the time-window constraints of the three jobs do not use time block 1 and time block 2, T_i for i = 1,2 does not include time blocks 1 and 2. The constraints (6.13) shown in Rows 4-9 of the table exhibit two separable blocks, namely, (Rows 4 and 5) and (Rows 6, 7, and 8). Since these blocks have no common variables, they can be decomposed into two separable constraints, when the GUB constraints are relaxed in a Lagrangian framework.

Row	x_{32}	x_{13}	x_{23}	x_{15}	x_{16}	x_{26}	x_{36}	x_{17}	x_{27}	RHS
COST	48	4	8	2	3	6	48	4	8	MIN
1		1		1	1			1		= 1
2			1			1			1	= 1
3	1						1			= 1
4	1	1	1							≤ 1
5		1	1							≤ 1
6				1						≤ 1
7				1	1	1				≤ 1
8					1	1	1	1	1	≤ 1
9								1	1	≤ 1

6.3. Tight Formulations Using Strong Valid Inequalities

The initial formulation of model GP can be further tightened by introducing some strong valid inequalities. These valid inequalities can be used in the framework of a branch-and-cut scheme by solving a readily available separation problem, or by implementing an ad-hoc procedure for finding a valid inequality that cuts off a fractional optimal solution obtained for the linear programming relaxation of model GP, denoted by $\overline{\text{GP}}$.

6.3.1 *A Class of Valid Inequalities from the Facets of the Generalized Assignment Polytope*

We can transform model GP into an equivalent set packing problem with all less-than-equal-to type constraints, and having a modified objective coefficient vector $\bar{c}_{ik} = \theta - c_{ik}$, for a sufficiently large number θ, by writing its objective function as maximize $-\sum_{i \in N} \sum_{k \in S_i} c_{ik} x_{ik}$ and including the (penalty) term $\theta \sum_{i \in N} (\sum_{k \in S_i} x_{ik} - 1)$ in this objective function. This transformation yields the following equivalent formulation of problem GP, denoted by GPP, where we have further decomposed the variables over the available blocks $q \in Q$ in order to expose certain special structures.

GPP: Maximize $\displaystyle\sum_{q \in Q} \sum_{i \in N} \sum_{k \in (T_q \cap S_i)} \bar{c}_{ik} x_{ik} - n\theta$

subject to

$$\sum_{k \in S_i} x_{ik} \leq 1 \quad \text{for } i \in N \tag{6.15}$$

$$\Big[\sum_{(l,\rho) \in J_k} x_{ik} \leq 1 \quad \text{for } k \in T_q \Big], \quad \text{for each block } q \in Q \tag{6.16}$$

$$x_{ik} \in \{0,1\} \quad \text{for } k \in S_i,\ i \in N \tag{6.17}$$

We now consider a class of valid inequalities for model GP by exploiting the GUB structured constraints (6.15) and the interval matrix structured constraints (6.16). Denote as CHGPP the convex hull of zero-one vertices of the polytope associated with GPP, that is, CHGPP = conv $\{x \in R^{n_0}$: x satisfies (6.15), (6.16), (6.17)$\}$, where $n_0 = |\bigcup_{i \in N} S_i| = \sum_{i \in N} |S_i|$.

In order to generate a class of valid inequalities for CHGPP, consider the following surrogates of the constraints (6.16), corresponding to each machine availability interval $q \in Q$.

$$\sum_{i \in N} \sum_{k \in T_q \cap S_i} \Delta_{i,\mu_k} x_{ik} \le |T_q| \qquad \text{for each } q \in Q. \tag{6.18}$$

Inequalities (6.18) assert that the total job occupancy time over the interval q is no more than the total available time $|T_q|$. Note that $\Delta_{i,\mu_k} = \Delta_{i,j}$ for each $k \in T_q \cap S_i$ in the time block q, where $\mu_k = j$ for $k \in T_q$. Consider the relaxed, GUB constrained, generalized knapsack polytope (GPKP) associated with the constraints (6.15), (6.17), and (6.18), that is,

$$\text{GPKP} = \text{conv}\{x \in R^{n_0}: \begin{array}{ll} \sum_{k \in S_i} x_{ik} \le 1 & \text{for } i \in N, \\ \sum_{i \in N} \sum_{k \in T_q \cap S_i} \Delta_{i,\mu_k} x_{ik} \le |T_q| & \text{for each } q \in Q, \\ x_{ik} \in \{0,1\} \text{ for } k \in S_i,\ i \in N\ \} & \end{array}$$

Since a facet of GPKP is also a valid inequality for CHGPP and for problem GP, we investigate the facial structure of GPKP. (Recall that a *facet* of an n_0-dimensional polytope is a maximal proper face, or, equivalently, a face of dimension $n_0 - 1$. The inequality $\pi x \le \pi_0$ is a facet defining inequality for GPKP if and only if (i) $x \in$ GPKP implies $\pi x \le \pi_0$, and (ii) there exist n_0 affinely independently vertices x^j of GPKP satisfying $\pi x^j = \pi_0$, $j = 1, \ldots, n_0$. Such facet defining inequalities are often themselves referred to as facets. In general, affine independence does not imply linear independence. However, for facets such that $\pi_0 > 0$, it is well known that the two notions are equivalent. Since GPKP is a full dimensional polytope, there exists a set of facets uniquely defined up to multiplication by a positive scalar.)

It can be easily verified that all nonnegativity constraints $x_{ik} \geq 0$ are facets of GPKP. These constraints are referred to as *trivial facets* for GPKP. In addition, it is shown by Hammer et al.(1975) that, since the constraint matrix coefficients of GPKP are nonnegative and all constraints are less-than-or-equal-to type inequalities, all *nontrivial facets* are of the form $\sum_{i \in N} \sum_{k \in S_i} \pi_{ik} x_{ik} \leq \pi_0$, denoted as (π, π_0), having $\pi_{ik} \geq 0$ and $\pi_0 > 0$.

Proposition 6.4 Suppose that $\pi x \leq \pi_0$ with $\pi_0 > 0$ is a facet of GPKP. Then for each $k \in S_i$, $i \in N$, there exists a zero-one vertex $\bar{x}$ of GPKP such that $\bar{x}_{ik} = 1$ and $\pi \bar{x} = \pi_0$.

Proof. Since $\pi x \leq \pi_0$ is a nontrivial facet of GPKP and GPKP is a full dimensional polytope, there exist n_0 linearly independent vertices x^j of GPKP such that $\pi x^j = \pi_0$, $j = 1, \ldots, n_0$. Let R be the matrix whose rows represent these vertices. If for some $k \in S_i$, $i \in N$, we have $x^j_{ik} = 0$ for $j = 1, \ldots, n_0$, then this implies that R is singular. Hence, for each $k \in S_i$, $i \in N$, $x^j_{ik} = 1$ for some $j \in \{1, \ldots, n_0\}$, and this completes the proof. ■

Due to the interval matrix structure of constraints (6.16), we can further restrict the investigation of the facial structure of GPKP. Toward this end, consider the following proposition.

Proposition 6.5 For some $q \in Q$, if $\Delta_{i,k1} = \Delta_{i,k2}$ for all $k1, k2 \in T_q \cap S_i$, $(k1 \neq k2)$, for some $i \in N$, then $\pi_{ik1} = \pi_{ik2}$ in any facet $\sum_{i \in N} \sum_{k \in S_i} \pi_{ik} x_{ik} \leq \pi_0$ of GPKP with $\pi_0 > 0$.

Proof. Suppose that $\pi x \leq \pi_0$ is a facet of GPKP such that $\pi_{ik1} \neq \pi_{ik2}$. Without loss of generality, let $\pi_{ik1} < \pi_{ik2}$. Since $\pi x \leq \pi_0$ is a facet of GPKP, by Proposition 6.4, there exists a zero-one vertex $\bar{x} \in$GPKP such that $\pi\bar{x} = \pi_0$ and $\bar{x}_{ik1} = 1$, so that $\bar{x}_{ik2} = 0$ by (6.15). Denote $\tilde{x} = \bar{x}$, except that $\tilde{x}_{ik1} = 0$ and $\tilde{x}_{ik2} = 1$. Then $\tilde{x}$ is also feasible to GPKP. But $\pi\tilde{x} = \pi_0 - \pi_{ik1} + \pi_{ik2} > \pi_0$, leading to a contradiction. This completes the proof. ■

Proposition 6.5 implies that we can further restrict our investigation of the facial structure of GPKP as follows. For each $q \in Q$, let $y_{iq} = \sum_{k \in T_q \cap S_i} x_{ik}$ $\forall i \in N$ such that $T_q \cap S_i \neq \phi$. For each $q \in Q$, denote by μ_q the common value of μ_k for all $k \in T_q$. Consider the convex hull of the following generalized assignment polytope, denoted by GPGAP, where $n_1 = \sum_{q \in Q} |\{i \in N : T_q \cap S_i \neq \phi\}|$, and where undefined variables are assumed to be zero.

$$\text{GPGAP} = \text{conv}\{y \in R^{n_1} : \begin{array}{ll} \sum_{q \in Q} y_{iq} \leq 1 & \text{for each } i \in N, \\ \sum_{i \in N} \Delta_{i,\mu_q} y_{iq} \leq |T_q| & \text{for each } q \in Q, \\ y_{iq} \in \{0,1\} & \} \end{array}$$

Remark 6.2 Gottlieb and Rao (1990a, 1990b) describe some classes of facets for the generalized assignment problem GAP. All these facets can be implemented as valid inequalities for model GP, as shown in the following proposition.

Proposition 6.6 Suppose that $\sum_{i \in N} \sum_{q \in Q} \pi_{iq} y_{iq} \leq \pi_0$ with $\pi_0 > 0$ is a facet of GPGAP. Then

$$\sum_{i \in N} \sum_{q \in Q} \sum_{k \in T_q \cap S_i} \pi_{iq} x_{ik} \leq \pi_0 \tag{6.19}$$

is a facet of GPKP, which consequently is a valid inequality for problem GP.

Proof. Since $\sum_{i \in N} \sum_{q \in Q} \pi_{iq} y_{iq} \leq \pi_0$ is a facet of GPGAP, and since $y_{iq} = \sum_{k \in T_q \cap S_i} x_{ik}$, $\forall (i, q)$, is feasible to GPGAP for any feasible solution x to GPKP, the inequality (6.19) is valid for GPKP, and consequently, is valid for problem GP. Now, suppose on the contrary that the valid inequality (6.19) is not a facet of GPKP. Since GPKP is a full dimensional polytope, it follows that the inequality (6.19) is implied by some facets of GPKP. Suppose that the inequality (6.19) can be represented by two or more distinct facets $(\bar{\pi}^j, \bar{\pi}^j_0)$ $j \in J$ of GPKP such that $\pi = \sum_{j \in J} \lambda_j \bar{\pi}^j$ and $\pi_0 = \sum_{j \in J} \lambda_j \bar{\pi}^j_0$ where $\sum_{j \in J} \lambda_j = 1$, and $\lambda_j > 0$, $\forall j \in J$. By Proposition 6.5, these facets can be written as

$$\sum_{i \in N} \sum_{q \in Q} \bar{\pi}^j_{iq} \left[\sum_{k \in T_q \cap S_i} x_{ik} \right] \leq \bar{\pi}^j_0 \quad \forall j \in J. \tag{6.20}$$

Now, note that for any $j \in J$, $\sum_{i \in N} \sum_{q \in Q} \bar{\pi}^j_{iq} y_{iq} \leq \bar{\pi}^j_0$ are also valid inequalities for GPGAP. This follows because if for any $j \in J$, we have $\sum_{i \in N} \sum_{q \in Q} \bar{\pi}^j_{iq} \hat{y}_{iq} > \bar{\pi}^j_0$ for some binary solution $\hat{y}$ to GPGAP, then by selecting $\hat{x}$ as a binary vector such that for each (i,q) for which y_{iq} is defined, the correspondence $\sum_{k \in T_q \cap S_i} \hat{x}_{ik} = \hat{y}_{iq}$ holds, we see that $\hat{x}$ is feasible to GPKP, but (6.20) is violated, a contradiction. But this means from above that these valid inequalities imply $\sum_{i \in N} \sum_{q \in Q} \pi_{iq} y_{iq} \leq \pi_0$, which is a contradiction to the hypothesis that the latter inequality is a facet of GPGAP. Hence, the inequality (6.19) is a facet of GPKP, and this completes the proof. ■

The following proposition provides a separation procedure in order to obtain a class of valid inequalities for problem GP from each individual knapsack constraint of GPGAP.

Proposition 6.7 All nontrivial facets associated with the individual knapsack constraints of GPGAP are also facets of GPGAP, which consequently correspond to facets of GPKP via equation (6.19).

Proof. The proof follows by Proposition 5.10 in Gottlieb & Rao (1990a) and by Proposition 6.6. ∎

Remark 6.3 We can solve a standard separation problem (see Nemhauser and Wolsey, 1988) in order to (possibly) obtain a minimal cover inequality from a knapsack constraint of GPGAP that cuts off a given fractional solution to $\overline{\text{GP}}$. A sequential lifting procedure can polynomially strengthen such a minimal cover inequality to a facet of the knapsack polytope (see Zemel, 1988), and by Proposition 6.6, such a constraint would then correspond to a facet of GPKP. The following example illustrates this remark.

Example 6.4 Consider the following problem that has four jobs to be scheduled on a single machine with the data,

$(r_{1,1},\ d_{1,1}, \Delta_{1,1},\ w_{1,1}) = (0, 9, 1, 1),$

$(r_{2,1},\ d_{2,1}, \Delta_{2,1},\ w_{2,1}) = (0, 9, 1, 1),$

$(r_{3,1},\ d_{3,1}, \Delta_{3,1},\ w_{3,1}) = (0, 9, 2, 1),$

$(r_{4,1},\ d_{4,1}, \Delta_{4,1},\ w_{4,1}) = (0, 9, 2, 1).$

and has the duration [3,5], blocked-off on the single machine. The available time slots are numbered $k = 1,\ldots,K = 7$.

Machine 1 slots	1	2	3			4	5	6	7	
Time units	0	1	2	3	4	5	6	7	8	9

An optimal solution $\bar{x}$ to $\overline{GP}$ is given by $\bar{x}_{11} = 1$, $\bar{x}_{22} = 1/2$, $\bar{x}_{23} = 1/2$, $\bar{x}_{35} = 1$, $\bar{x}_{43} = 1/2$, $\bar{x}_{49} = 1/2$, and $\bar{x}_{ik} = 0$ otherwise. For $q = 1$, we have that $T_q = \{1, 2, 3\}$ and the surrogate inequality of the type (6.18) is

$$x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + 2x_{32} + 2x_{33} + 2x_{42} + 2x_{43} \leq 3. \qquad (6.21)$$

Consider the corresponding continuous solution $\bar{y}$ translated to y variables as given by $\bar{y}_{11} = 1$, $\bar{y}_{12} = 0$, $\bar{y}_{21} = 1$, $\bar{y}_{22} = 0$, $\bar{y}_{31} = 0$, $\bar{y}_{32} = 1$, $\bar{y}_{41} = 1/2$, and $\bar{y}_{42} = 1/2$. Corresponding to (6.21), the knapsack constraint of GPGAP is $y_{11} + y_{21} + 2y_{31} + 2y_{41} \leq 3$. From this knapsack constraint, we can obtain an extended minimal cover inequality, $y_{11} + y_{21} + y_{31} + y_{41} \leq 2$, which is a facet of the corresponding knapsack polytope and which deletes the solution $\bar{y}$. By Proposition 6.7, we have the following facet of GPGAP given by (6.19), which is a valid inequality for model GP: $x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{32} + x_{33} + x_{42} + x_{43} \leq 2$. Note that this inequality cuts off the fractional optimal solution of $\overline{GP}$. Furthermore, augmenting $\overline{GP}$ by adding this cutting plane to model GP produces the integer optimal solution given by $x_{11} = 1$, $x_{24} = 1$, $x_{36} = 1$, $x_{43} = 1$, and $x_{ik} = 0$ otherwise.

6.3.2 A Class of Valid Inequalities from the Clique Inequality of the Set Packing Polytope

In this section, we consider a class of strong valid cutting plane inequalities that are derived from the facets of the set packing polytope.

For the convenience of presentation, let us define some notation. Let the index set M_G denote indices of the GUB constraints (6.1), and similarly, let M_P denote indices of the set packing constraints (6.2). Note that each constraint $i \in M_G$ corresponds to a target i, and each constraint $k \in M_P$ corresponds to a slot k, so that we can assume $M_G \equiv \{1,..,n\}$, and $M_P \equiv \{n+1,...,n+K\}$. Define $M_{GP} \equiv M_G \bigcup M_P$. Let us also refer to the variables as being simply singly subscripted, and given by x_j for j belonging to an index set N_{GP}. Furthermore, denote by N_i the set of indices of variables contained in constraint $i \in M_{GP}$, and conversely, let M_j denote the index set of constraints in M_{GP} that include the variable x_j, for each $j \in N_{GP}$. Note that with this revised compact notation, model GP can be simply restated as

$$\text{GP: Minimize } \{ \sum_{j \in N_{GP}} c_j x_j : \sum_{j \in N_i} x_j \ = 1 \ \forall \ i \in M_G, \ \sum_{j \in N_i} x_j \ \leq 1 \quad \forall \ i \in M_P, \ x_j \in \{0, 1\} \ \forall \ j \in N_{GP}\},$$

and similarly, we may restate model GPP as follows.

$$\text{GPP: Maximize } \{ \sum_{j \in N_{GP}} \bar{c}_j x_j : \sum_{j \in N_i} x_j \ \leq 1 \ \forall \ i \in M_G, \ \sum_{j \in N_i} x_j \ \leq 1 \quad \forall \ i \in M_P, \ x_j \in \{0, 1\} \ \forall \ j \in N_{GP}\}$$

For simplicity of notation, let us rewrite model GPP as

$$\text{GPP}: \ \text{Maximize}\{\bar{c}x \ : \ Ax \leq e, \ \ x_j \in \{0,1\}\},$$

where $e = \{1,...,1\}^T$. The *intersection graph* $G_I = (V, E)$ of the matrix A has one node for each column of the matrix A, and one arc for every pair of non-orthogonal columns of A, i.e., $(i, j) \in E$ if and only if the corresponding columns a_i, a_j of A satisfy $a_i^T a_j \geq 1$. Let A_{G_I} be the node-arc incidence matrix of G_I, and denote by VP the weighted *node (vertex) packing* problem on the intersection graph G_I. Define a *clique* as a maximal complete subgraph of G_I. Then, we have a well-known class of facets for CHGPP (see Padberg, 1973).

Proposition 6.8 The inequality $\sum_{j\in H} x_j \leq 1$ is a facet of the set packing polytope if and only if H is the node set of a clique of G_I. ■

The size of the associated intersection graph G_I is very large to permit the generation of all the clique facets of VP. However, the special structure of the matrix A lends itself to a convenient decomposition of its intersection graph G_I for identifying all the cliques of G_I. Consider a subgraph G_I^q of G_I, corresponding to the block interval $q\in Q$. Hence, $G_I^q = \{V^q,\ E^q\}$, where $V^q = \{j \in N_i : i \in T_q\}$, and $E^q = \{(i, j) \in E : i, j \in V^q\}$. Note that $V = \bigcup_{q\in Q} V^q$. Let R_q be a clique of the subgraph G_I^q. Then,

$$\sum_{j\in R_q} x_j \leq 1 \tag{6.22}$$

is a valid inequality of GPP. Furthermore, we can obtain a complete subgraph corresponding to each GUB constraint of model GPP.

$$\sum_{j\in N_i} x_j \leq 1 \quad \forall i\in M_G \tag{6.23}$$

Proposition 6.9 Every clique facet of GPP belongs either to class (6.22) or to class (6.23).

Proof. Since the matrix A has a block diagonal structure, for any $j_1 \in V^{q_1}$, $j_2 \in V^{q_2}$, where $q_1 \neq q_2$, we have $a_{j_1}^T a_{j_2} \geq 1$, if j_1 and j_2 appear in some common constraint in M_G, and $a_{j_1}^T a_{j_2} = 0$, otherwise. Hence, the cliques of G_I, except for those of type (6.23), can only exist in some subgraph G_I^q, $q \in Q$. This completes the proof. ■

Proposition 6.9 asserts that we need to search only the subgraphs G_I^q, $q \in Q$, in order to generate all the clique facets of CHGPP. Although we have reduced the search space for generating all the clique inequalities, finding all the cliques within each G_I^q still remains a hard problem. However, we can devise a simple separation procedure to generate a strong valid cutting plane inequality that deletes off a fractional optimal solution of $\overline{GP}$ as follows.

Separation Procedure: Let $\bar{x}$ be a fractional optimal solution to $\overline{GP}$, so that $F = \{j: \bar{x}$ is fractional $\} \neq \phi$. If any $\bar{x}_j = 1$, all the vertices that are adjacent to the vertex j in the intersection graph cannot be packed, i.e., $\bar{x}_i = 0$, $\forall (i,j) \in E$. Hence, we can further reduce the size of the subgraph G_I^q for finding a clique that generates a strong valid cutting plane inequality. In order to implement this reduction, we first try to find a complete subgraph on the vertex set corresponding to the set of fractional variables $\bar{x}_j$ in the block q. If we find such a complete subgraph with vertex set H such that $\sum_{j \in H \cap F} \bar{x}_j > 1$, then we will have identified a clique inequality that cuts off $\bar{x}$. Subsequently, we can extend H to a maximal complete subgraph (clique R_q) of the subgraph G_I^q by examining the zero variables in V^q, and hence obtain a facetial cutting plane for GPP.

The following example illustrates the foregoing process for generating a strong valid cutting plane inequality from the facets of the set packing polytope.

Example 6.5 Consider the following problem that has three jobs to be scheduled on two machines, and has machine blocked-off duration times as specified below.

$$(r_{1,1},\ d_{1,1},\ \Delta_{1,1},\ w_{1,1}) = (0, 9, 2, 1),\ (r_{1,2} ,\ d_{1,2},\ \Delta_{1,2},\ w_{1,2}) = (0, 9, 2, 1),$$
$$(r_{2,1},\ d_{2,1},\ \Delta_{2,1},\ w_{2,1}) = (0, 9, 3, 1),\ (r_{2,2} ,\ d_{2,2},\ \Delta_{2,2},\ w_{2,2}) = (0, 9, 3, 1)$$
$$(r_{3,1},\ d_{3,1},\ \Delta_{3,1},\ w_{3,1}) = (0, 9, 4, 1),\ (r_{3,2} ,\ d_{3,2},\ \Delta_{3,2},\ w_{3,2}) = (0, 9, 4, 1)$$

The machine blocked-off times are as follows.

Machine 1: [3,5]

Machine 2: [4,6]

Machine 1 slots	1	2	3			4	5	6	7	
Machine 2 slots	8	9	10	11			12	13	14	
Time units	0	1	2	3	4	5	6	7	8	9

An optimal solution $\bar{x}$ to $\overline{GP}$ is given by $\bar{x}_{1,9} = 1/2$, $\bar{x}_{1,11} = 1/2$, $\bar{x}_{2,3} = 1$, $\bar{x}_{3,7} = 1/2$, $\bar{x}_{3,11} = 1/2$, and $\bar{x}_{ik} = 0$ otherwise. For the solution $\bar{x}$ to $\overline{GP}$, variables $x_{1,9}$, $x_{1,11}$, $x_{3,11}$ are fractional in block q=3, where T_q ={8, 9, 10, 11}. However, the indices of these variables $\{x_{1,9}, x_{1,11}, x_{3,11}\}$ themselves yield a complete subgraph H of the subgraph of G^P induced by the fractional variables of V^q. The node set of the extended complete subgraph of G^P, which includes H in its node set, corresponds to the indices of the variables $\{x_{1,9}, x_{1,10}, x_{1,11}, x_{2,10}, x_{2,11}, x_{3,11}\}$. Hence, the corresponding clique inequality of type (6.22) is $x_{1,9} + x_{1,10} + x_{1,11} + x_{2,10} + x_{2,11} + x_{3,11} \leq 1$, which cuts off the fractional optimal solution $\bar{x}$ to $\overline{GP}$. In addition, by augmenting $\overline{GP}$ with this additional clique inequality, we obtain an integer optimal solution, given by $x_{1,7} = 1$, $x_{2,3} = 1$, $x_{3,11} = 1$, and $x_{ik} = 0$ otherwise, which therefore solves problem GP.

7. *Approximate and Optimal Algorithms for Problem GP*

As mentioned in the previous chapter, the worst-case time complexity of finding a feasible solution to problem GP is NP-complete in the strong sense. Hence, we may not expect even a pseudo-polynomial-time algorithm for obtaining a feasible solution, unless P = NP. Even so, we begin below by devising a simple truncated implicit enumeration heuristic which appears to work quite well in practice. In particular, the special structures inherent in model GP enable us to construct several effective approximate and exact algorithms that provide solutions within specified tolerance of optimality. These algorithms are presented in turn below.

7.1. Enumeration, and Primal and Dual Linear Programming and Lagrangian Relaxation Based Heuristics for Problem GP

7.1.1. Truncated Enumeration Heuristic HGP

This truncated implicit enumeration heuristic exploits the GUB structure of the problem when extending a partial solution, and it attempts to construct a good quality feasible solution in a constructive depth-first enumeration fashion, truncating the process once such a feasible solution has been found. Moreover, due to the binary coefficient structure of the constraint matrix, computations such as checks for feasibility can be made more efficient by using a bit manipulation, or, logical ″AND″ and ″OR″ operations.

In the heuristic HGP described below, we use the following notation.

PS: Partial solution comprised of an index set x_j such that $x_j = 1$.

P_i : A set of variables corresponding to unassigned target i, which are forced to zero due to the partial schedule PS, i.e. , $P_i = \{j : j \in N_i \cap N_s$, for all $s \in M_k \cap M_P$, $\forall k \in PS\}$.

PBS(i): An index set of variables x_j for $j \in (N_i - P_i)$, corresponding to an unassigned target $i \in M_G$ such that putting $x_j = 1$ is known (by a previous attempt) to cause an infeasible partial schedule.

Heuristic HGP

INITIALIZATION Suppose that the targets are ordered and reindexed according to some suitable rule (see Remark 7.1 below). Each level t of the depth-first enumeration tree that is constructed then corresponds to the target t, for t=1,..,n. Select target t = 1, and let PS = ϕ, and PBS(i) = ϕ for all $i \in M_G$.

STEP 1. Determine the set P_t.

STEP 2. Find the index j_t of target t such that

$$j_t = \text{argmin} \{ c_j : j \in [N_t - (P_t \cup \text{PBS}(t))] \}$$

If j_t exists, then set PS = PS + $\{j_t\}$, and go to STEP 4. Otherwise, go to STEP 3.

STEP 3. (Backtracking) If t=1 (or equivalently, if PS = ϕ), then the given instance of problem GP is infeasible. Otherwise, perform the following steps.

```
    3.1. Set PBS(t) = $\phi$.
    3.2. Replace PS $\leftarrow$ PS - $\{j_{t-1}\}$.
    3.3. Replace PBS(t $-$ 1 ) $\leftarrow$ PBS( t $-$ 1 ) + $\{j_{t-1}\}$.
    3.2. Replace t $\leftarrow$ t - 1.
    3.5. Go to STEP 2.
```

STEP 4. (Stopping criterion) If t = $|M_G| \equiv n$, then stop; we have found a feasible solution to problem GP. Otherwise, set $t \leftarrow t + 1$ and go to STEP 1.

Remark 7.1 Naturally, heuristic HGP will find a feasible solution in a finite number of steps, if problem GP is feasible. For computational purposes, we may need to employ some premature termination criteria in order to practically avoid the exponential worst-case complexity of the heuristic HGP. For example, we can prescribe a number n_b for the maximum allowed number of backtrackings so that the heuristic HGP is terminated if it backtracks n_b times. In the same spirit, we can also set a maximum allowed limit τ_{HGP} on the run time for the heuristic HGP. Whenever the procedure backtracks, if the elapsed time exceeds τ_{HGP}, then the heuristic can be terminated. Furthermore, we can enhance its performance by rearranging the set of targets in the search for a feasible solution according to some rule which reflects the problem characteristics as governed by release times, deadlines, processing times, and the structure of the cost function. A set of possible alternative criteria for rearranging the targets are given below. (Based on some preliminary experience, we adopted the first of these in the present study.)

AR1: Arrange the variables x_j in ascending order of $\frac{c_j}{|M_j|}$.

AR2: Arrange the targets in ascending order of their release times. Break any ties by arranging the targets in descending order of ΔC_i, where $\Delta C_i = c_{j_2} - c_{j_1}$, and where $j_1 = \underset{j \in N_i}{\operatorname{argmin}}\{c_j\}$ and $j_2 = \underset{j \in (N_i - j_1)}{\operatorname{argmin}}\{c_j\}$, $i \in M_G$. The value of ΔC_i can be updated while dynamically assigning each target to a level as heuristic HGP progresses. Alternatively, we can directly arrange the targets according to a descending order of ΔC_i, using the early release time rule to break the ties, if necessary.

AR3 : Arrange the targets dynamically in ascending order of $|N_i - P_i|$. This rule can be expected to generate a smaller partial enumeration tree, and it facilitates finding a feasible solution quickly, but not necessarily a good quality solution .

Remark 7.2 We can also implement some local improvement strategies to be used in concert with heuristic HGP. Suppose that HGP produces some feasible solution $\tilde{x}$. Due to the block-diagonal structure of model GP, $\tilde{x}$ can be partitioned into separable blocks $q \in Q$, each comprised of certain assigned targets. We can then improve on the feasible solution $\tilde{x}$ by sequentially finding a better schedule for targets within each block $q \in Q$, or more generally, by considering suitable groups of blocks from Q. This can be done via an exact branch-and-bound algorithm, or via the use of some enhanced heuristic as described next.

7.1.2. Linear Programming Based Heuristic HGPLP

Despite the effectiveness of heuristic HGP, as will be demonstrated in the computational experience section, this procedure can be enhanced to yield very close to optimal solutions by using the linear programming relaxation $\overline{\text{GP}}$ of problem GP. The prescribed modification of HGP, based on an optimal solution to $\overline{\text{GP}}$, operates as follows.

Heuristic HGPLP

LINEAR PROGRAMMING STEP. Solve $\overline{GP}$, and let $\bar{x}$ be an optimal solution. If $\bar{x}$ is integral, then stop; $\bar{x}$ also solves problem GP. Otherwise, $\nu(\overline{GP})$ is a lower bound on $\nu(GP)$. Proceed to the Main Step.

MAIN STEP (Prioritized Search with HGP). Partition the set of targets into $M^+ = \{i \in M_G : \bar{x}_j = 1$ for some $j \in N_i\}$ and $M_G - M^+$. Order the targets separately within each of these two sets according to Remark 7.1. Then, implement heuristic HGP by starting with an advanced starting partial solution $PS = \{j : \bar{x}_j = 1\}$, and extending this partial schedule by using only the variables that are positive in the linear programming solution, that is, by replacing N_t with N_t^+ at Step 2 of HGP, where $N_t^+ = \{j \in N_t : \bar{x}_j > 0\}$, for $t \in M_G$. If a feasible solution is found, then stop with this as the prescribed heuristic solution for problem GP. Otherwise, implement HGP with an advanced partial solution $PS = \{j : \bar{x}_j = 1\}$, but now permitting the use of all variables in N_t for $t \in M_G$.

7.1.3. Lagrangian Dual Based Heuristic HGPLD

As will be seen in Chapter 8, the linear programming relaxation $\overline{GP}$ provides a very tight lower bound on problem GP, thereby yielding near optimal solutions via the heuristic HGPLP. However, solving $\overline{GP}$ via the simplex method is a major computational bottleneck because of the high degree of degeneracy inherent in the problem. (See Marsten (1974) for a similar experience in solving large-scale set partitioning problems.) To overcome this difficulty, one can perhaps solve the dual to $\overline{GP}$ using a package such as CPLEX (Bixby, 1991) that incorporates efficient pivot selection and stalling prevention strategies. An alternative is to exploit the special structure of $\overline{GP}$

within the context of a Lagrangian relaxation approach (see Fisher, 1981). Toward this end, let u_i, $i \in M_G$ be the Lagrange multiplier associated with the GUB constraints of model GP. Then, the corresponding Lagrangian relaxation of model GP, denoted by LRGP(u), can be constructed as follows.

LRGP(u): Minimize $$\sum_{i \in M_G} \sum_{j \in N_i} (c_j - u_i) x_j + \sum_{i \in M_G} u_i$$

subject to

$$\sum_{j \in N_i} x_j \leq 1 \quad \text{for } i \in M_P \tag{7.1}$$

$$x_j \in \{0, 1\}, \quad j \in N_{GP} \tag{7.2}$$

For any $u \in R^{|M_G|} \equiv R^n$, we have that v(LRGP(u)) is a lower bound on v(GP). The best choice of u is one which yields the greatest lower bound, and is given as an optimal solution to the following associated Lagrangian dual problem:

LDGP: $\underset{u \in R^n}{\text{Maximize}}$ v(LRGP(u)).

Since the constraint set (7.1) has an interval matrix structure by Proposition 6.3, as well as a block-diagonal structure, LRGP(u) is a separable shortest path problem on an acyclic digraph G. Consequently, for any $u \in R^n$ every feasible extreme point solution to LRGP(u) is integral. This integrality condition leads to the following result.

Proposition 7.1 $v(\overline{GP}) = v(\text{LDGP})$. ■

In order to solve the Lagrangian dual problem LDGP, we propose to use a conjugate subgradient optimization algorithm. Embedded within this procedure, we execute a heuristic scheme to actively search for improving feasible solutions to problem GP. Below, we present a generic statement of the resulting heuristic, denoted by heuristic HGPLD, and then address the design of suitable implementation strategies for its various steps that help accelerate the solution process.

Heuristic HGPLD

INITIALIZATION Let N_1 be the maximum number of iterations permitted, choose ε as some suitable termination tolerance, and initialize the iteration counter as $k=1$. Determine an initial feasible solution for problem GP via the heuristic HGP of Section 7.1.1, and let $\bar{z}$ be its objective value. Also, select an initial Lagrangian dual vector u^k, and denote by $\hat{z}$ the incumbent solution value of LDGP. (Initially, set $\hat{z} = -\infty$.) Proceed to Step 1.

STEP 1. (Subproblem Solution) Solve the separable shortest path problem LRGP(u^k). Let $\bar{x}^k$ be an optimal solution having an objective value $z^k = v(\text{LRGP}(u^k))$, and compute a subgradient ξ^k to the Lagrangian dual function $v(\text{LRGP}(u))$ at $u = u^k$ as $\xi_i^k = 1 - \sum_{j \in N_i} \bar{x}_j^k$, $i \in M_G$. If $z^k > \hat{z}$, then set $\hat{z} = z^k$ and go to Step 3. Otherwise, proceed to Step 2.

STEP 2. (Determination of New Dual Solution) Find a direction of motion d^k and select a step size λ^k. (Prescribed techniques for making these selections are addressed.) Update the Lagrange multipliers according to $u_i^{k+1} = u_i^k + \lambda^k d_i^k$ for each $i \in M_G$. If $k \geq N_1$, then terminate. Otherwise increment k by 1 and return to Step 1.

STEP 3. (Application of Heuristic HGP for Problem GP) Let $(\bar{u}_i^k, i \in M_{GP})$ be the complete dual solution corresponding to $(\bar{u}_i^k, i \in M_G)$, where the remaining dual values $\bar{u}_i^k, i \in M_P$ are obtained from the solution to LRGP$(\bar{u}^k)$. Define the reduced cost coefficients $\bar{c}_j = c_j - \sum\limits_{i \in M_j} \bar{u}_i^k$, $\forall j \in N_{GP}$, denote $M_G^+ = \{i \in M_G\colon \sum\limits_{j \in N_i} \bar{x}_j^k = 1\}$ and let $M_{\bar{G}} = M_G - M_G^+$. (Recall that $\bar{x}^k$ is a binary solution to the set packing constraints). If $M_{\bar{G}} = \phi$, then stop; $\bar{x}^k$ is an optimal solution for problem GP. Otherwise, order the targets separately within each of the two sets M_G^+ and $M_{\bar{G}}$ according to Remark 7.1, and concatenate these in the stated order. Then, implement heuristic HGP by starting with an advanced starting partial solution $PS = \{j : \bar{x}_j^k = 1 \text{ and } j \in N_i \text{ for some } i \in M_G^+\}$ where the targets in PS are arranged as they are in M_G^+. (Note that by its nature, the heuristic HGP will attempt to find a good completion to this starting partial solution, if one exists, failing which, it will backtrack and revise this partial solution accordingly.) If the objective value $\tilde{z}$ of the resulting solution is lesser than $\bar{z}$, then let this solution be the revised prescribed heuristic solution to GP and set $\bar{z} = \tilde{z}$. Now, if $\dfrac{(\bar{z} - \hat{z})}{\bar{z}} \leq \varepsilon$, then stop; the optimal solution value for GP is within $(1 - \varepsilon) \times 100\%$ of the current incumbent solution value. Also, if $||\xi^k||$ is insignificantly small, then stop; u^k is a (near) optimal solution to the Lagrangian dual problem LDGP. Otherwise, return to Step 2.

We now consider various algorithmic strategies related to the choice of the initial dual solution, the direction of motion, and the choice of step lengths. These are discussed in turn below.

Initial Starting Dual Solution

The procedure HD$\overline{\text{GP}}$ given below determines an advanced-start dual feasible solution to $\overline{\text{GP}}$, that can be used in the Initialization Step of the Lagrangian heuristic HGPLD. Note that defining u_i as the dual variable corresponding to constraint i, $i \in M_{GP}$, the dual to $\overline{\text{GP}}$ can be written as follows.

$$\text{D}\overline{\text{GP}}\text{: Maximize } \{ \sum_{i \in M_{GP}} u_i : \sum_{i \in M_j} u_i \leq c_j \text{ for } j \in N_{GP},$$

$$u_i : \text{unrestricted for } i \in M_G,\ u_i \leq 0 \text{ for } i \in M_P \} .$$

Dual Heuristic HD$\overline{\text{GP}}$

STEP 1. Put $u_i = \max \{c_j : j \in N_i\}$ for each $i \in M_G$, and let $u_i = 0$ for all $i \in M_P$.

STEP 2. Dualize the GUB constraints using the dual variables of Step 1, and let the resulting reduced costs be $R_j = c_j - u_{j(G)}$, where $j(G) = M_j \cap M_G$, for all $j \in N_{GP}$. Define a violation measure $V_i = \sum_{j \in \overline{N}_i} |R_j|$, where $\overline{N}_i = \{j \in N_i : R_j < 0\}$, for each $i \in M_P$.

STEP 3. Pick $t = \text{argmax} \{V_i : i \in M_P\}$. If $V_t = 0$, go to Step 5. Otherwise, put $u_t = \min \{R_j : j \in \overline{N}_t\}$.

STEP 4. Dualize the set packing constraint $t \in M_P$ using the dual variable u_t. Accordingly, replace R_j by $R_j - u_t$ for all $j \in N_t$. Hence, we can now replace $\overline{N}_i$ by $\overline{N}_i - \overline{N}_i \cap \overline{N}_t$ and V_i by $V_i - \sum_{j \in \overline{N}_i \cap \overline{N}_t} |R_j|$ for all $i \in M_P$. Return to Step 3.

STEP 5. Using the current dual solution, dualize the rows in M_P alone, and let the reduced costs be $R_j = c_j - \sum_{i \in M_P \cap M_j} u_i$ for all $j \in N_{GP}$. Put $u_i = \min \{R_j : j \in N_i\}$ for each $i \in M_G$, leave u_i, $i \in M_P$ as at present, and stop with the prescribed dual solution at hand.

Proposition 7.2 Heuristic HD$\overline{\text{GP}}$ generates a dual feasible solution for $\overline{\text{GP}}$ with an effort having complexity $O(m^2nT^2)$.

Proof. Whenever $V_t = 0$ at Step 3, it is readily verified that a dual feasible solution is at hand. Step 5 further improves on this dual solution by revising the dual variables associated with the constraints M_G, if possible. Moreover, the values of V_i are nonincreasing, and in each pass through Step 3, some additional V_i value is made zero. Hence, the heuristic finds a dual feasible solution in a finite number of steps. Since $|M_P|$ is $O(mT)$ and $|N_{GP}|$ is $O(mnT)$, the stated complexity of the dual heuristic is readily verified. This completes the proof. ∎

Remark 7.3 Due to the GUB structure of model GP, we can also easily obtain initial lower and upper bounds of the objective value $\nu(\text{GP})$. That is, $\sum_{i \in M_G} \min_{j \in N_i}\{c_j\} \leq \nu(\text{GP}) \leq \sum_{i \in M_G} \max_{j \in N_i}\{c_j\}$.

Choice of Direction of Motion

For selecting a direction of motion at Step 2 of the heuristic HGPLD, we adopt the following conjugate subgradient average direction strategy (ADS) of Sherali and Ulular (1989):

$$d^1 = \xi^1, \text{ and } \{d_i^k = \xi_i^k + \frac{||\xi^k||}{||d^{k-1}||} d_i^{k-1}, \qquad \forall i \in M_G\}, \qquad \text{for } k \geq 2 \tag{7.3}$$

where $||.||$ denotes the Euclidean norm. Note that the direction d^k simply bisects the angle between ξ^k and d^{k-1}, and in this sense, is an "average direction." Sherali and Ulular (1989) show that this direction has both a theoretical basis from a convergence viewpoint, and has an empirical computational advantage over the procedures of Held et al. (1974) and Camerini et al.(1975).

Choice of Step Sizes

The choice of step sizes to be used in the context of subgradient optimization is a very crucial issue. In practice, the selection of a proper step size λ^k which enables a near optimal incumbent solution to be obtained in a moderate number of iterations requires an elaborate fine tuning. According to Sherali and Ulular (1989) and similar to step sizes prescribed by Held et al. (1974), a suitable step size that can be used in concert with the ADS direction strategy is given by

$$\lambda^k = \beta^k \frac{(\bar{z} - z^k)}{||d^k||^2} \tag{7.4}$$

where $\bar{z}$ is the current best known objective value for problem GP, and β^k is a suitable parameter. Following the experience of Sherali and Ulular (1989), we use the following "block-halving" strategy in order to accelerate the convergence process. In this strategy, the iterations are partitioned into blocks based on an integer parameter N_2 such that at the top of each block when k mod(N_2) = 1, we compute the step length λ^k using the formula (7.3) and with $\beta^k = \bar{\beta}/2^{\lfloor k/N_2 \rfloor}$. Then, within each block, the step size is held constant, except that it is halved each time the process goes through some N_3 consecutive no improvements in the dual objective value. Additionally, each time the step length needs to be revised (recomputed via (7.4), or halved) over the N_1 iterations, the dual iterate is reset to the incumbent solution before using this modified step length. Typical parameter values prescribed are $N_1 = 200$, $N_2 = 75$, $N_3 = 5$, and $\overline{\beta} = 0.75$ We will refer to the combination of the ADS conjugate subgradient strategy and the foregoing block-halving step size strategy as *ADSBH* .

For the purpose of comparison, we attempted various combinations of conjugate and pure subgradient direction finding strategies along with various step-size rules. In our experience we found ADSBH to dominate the other combinations. (See Sherali and Lee, 1991.) Among the latter alternatives, a competitive combination turned out to be one which employs $d^k = \xi^k$ along with the block halving step size strategy. Because of the resemblance of this to Held et al.'s (1974) pure subgradient algorithm, we denote this combination by *HWCBH.* Chapter 8 reports on results using both ADSBH and HWCBH in the context of the Lagrangian heuristic HGPLD.

7.2. A Branch-and-Bound Algorithm

In this chapter, we develop a branch-and-bound algorithm based on bounds generated via a Lagrangian relaxation procedure, and a depth-first-search rule for selecting an active node in the branching process. At each node of this branch-and-bound procedure, we employ a suitable formulation to obtain a lower bound by solving the underlying linear programming relaxation, as well as to obtain an improved feasible solution. Below, we consider various algorithmic strategies related to the Lagrangian dual procedure, preprocessing rules, logical tests based on reduced costs, the development of a branching tree, the node selection rule employed, and the fathoming process.

Notationally, let x^b denote the current best primal feasible solution found thus far, and let UB denote the upper bound corresponding to x^b. Also, at each node k of the branching tree, let u^k denote the best dual feasible solution found to the continuous relaxation of the node subproblem, and let LB_k denote the lower bound corresponding to u^k.

7.2.1. Preprocessing Rules

Preprocessing rules can be quite useful in reducing the number of variables and constraints. These rules employ simple row and column scans to identify redundant rows, delete variables that can be fixed at 0 or 1, and to modify matrix coefficients. Below, we describe some simple preprocessing rules for detecting infeasibilities or reducing the size of the problem by identifying variables that can be fixed at 0 or 1. These rules are applied to the initial problem as well as to each node of the branching tree, and can be very effective in solving the problem.

Problem Decomposition Rule: Suppose that all the jobs are sorted in ascending order of $\min_{j\in M}\{r_{i,j}\}$. Define $\bar{h} = \min\{h\colon \max_{j\in M}\{d_{i,j}\} \le \min_{j\in M}\{ r_{h+1,j} \}, \text{for } 1 \le i \le h\}$, where $r_{n+1,j} \equiv \infty$. Then, we can decompose the set of jobs into sets N_1 and N_2, where $N_1 = \{1,\ldots, \bar{h}\}$ and $N_2 = N - N_1$. Treating N_2 as the original set of jobs N, this operation can be repeated with N_2. In this manner, we can obtain a set of separable problems that can be solved independently to obtain an optimal solution.

Problem Reduction Rule (Variable Fixing): If $|N_i| = 1$ for any $i \in M_{G,}$ then we can set the corresponding variable $x_j = 1$, delete x_j in the variable set, delete the set of constraints M_j from the model, and also put $x_k = 0$ for all $k \in N_t$, $t \in M_j$, $k \neq j$. In the same spirit, if for some $j \in N$, there exists an $i \in M_G$ $j \notin N_i$, such that $\bigcup_{k \in N_i} (M_k \cap M_P) \subseteq M_j \cap M_P$, then we can set $x_j = 0$ and delete this variable from the model.

An Implication of the Reduced Costs: At each node, we preform a variable-fixing procedure prior to any further branching. That is, at the root node, for example, we can delete from the model any variable $j \in N_P$ for which $\overline{c}_j^0 \; = c_j - \sum_{i \in M_j} u_i^0 > UB - LB_0$, since $x_j = 1$ cannot be optimal. Similarly, we can apply such a variable-fixing procedure at each node k based on the reduced costs corresponding to the dual variables u^k obtained for the associated node subproblem, in order to delete any variable from the corresponding subproblem whose reduced cost exceeds the bound gap $(UB - LB_k)$.

7.2.2. Development of the Branching Tree

We implement a depth-first-search rule to construct the branching tree. Accordingly, we denote by node k the currently active node at the kth level of the branching tree. At the kth level of the branching tree $(k \geq 0)$, the basic branching step selects a GUB constraint i(k) corresponding to job i(k) on which we have not as yet branched at a predecessor node on the chain to the root node, and creates a node for each $j \in N_{i(k)}$. We consider two rules for selecting a constraint. First, we implement a *fixed-order rule* for choosing a constraint as the branching constraint. That is, at the root node, a branching sequence is determined *a priori* in descending order of the values u_i^0, $i \in M_G$. As a second rule, we *dynamically* select a branching constraint i(k) at the kth level of the branching tree such that $i(k) \in \arg\max \{\, u_i^k : \;\; i \in$ (unbranched constraints in the chain from node k to the root node)} and where u^k is the best dual solution obtained for the most recent Lagrangian dual problem solved at level k. In Chapter 8, we report computational results using these rules for the selection of a branching constraint in the context of a branch-and-branch algorithm.

Having selected a branching constraint i(k) at the kth level of the branching tree, we create a node for each $j \in N_{i(k)}$, fixing $x_j = 1$ and $x_t = 0$ for all $t \in (N_{i(k)} - j)$ at each such node. These nodes for level (k+1) are stored in a linked list of open nodes, denoted by Λ, in descending order of the reduced costs $\bar{c}_j^k \equiv c_j - \sum_{i \in M_j} u_i^k$, to be explored further in a Last-In-First-Out (LIFO) fashion, where u^k is the best dual solution obtained for the Lagrangian dual problem solved at the immediate predecessor node at level k.

7.2.3. Subproblem Generation

Each node of the branch-and-bound tree defines a new scheduling problem of reduced size. Consider a partial solution FS(k) generated at node k at the kth level of the branching tree, obtained by fixing some variable $x_j = 1$, $j \in N_{i(s)}$, for each level s < k of the branching tree. Specifically, let FS(k) be the set of indices of variables x_j, $j \in N_{i(s)}$, s = 0,...,k-1, that are restricted to unity on the unique chain of the branching tree traced from node k to the root node. At the kth level of the branching tree, we generate for level (k+1) a set of subproblems P_s^{k+1}, for $s \in N_{i(k)}$, that can be described as follows.

$$\begin{aligned} P_s^{k+1}: \text{ Minimize}\{ \sum_{j\in N_{GP}} c_j x_j : & \sum_{j\in N_i} x_j = 1 \ \forall\ i\in M_G, \\ & \sum_{j\in N_i} x_j \leq 1 \ \forall\ i\in M_P, \\ & x_t = 1 \ \forall t\in FS(k),\ x_s = 1,\ x_j\in\{0, 1\} \ \forall j\in N_{GP}\} \end{aligned}$$

Note that each Problem P_s^k, for $s\in N_{i(k-1)}$, has the same structure as problem GP, but the size is reduced. Problem P_s^k can be further reduced by explicitly deleting zero variables and resulting trivial constraints. However, this reduction can be efficiently handled by implicitly deleting zero arcs and isolated nodes of the underlying acyclic digraph of problem GP. Obviously, some problem P_s^k is infeasible if $(M_P \cap M_s) \cap (\bigcup_{j\in FS(k)} M_j) \neq \phi$. These trivial infeasible problems are not stored in the list Λ of open nodes. Instead of storing an explicit representation of the formulation P_s^k, or for that matter, instead of even storing an explicit representation of Λ, we record minimal information that can be used to recover the list Λ, as well as recover for each node in the list Λ, the level of the node, the index set of the fixed variables at the node, and the (near) optimal dual solution of the immediate predecessor. Hence, when we (conceptually) retrieve an open node from Λ, we can efficiently construct an explicit representation of the corresponding subproblem. Moreover, note that the foregoing preprocessing procedures and the variable-fixing procedure are applied to each new reduced node subproblem created.

7.2.4. Node Selection Rule and Fathoming Process

Upon completion of a branching step, we scan the list Λ of open nodes according to the LIFO rule, and select a node to explore further. If there are no open nodes, the algorithm terminates, prescribing the best primal solution that we have found thus far.

Otherwise, we analyze the reduced problem corresponding to the selected node. If this problem is shown to be infeasible or is fathomed by comparing its lower bound to the incumbent upper bound, then this node can be discarded. Upon fathoming this node, we backtrack on Λ in a LIFO fashion and select another node for investigation. Otherwise, we branch further from this node. For each explored unfathomed node, of which there is one at each level, the corresponding (near) optimal dual solution (u_i^k, $i \in M_G$) is saved so that it can be used as an advanced starting solution for the immediate descendant nodes.

7.2.5. Algorithm GPBB

We now present the proposed detailed algorithmic steps for finding exact and/or approximate solutions for problem GP using the foregoing branch-and-bound procedure. This algorithm requires a specification of the following parameters:

1. ε = guaranteed optimality tolerance parameter. Hence, whenever $\frac{(UB - LB_k)}{UB} \leq \varepsilon$, we fathom node k. If $\varepsilon = 0$, the algorithm is an exact algorithm for finding an optimal integer solution to problem GP. On the other hand, if $0 < \varepsilon < 1$, the algorithm tries to find a feasible solution such that the optimal objective value of problem GP lies in the interval $[(1 - \varepsilon)UB, UB]$.

2. τ_{BB} = the maximum allowed run time. If the total run time of the branch-and-bound algorithm, checked at the top of the loop, exceeds τ_{BB}, then the algorithm is terminated, and the current best feasible solution is taken as the prescribed solution.

3. τ_{HGP} = the maximum allowed run time for the heuristic HGP, at the end of which, this procedure is aborted.

INITIALIZATION: Initialize the branching tree level of the root node as $k=0$. Determine an initial feasible solution x^b for problem GP via the heuristic HGP of Section 7.1.1, and let UB be its objective value. Also select an initial Lagrangian dual vector u^0 as $u_i^0 = \min_{j \in N_i}\{c_j\}$ $\forall$ $i \in M_G$, and denote by $LB_0 = \sum_{i \in M_p} u_i^0$ the incumbent solution value of LDGP. If $\frac{(UB - LB_0)}{UB} \leq \varepsilon$, for some tolerance $0 \leq \varepsilon < 1$, terminate with x^b as the prescribed solution. Else, proceed to Step 1.

STEP 1. (Root Node) Solve the Lagrangian dual problem (LDGP) of model GP via the conjugate subgradient optimization method described in Section 7.1.3 to find a revised (near) optimal dual solution u^0. Also denote by LB_0 the (near) optimal objective function value of LDGP. If $\frac{(UB - LB_0)}{UB} \leq \varepsilon$, then stop with x^b as an ε-optimal solution. Otherwise, execute the variable-fixing procedure using the reduced cost vector $\bar{c}^0$. Put $FS(k) = \phi$, and proceed to Step 2.

STEP 2. (Generation of Subproblems) Select the branching GUB constraint or job $i(k)$ as in Section 7.2.2. Let $J_{i(k)}$ = $\{j \in N_{i(k)} : M_j \cap M_l = \phi \ \ \forall \ l \in FS(k)\}$. (Note that if $J_{i(k)} = \phi$, then the subproblem at the preceding level would have been infeasible, even in the continuous sense.) Construct subproblems P_s^{k+1} for all $s \in J_{i(k)}$, and add these subproblems to the linked list Λ of open nodes, according to a descending order of the reduced costs $\bar{c}_s^k$. Increment k by one and proceed to Step 4.

STEP 4. (Run Time Check) Check the accumulated run time. If this time exceeds the limit τ_{BB}, then stop. Otherwise, proceed to Step 5.

STEP 5. (Problem Selection and Relaxation) Select the subproblem P_t^k which is the most recent member of the list Λ, and delete it from Λ. Let FS(k) = FS(k-1) + {t}. Solve the Lagrangian dual of Problem PS_t^k using an advanced starting dual solution available from its immediate predecessor node in the branch-and-bound tree. Let u^k be the best dual solution obtained of objective value LB_k. If $\frac{(UB - LB_k)}{UB} \leq \varepsilon$, then fathom this node and go to Step 7. Otherwise, proceed to Step 6.

STEP 6. (New Incumbent Solution by Heuristic HGP) Let $\bar{x}^k$ denote an optimal solution to the Lagrangian relaxation subproblem of Problem P_t^k, corresponding to the (near) optimal dual solution u^k obtained at Step 5. Denote $M_G^+ = \{i \in M_G : \sum_{j \in N_i} \bar{x}_j^k = 1\}$ and let $M_{\bar{G}} = M_G - M_G^+$. If $M_{\bar{G}} = \phi$, then $\bar{x}^k$ is an optimal solution for Problem $\bar{P}_t^k$. In addition, if $\bar{x}^k$ is integral, it is optimal to P_t^k, and so, if $LB_k < UB$, then let $UB = LB_k$, and $x^b = \bar{x}$, and go to Step 7. Otherwise, order the variables separately within each of the two sets M_G^+ and $M_{\bar{G}}$ in ascending order of $\frac{\bar{c}_j}{|M_j|}$ and concatenate these in the stated order. Then, implement heuristic HGP using this ordering of the jobs, and starting with an advanced starting partial solution $PS = \{j : \bar{x}_j^k = 1 \text{ for } j \in N_i, i \in M_G^+\}$, where the jobs in PS are arranged as they are in M_G^+. If HGP terminates before the time limit τ_{HGP} with an indication that the problem is infeasible, then fathom this node and go to Step 7. If the objective value of the solution produced by HGP is lesser than UB, then replace x^b by this improved heuristic solution to GP, and set $UB = cx^b$. If $\frac{(UB - LB_k)}{UB} \leq \varepsilon$, then fathom the current node and go to Step 7. Otherwise, return to Step 2.

STEP 7. (Fathoming Step and Termination Check) Replace $FS(k) \leftarrow FS(k) - \{t\}$. If $\Lambda = \phi$ or $\frac{(UB - LB_0)}{UB} \leq \varepsilon$, then stop; we have found an ε-optimal solution x^b to problem GP. Otherwise, let P_s^r be the most recent subproblem recorded in Λ. Put $k = r$, and $FS(k) = FS(k - 1)$. Return to Step 4. ■

8. *Computational Results*

In this section, we report on our computational experience using a set of test problems with size characteristics given in Table 8.1. The data for these problems are generated using guidelines provided by the Naval Surface Warfare Center (NSWC), Dahlgren, Virginia. Accordingly, the release times $r_{i,j}$ were generated uniformly on the interval [0,50], the processing times $\Delta_{i,j}$ were generated uniformly over [3,6], and the deadlines $d_{i,j}$ were computed as $d_{i,j} = r_{i,j} + \Delta_{i,j} + \delta_{i,j}$, where $\delta_{i,j}$ are generated uniformly over [0,3]. The number of blocked-off durations on each illuminator were generated uniformly over [0,10], and were then distributed uniformly over the time horizon $[0, \max_{i,j}\{d_{i,j}\}]$. Also, the weights $w \equiv (w_{ik}, k \in S_i, i \in N)$ were taken as either w^1 or w^2 specified below, again using NSWC guidelines.

$$w^1_{ik} = \left\lfloor \frac{\max_{p,q}(d_{p,q} - r_{p,q})^2}{(d_{i,\mu_k} - r_{i,\mu_k})^2} \right\rfloor, \quad \text{and} \quad w^2_{ik} = \left\lfloor \frac{10^5}{(d_{i,\mu_k} - r_{i,\mu_k})^2} \right\rfloor \quad \forall k \in S_i,\ i \in N \quad (8.1)$$

Recall that the objective function of problem GP seeks to minimize $\sum_{i \in N}\sum_{k \in S_i} w_{ik}\tau_k x_{ik}$. Hence, using (8.1), a preference is given to assigning target i to that slot k on illuminator μ_k for which not only does that engagement occur early in the time horizon (i.e., τ_k is small), but also the corresponding engageability time-window duration $(d_{i,\mu_k} - r_{i,\mu_k})$ is relatively large. The latter aspect is important in that if the strike is planned early in a relatively large time-window interval, then a possible miss or an unscheduled re-targeting decision provides sufficient opportunity for using the same illuminator on a subsequent attempt. Both the choices of weights in (8.1) reflect this bias, using either a relative or an absolute time-window interval term.

Remark 8.1 Although we report on computational results using only the weight functions (8.1), there are alternative objective functions that merit attention. For example, suppose that we can estimate the joint probability $f_{i,k}$ of hitting target i and inflicting damage, given that a missile is scheduled for engagement at the end of time τ_k using illuminator μ_k. (NSWC has an operational Force Level Engageability and Predicted Intercept Algorithm that is capable of computing the conditional probability of inflicting damage, given a hit, based on the predicted intercept angle.) Also, define a weight function W_{ik} which is a perceived reward function if target i is successfully hit (i.e., hit and damaged) at the end of time slot k. (Note that W_{ik} plays the inverse role of w_{ik} in the minimization objective of Problem GP.) Then, we can use the following objective function in problem GP.

$$\text{Maximize } \sum_{i \in N} \sum_{k \in S_i} W_{ik} f_{ik} x_{ik} \tag{8.2}$$

This objective function (8.2) seeks to maximize the total expected reward of successful strikes. If $W_{ik} = 1$, $\forall$ (i,k), then (8.2) maximizes the aggregate probability of successful (damaging) hits. The cost function (8.2) requires the specification of an appropriate weight function w_{ik}. One such function can be specified as follows.

$$W_{ik} = (d_{i,\mu_k} - \tau_k)^2 \tag{8.3}$$

The value of weight function (8.3) is proportional to the square of the interval between the actual and final strike times. Hence, this weight function emphasizes that the target should be (a) struck as early as possible, and (b) should preferably be assigned to an illuminator that has a larger engagement time-window, noting that $r_{i,\mu_k} \leq \tau_k \leq d_{i,\mu_k}$. While the weight function (8.3) is determined by the length of the engagement time-window and the earliness of τ_k within this interval, we would like the weight function to be also influenced by the location of this interval in the time horizon. Consequently, we can suggest a function

$$W_{ik} = \frac{(d_{i,\mu_k} - r_{i,\mu_k})^2}{\tau_k} \tag{8.4}$$

The weight function (8.4) reflects both the effect of the length of the engagement duration as well as that of hitting the target early in the time horizon, and therefore appears to be preferable. Note any appropriate positive exponent other than 2 can be used in (8.4). In fact, we can generally prescribe

$$W_{ik} = \frac{h_1(d_{i,\mu_k} - r_{i,\mu_k})}{h_2(\tau_k)}$$

where h_1 (.) and h_2 (.) are some suitable chosen monotone increasing functions. ■

The computational times reported below are all batch processing times in CPU seconds on an IBM 3090 series 300E computer, with coding in PASCAL, and include model generation and all input and output times. For the simplex runs, we used Marsten's (1984) XMP routines, employing the dual simplex option to overcome primal degeneracy difficulties.

8.1. Computational Results for Heuristics HGP and HGPLP

Table 8.2 presents the results obtained on applying heuristic HGP and HGPLP on the first ten test problems of Table 8.1, using the weight function w^1 as defined in (8.1). Observe that the solutions obtained by HGPLP are all guaranteed to be within 95-99% of optimality, with a major portion of the computational effort being extended in solving the linear programming relaxation. On the other hand, the heuristic HGP consumes only 1 % of the time required by HGPLP to produce fairly good, although not as good, solutions that lie within 85-97% of optimality.

8.2. Computational Results for Heuristic HGPLD

The motivation behind developing the Lagrangian relaxation based heuristic was to compromise between the relative quality of the solutions obtained, versus the effort expended, by the heuristics HGP and HGPLP. Indeed, as seen from the results given in Table 8.3, heuristic HGPLD provides a favorable compromise as desired. Here, the runs have been made on the same first ten problems of Table 8.1 using the weight function w^1, and using $\varepsilon = 0.05$, and then again with $\varepsilon = 0.01$. Note that HGPLD produces solutions of comparable, and often better, quality than does heuristic HGPLP, but is roughly five time faster than the latter method. Also, although ADSBH produces marginally better dual solutions than does HWCBH, the two strategies are practically indistinguishable. Table 8.4 presents results on applying Heuristic HGPLD to Problems 11 - 26 of Table 8.1, using the weight functions w^1 and w^2. For the sake of comparison, we also include the run times for the simplex algorithm (XMP) applied to these problems when using the weight function w^1. Note that the simplex run times are 5-10 times that for applying the entire heuristic HGPLD, while the conjugate subgradient strategy produces the same linear programming solution value for the underlying Lagrangian dual, also yielding a high quality heuristic solution. Hence, we prescribe the heuristic HGPLD with the ADSBH direction and step size strategy for implementation.

8.3. Computational Results for Algorithm GPBB

Tables 8.5 and 8.6 present computational results for Algorithm GPBB with the fixed-order branching rule and the dynamic branching rule respectively, using the first ten test problems of Table 8.1, along with weight function w^1 as defined in (8.1), and while using two optimality termination criteria, namely, $\varepsilon = 0.01$ and $\varepsilon = 0.005$. Note that the maximum number of subgradient iterations for the Lagrangian dual problem at the root node is set at 300, and the maximum number of such iterations at the other branching nodes is set at 30. The maximum run times permitted for these test problems are set at $\tau_{BB} = 30$ seconds and $\tau_{HGP} = 1$ second. Observe that the algorithm GPBB with the ADS strategy works better than the algorithm with the HWC strategy for these test problems, and for the most part, finds a solution within the prescribed optimality tolerance of 1% within 3-6 seconds of CPU time.

Table 8.1 Size Characteristics of the Test Problems

Problem	Number of Targets	Number of Illuminators	Number of Variables of Model GP	Number of Rows of Model GP
Problem 1	30	6	1015	159
Problem 2	40	8	1808	243
Problem 3	40	5	1581	347
Problem 4	45	5	1981	370
Problem 5	45	7	2684	460
Problem 6	50	5	1176	178
Problem 7	50	5	2196	387
Problem 8	50	5	2141	373
Problem 9	55	7	3275	502
Problem 10	60	8	4084	563
Problem 11	20	3	455	175
Problem 12	25	7	1583	456
Problem 13	30	5	1256	318

Table 8.1 (Continued) Size Characteristics of the Test Problems

Problem	Number of Targets	Number of Illuminators	Number of Variables of Model GP	Number of Rows of Model GP
Problem 14	35	7	2041	422
Problem 15	40	5	1768	355
Problem 16	40	6	2014	410
Problem 17	45	7	2684	460
Problem 18	45	8	3123	553
Problem 19	50	5	2141	373
Problem 20	50	8	3292	585
Problem 21	55	7	3275	502
Problem 22	55	8	3589	550
Problem 23	60	8	4084	563
Problem 24	60	7	3579	480
Problem 25	65	7	3809	523
Problem 26	65	8	4430	573

Table 8.2 Computational Results for HGP and HGPLP using Weight Function w[1].

Problem	HGP		XMP (Dual Simplex)		HGPLP		
	v(HGP)	Run time	$v(\overline{GP})$	Run time	v(HGPLP)	Run time	% Optimality
Problem 1	666	0.09	579.5	4.8	606	6.8	94.4%
Problem 2	943	1.02	854.7	13.5	862	12.6	99.3%
Problem 3	2070	0.09	2014.5	6.1	2018	7.21	99.8%
Problem 4	2210	0.18	2102.6	10.1	2209	12.2	94.1%
Problem 5	2125	0.19	2083.5	14.5	2090	16.5	99.6%
Problem 6	1111	0.09	998	14.5	1010	16.72	98.6%
Problem 7	2756	0.2	2567.5	12.44	2591	13.7	99.1%
Problem 8	2697	0.2	2525.8	12.1	2548	16.2	99.1%
Problem 9	2205	0.3	2150.8	33.15	2166	32.2	99.3%
Problem 10	2639	0.4	2592.0	32.2	2618	32.3	99.1%

Table 8.3 Computational Results for Heuristic HGPLD using Weight Function w^1.

PROBLEM	v(HGPLD)		v(LDGP)		Run time (unit=seconds)		% Optimality ($\epsilon = 0.05$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	606	606	577.9	569.7	1.7	0.9	94.2	94.1
Problem 2	880	881	843.5	840.1	2.4	1.5	94.7	94.2
Problem 3	2070	2070	1971.4	1971.4	0.2	0.2	94.0	94.0
Problem 4	2152	2168	2059.9	2090.9	1.8	2.9	94.5	96.3
Problem 5	2124	2124	2022.9	2033.2	0.6	0.6	94.0	94.6
Problem 6	998	1008	950.5	961.7	2.4	2.9	94.1	94.2
Problem 7	2640	2596	2524.4	2528.4	1.7	1.2	94.5	97.4
Problem 8	2580	2632	2461.2	2516.3	1.3	2.6	94.2	94.4
Problem 9	2187	2187	2099.2	2099.2	0.6	0.6	94.9	94.9
Problem 10	2639	2639	2515	2515	0.4	0.4	94.1	94.1

PROBLEM	v(HGPLD)		v(LDGP)		Run time (unit=seconds)		% Optimality ($\epsilon = 0.01$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	606	598	578.7	579.1	2.5	2.4	94.3	96.9
Problem 2	875	879	852.8	852.2	4.4	4.2	97.7	97.2
Problem 3	2028	2028	2010.4	2007.8	0.9	1.2	99.2	99.0
Problem 4	2142	2142	2102.1	2102.1	6.5	6.2	98.1	98.1
Problem 5	2092	2090	2078.0	2077.8	2.9	2.5	99.4	99.4
Problem 6	998	1002	988.1	982.8	4.2	6.5	99.1	98.3
Problem 7	2578	2596	2566.2	2566.2	4.8	7.5	99.6	98.9
Problem 8	2574	2584	2524.5	2524.4	7.1	6.3	98.1	97.7
Problem 9	2168	2170	2149.5	2148.2	7.8	2.2	99.2	99.0
Problem 10	2605	2610	2580.6	2582.5	2.1	2.6	99.1	99.0

Table 8.4 Further Computational Results for Heuristic HGPLD with $\varepsilon = 0.05$.

Problem	XMP (Weight function w^1)			ADSBH (Weight function w^1)				ADSBH (Weight function w^2)		
	$v(\overline{GP})$	Run time	Solution	Obj. Ratio#	v(HGPLD)	Run time	% Opt	v(HGPLD)	Run time	% Opt
Problem 11	1159	1.86	Integer	1.00	1177	0.51	96.1	258545	0.81	94.5
Problem 12	1017	4.01	Integer	1.00	1027	0.6	98.1	246253	0.5	98.1
Problem 13	1649	3.92	Integer	1.00	1671	0.81	94.5	367863	0.41	96.3
Problem 14	1703	7.39	Integer	1.00	1721	0.31	96.9	409773	0.31	952
Problem 15	2052.0	7.31	Integer	1.00	2099	0.32	94.2	470250	0.42	94.3
Problem 16	1649	7.31	Integer	1.00	1731	0.42	94.2	424815	0.38	94.4
Problem 17	2083.5	16.2	Fractional	1.00	2124	0.70	94.0	496986	0.43	98.0
Problem 18	2112	14.12	Integer	1.00	2163	0.2	94.6	483348	0.32	94.6
Problem 19	2524.8	14.9	Fractional	1.00	2580	1.5	94.2	590356	0.92	96.0
Problem 20	2112	20.1	Integer	1.00	2152	0.60	96.9	491452	0.4	96.7
Problem 21	2150.8	34.1	Fractional	1.00	2187	0.9	94.9	533928	0.87	94.9
Problem 22	2159.5	21.1	Fractional	1.00	2185	0.6	94.3	544533	0.41	94.1
Problem 23	2592.0	32.2	Fractional	1.00	2639	0.65	94.1	634310	0.55	94.8
Problem 24	3077	34.8	Integer	0.998	3143	2.71	94.5	770460	3.1	96.4
Problem 25	-	> 120	-	-	3199	0.5	94.1	753965	2.1	94.3
Problem 26	2875	32.63	Fractional	0.999	2929	2.4	96.1	703559	2.1	94.9

Obj. Ratio $= \dfrac{v(\text{LDGP})}{v(\overline{\text{GP}})}$, where v(LDGP) is obtained using the ADSBH strategy.

Table 8.5 Computational Results for Algorithm GPBB using the Fixed-Order Branching Rule along with Weight Function w^1 and $\varepsilon = 0.01$.

PROBLEM	Best UB found		v(LDGP) at root node		Run time/# nodes		% Optimality [1]($\varepsilon = 0.01$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	581	582	579.4	579.3	2.8/0	2.9/36	99.8	99.7
Problem 2	859	867	854.7	855.1	2.7/0	30.0/344	99.5	98.7
Problem 3	2020	2029	2015.5	2015.5	3.6/0	2.9/0	99.8	99.4
Problem 4	2142	2160	2102.4	2102.4	30.0/110	30.0/252	98.2	97.4
Problem 5	2086	2105	2082.8	2083.3	2.4/0	2.2/0	99.9	99.0
Problem 6	998	1000	988.1	987.1	3.0/0	30.0/34	99.1	98.8
Problem 7	2590	2579	2567.1	2567.3	4.5/0	19.2/143	99.1	99.6
Problem 8	2557	2552	2525.4	2525.7	30/136	16.2/81	98.8	99.1
Problem 9	2165	2171	2150.2	2150.3	3.1/0	2.9/0	99.4	99.1
Problem 10	2613	2592	2590.9	2591.8	3.6/0	11.2/12	99.2	99.0

[1] Optimality = 100 x [1-(Best UB - ν (LDGP))/ Best UB]

Table 8.5 (Continued) Computational Results for Algorithm GPBB using the Fixed-Order Branching Rule along with Weight Function w^1 and $\varepsilon = 0.005$.

PROBLEM	Best UB found		v(LDGP) at root node		Run time/# nodes		% Optimality ($\varepsilon = 0.005$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	581	582	579.3	579.3	2.8/0	1.5/9	99.8	99.7
Problem 2	859	860	854.7	855.1	2.7/0	16.0/153	99.5	99.5
Problem 3	2020	2029	2015.4	2015.5	3.5/0	6.7/98	99.8	99.5
Problem 4	2142	2160	2102.4	2102.3	30.0/175	30.0/227	98.1	97.2
Problem 5	2086	2087	2082.8	2083.3	2.4/0	2.8/0	99.9	99.9
Problem 6	998	1000	989.2	987.1	30.0/26	30.0/34	99.2	98.8
Problem 7	2576	2573	2567.2	2567.3	26/179	21.4/89	99.7	99.6
Problem 8	2547	2544	2525.4	2525.7	30.0/145	30.0/102	99.2	99.3
Problem 9	2160	2160	2150.6	2150.6	6.0/0	4.8/0	99.6	99.6
Problem 10	2601	2621	2591.8	2591.9	6.3/0	30.0/159	99.7	98.9

Table 8.6 Computational Results for Algorithm GPBB using the Dynamic Branching Rule along with Weight Function w^1 and $\varepsilon = 0.01$.

PROBLEM	Best UB found		v(LDGP) at root node		Run time/# nodes		% Optimality ($\varepsilon = 0.01$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	581	582	579.4	579.3	2.8/0	2.8/36	99.8	99.7
Problem 2	859	867	854.7	855.1	2.7/0	30.0/347	99.5	98.7
Problem 3	2020	2029	2015.5	2015.5	3.5/0	5.5/0	99.8	99.4
Problem 4	2142	2160	2102.4	2102.4	30.0/158	30.0/252	98.2	97.2
Problem 5	2088	2105	2082.8	2083.3	2.4/0	2.8/0	99.9	99.0
Problem 6	998	1000	988.1	987.1	4.9/0	30.0/34	99.1	98.8
Problem 7	2590	2579	2567.1	2567.3	4.5/0	19.2/143	99.1	99.6
Problem 8	2557	2552	2525.4	2525.7	30/141	16.1/176	98.8	99.0
Problem 9	2165	2171	2150.2	2150.3	3.1/0	2.9/0	99.4	99.1
Problem 10	2613	2617	2590.9	2591.9	3.6/0	12.0/12	99.2	99.0

Table 8.6 (Continued) Computational Results for Algorithm GPBB using the Dynamic Branching Rule along with Weight Function w^1 and $\varepsilon = 0.005$.

PROBLEM	Best UB found		v(LDGP) at root node		Run time/# nodes		% Optimality ($\varepsilon = 0.005$)	
	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH	ADSBH	HWCBH
Problem 1	581	582	579.3	579.3	2.8/0	2.2/9	99.8	99.7
Problem 2	859	861	854.7	855.1	2.7/0	30.0/247	99.5	99.4
Problem 3	2020	2029	2015.4	2015.5	3.5/0	19.9/235	99.8	99.5
Problem 4	2142	2160	2102.4	2102.3	30.0/175	30.0/260	98.1	97.4
Problem 5	2086	2087	2082.8	2083.3	2.4/0	2.8/0	99.9	99.9
Problem 6	998	1000	989.2	987.1	30.0/25	30.0/28	99.2	98.8
Problem 7	2588	2573	2567.2	2567.3	30.0/242	21.4/88	99.4	99.8
Problem 8	2550	2544	2525.4	2525.7	30.0/176	30.0/109	99.1	99.3
Problem 9	2160	2160	2150.6	2150.6	6.1/0	4.8/0	99.6	99.6
Problem 10	2601	2617	2591.8	2591.9	6.3/0	30.0/155	99.7	99.0

9. *An Extension of the General Discrete Data Problem*

In this chapter, we consider an extension of problem GP and provide a zero-one integer programming model for the extended problem. Suppose that there are n targets, indexed $i \in N \equiv \{1,...,n\}$, m illuminators, indexed $j \in M \equiv \{1,...,m\}$, and that there are p launchers, indexed $l \in L \equiv \{1,...,p\}$. Each target i has a specified availability duration $[r_{i,j,l}, d_{i,j,l}]$, which depends on illuminator j and launcher l, and each target must be processed only during such an interval, depending on its assignment. Moreover, if target i is assigned to illuminator j, then it must be processed for a given duration $\Delta_{i,j}$ without interruption, prior to its scheduled engagement time $t_{i,j}$. In addition, there may be certain specified blocks of duration for which each illuminator might be unavailable. Accordingly, a feasible schedule is comprised of a set of illuminator durations for each target, along with an assignment of this duration to a particular illuminator, so that the target processing time intervals over the set of targets assigned to each illuminator are all disjoint. Among such feasible schedules, we are interested in finding a schedule which minimizes the total weighted engagement time. We denote this *extended problem* by *problem EGP* .

We assume that the processing times $\Delta_{i,j}$, and the target availability interval (time-window) endpoints $r_{i,j,l}$ and $d_{i,j,l}$, as well as the end points of each unavailability interval duration on each of the illuminators $j \in M$, are all nonnegative integer valued data. Under this assumption, letting $T \equiv \underset{i,j,l}{\text{maximum}}\{d_{i,j,l}\}$ be the problem time horizon given by the latest due-date, the interval [0,T] may be decomposed into slots of unit duration, distributed over all the illuminators. Let $k \in K = \{1,...,K\}$ index all the available or unassigned slots over all the illuminators.

To formulate problem EGP, let τ_k be the upper interval end-point of slot k, for $k \in K$. For each target $i \in N$ and each slot $k \in K$, define $L_{ik} = \{l \in L : [\tau_k - \Delta_{i,\mu_k}, \tau_k] \subseteq [r_{i,\mu_k,l}, d_{i,\mu_k,l}]\}$. We also define for each target $i \in N$, a set

$$S_i = \{k \in K : [\tau_k - \Delta_{i,\mu_k}, \tau_k] \subseteq [r_{i,\mu_k,l}, d_{i,\mu_k,l}] \text{ and } (\tau_k - \Delta_{i,\mu_k}, \tau_k) \bigcap [\text{unavailability intervals for illuminator } \mu_k] = \phi, \text{ for some launcher } l \in L\}$$

where μ_k is the index of the illuminator that corresponds to slot k. Note that the set S_i denotes the slots $k \in K$ for which target i can be feasibly scheduled (independently) to complete by time τ_k. Accordingly, we define the binary decision variables $x_{ik} = 1$ if target i is scheduled to complete at the end of slot k, and $x_{ik} = 0$ otherwise. Observe that each slot k corresponds to some illuminator $\mu_k \in M$, and hence if $x_{ik} = 1$, then target i is implicitly being assigned to illuminator μ_k. Moreover, the cost in the objective function for this assignment is $c_{ik} = \min_{l \in L_{ik}}\{w_{ikl}\}\tau_k$ for $i \in N$, $k \in S_i$, where w_{ikl}, for $k \in S_i$, $i \in N$, are some positive weights, and so, x_{ik} implies that the striking launcher is the one that determines the cost c_{ik}. Furthermore, in order to ensure that the schedule of the n targets do not overlap on any of the illuminators, let us define for each slot $k \in K$, the set $J_k = \{(i, \rho) : i \in N, \rho \in S_i, \text{ and } [\tau_k - 1, \tau_k] \subseteq [\tau_\rho - \Delta_{i,\mu_k}, \tau_\rho]\}$. Note that for each $k \in K$, J_k is the set of combinations (i,ρ) such that slot k will be occupied if $x_{i\rho} = 1$. Then, problem EGP can be formulated as follows.

EGP: Minimize $\sum_{i \in N} \sum_{k \in S_i} c_{ik} x_{ik}$

subject to

$$\sum_{k \in S_i} x_{ik} = 1 \quad \text{for } i \in N$$

$$\sum_{(i,\rho)\in J_k} x_{i\rho} \leq 1 \quad \text{for } k\in K$$

$$x_{ik} \in \{0, 1\} \quad \text{for } k\in S_i,\ i\in N$$

Remark 9.1 Observe that model EGP has a structure that is identical to that of model GP. Hence, we can implement all the results and algorithms developed for model GP in order to solve the problem EGP. However, since model EGP does not incorporate any constraint that limits the number of assignments made to any given launcher, an optimal solution to model EGP might cause a load balancing problem on the launchers in practice. For example, launcher l may be scheduled to fire several (up to m) SAM's which are to strike their respective targets at the same time, although each target $i\in N$ is scheduled to be illuminated by some distinct illuminator $j\in M$. In order to overcome any conflict that such a solution might entail, some launcher-based restriction needs to be included in model EGP. This topic will be considered as a future research task.

10. Summary and Recommendations for Further Research

We have constructed a strong zero-one integer formulation for problem GP, and have devised an effective Lagrangian relaxation based heuristic that finds good quality solutions with an effort that is well within acceptable standards. Furthermore, using these tight lower and upper bounds for problem GP, a branch-and-bound algorithm has been developed that generates solutions guaranteed to be within 99% of optimality in 3-6 seconds of CPU time for 90% of the test problems solved. Although for the test problems we did not find it necessary to generate additional strong valid inequalities, we have shown via examples how strong valid inequalities can be generated by exploiting the structure of problem GP in order to tighten its linear programming relaxation. Such inequalities can be employed within the framework of the Lagrangian relaxation procedure to derive improved heuristics and exact procedures aimed at hard instances of this problem. In particular, due to the inherent theoretical complexity of problem GP, we might encounter a very hard problem that weakens the effectiveness of the heuristic HGPLD and consequently deteriorates the performance of algorithm GPBB. For example, algorithm GPBB fails to find a 99% optimal solution for test Problem 4 in Table 8.5 within 100 CPU seconds. This phenomenon motivates an enhanced branch-and-bound algorithm that implements strong valid inequalities developed in Section 6.3 and Chapter 2. Especially, such valid inequalities in Chapter 2 obtained by the Reformulation-Linearization-Technique procedure tighten the formulation of problem GP, and are viable candidates to be used in the framework of a suitable branch-and-cut procedure. In the same spirit, an improved Lagrangian dual heuristic may be incorporated for finding a good quality primal feasible solution to hard test problems, through an implementation of these strong valid inequalities.

References

W.P. Adams and P.M. Dearing, "On the equivalence between roof duality and Lagrangian duality for unconstrained 0-1 quadratic programming problems," *Report No. URI-061* Clemson University, Clemson, SC (1988).

W.P. Adams and H.D. Sherali, "A tight linearization and an algorithm for zero-one quadratic programming problems," *Management Science* 32 (1986), 1274-1290.

W.P. Adams and H.D. Sherali, "Linearization strategies for a class of zero-one mixed integer programming problems," *Operations Research* 38 (1990), 217-226.

W.P. Adams, A. Billionnet, and A. Sutter, "Unconstrained 0-1 optimization and Lagrangian relaxation," *Discrete Applied Mathematics* 29 (1991), 131-142.

A. Bachem and M. Grotschel, "New aspects of polyhedral theory," in B. Korte, ed., *Modern Applied Mathematics, Optimization and Operations Research,* North Holland, (1982), 51-106.

A. Bachem, M. Grotschel, and B. Korte, ed., *Mathematical Programming: The State of the Art* (Springer, 1983).

K.R. Baker, *Introduction to Sequencing and Scheduling* (Wiley, New York, 1974).

E. Balas, "Facets of the knapsack polytope," *Mathematical Programming* 8 (1975), 146-164.

E. Balas, "Some valid inequalities for the set partitioning problem," *Annals of Discrete Mathematics* 1 (1977), 13-47.

E. Balas, "Finding out whether a valid inequality is facet defining," *MSSR No. 558* Management Science Research Group, GSIA, Carnegie Mellon University, Pittsburgh, PA (1990)

E. Balas and J.B. Mazzola, "Nonlinear 0-1 programming: I. linearization techniques," *Mathematical Programming* 30 (1984), 1-21.

E. Balas and J.B. Mazzola, "Nonlinear 0-1 programming: II. dominance relations and algorithms," *Mathematical Programming* 30 (1984), 22-45.

E. Balas and S.M. Ng, "On the set covering polytope: II. lifting the facets with coefficients in {0,1,2}," *Mathematical Programming* 45 (1989), 1-20.

E. Balas and M. Padberg, "On the set covering problem," *Operations Research* 20 (1972), 1152-1161.

E. Balas and M. Padberg, "On the set covering problem: II. an algorithm for set partitioning," *Operations Research* 23 (1975), 3 74-90.

E. Balas and M. Padberg, ″Set partitioning: a survey,″ *SIAM Review* 18 (1976), 710-760.

E. Balas and C. Yu, ″Finding a maximum clique in an arbitrary graph,″ *SIAM J. Computing* 15 (1986), 1054-1068.

E. Balas and E. Zemel, ″Critical cutsets of graphs and canonical facets of set-packing polytopes,″ *Mathematics of Operations Research* 2 (1977), 15-19.

E. Balas and E. Zemel, ″Facets of the knapsack polytope from minimal covers,″ *SIAM J. APPL. MATH.* 34 (1978), 119-148.

E. Balas and E. Zemel, ″An algorithm for large zero-one knapsack problems,″ *Operations Research* 28 (1980), 1130-1145.

E. Balas and E. Zemel, ″Lifting and complementing yields all the facets of positive zero-one polytopes,″ *Mathematical Programming* (Elsevier Science Publishing Co., 1984), 13-24.

E. Balas, S. Ceria, and G. Cornuejols, ″A lift-and-project cutting plane algorithm for mixed 0-1 programs,″ *MSSR No. 576* GSIA, Carnegie Mellon University, Pittsburgh, PA (1991).

J.L. Balintify, 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 (1978), 63-76.

F. Barahona, ″The maxcut problem on graphs not contractible to K_5,″ *Operations Research Letters* 2 (1983), 107-111.

F. Barahona and A.R. Mahjoub, ″On the cut polytope,″ *Mathematical Programming* 36 (1986), 137-173.

F. Barahona, M. Grotschel, and A. R. Mahjoub, ″Facets of the bipartite subgraph polytope,″ *Mathematics of Operations Research* 10 (1985), 340-358.

F. Barahona, M. Junger, and G. Reinelt, ″Experiments in quadratic 0-1 programming,″ *Mathematical Programming* 44 (1989), 127-137.

M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali, *Linear Programming and Network Flows* (Wiley, New York, 1990).

A. Billionnet and B. Jaumard, ″A decomposition method for minimizing quadratic pseudo-boolean functions,″ *Operations Research Letters* 8 (1989), 161-163.

R.E. Bixby, ″Two applications of linear programming,″ *Workshop on Parallel Computing of Discrete Optimization Problems*, Minneapolis, MN (May, 1991).

E. Boros and P.L. Hammer, ″Cut-polytopes, boolean quadric polytopes and nonnegative quadratic pseudo-boolean functions,″ *DIMACS Technical Report 90-37* (1990).

E. Boros, Y. Crama, and P.L. Hammer, "Upper bounds for quadratic 0-1 maximization problems," *Operations Research Letters* 9 (1990), 73-79.

E. Boros, Y. Crama, and P.L. Hammer, "Chvatal cuts and odd cycle inequalities in quadratic 0-1 optimization" *SIAM J. Disc. Math.* 5 (1992), 163-177.

J. Bourjolly, P.L. Hammer, and B. Simeone, "Node weighted graphs having the Konig-Egervary property," *Mathematical Programming Study* 22 (1984), 44-63.

S.C. Boyd and W.R. Pulleyblank, "Facet generating techniques," *TR-91-31* Department of Computer Science, University of Ottawa, Ottawa (1991).

D.D. Boyer, E.L. Perry, and E.L. Price, "Force level control in naval anti-air warfare," *Proceedings of the Symposium on Command and Control Research* Naval Postgraduate School, Monterey, California, (June, 1990).

G.H. Bradley, "Transformation of integer programs to knapsack problems," *Discrete Mathematics* 1 (1971), 29-45.

G.H. Bradley, P.L. Hammer, and L. Wolsey, "Coefficient reduction for inequalities in 0-1 variables," *Mathematical Programming* 7 (1974), 263-282.

P.M. Camerini, L. Fratta, and F. Maffioli, "On improving relaxation methods by modified gradient techniques," *Mathematical Programming Study* 3 (1975), 26-34.

R. Carraghan and P.M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters* 9 (1990), 375-382.

T.J. Chan and C. A. Yano, "A multiplier adjustment approach for the set partitioning problem," *Operations Research* 40 (1992), S40-S47.

V. Chvatal, "On certain polytopes associated with graphs," *J. of Combinatorial Theory (B)* 18 (1975), 138-154.

G. Cornuejols and A. Sassano, "On the 0,1 facets of the set covering problem," *Mathematical Programming* 43 (1989), 43-57.

H. Crowder, E.L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research* 31 (1983), 803-834.

M. Deza and M. Laurent, "Facets for the complete cut cone," *Research Memorandum* Dept. of Math. Eng. and Inst. Physics, Univ. of Tokyo (1988).

M. Deza and M. Laurent, "Facets for the complete cut cone II: clique-web inequalities," *Document 57, LAMSADE* Universite de Paris Dauphine (1989).

M. Deza and M. Laurent, "New results on facets of the cut cone," *Research Report. B-227* Tokyo Institute of Technology (November, 1989).

M. Deza, K. Fukuda, and M. Laurent, "The inequicut cone," *Research Report 89-04* University of Tsukuba, (1989).

B.L. Dietrich and L.F. Escudero, "Coefficient reduction for knapsack-like constraints in 0-1 problems with variable upper bounds," *Operations Research Letters* 9 (1990), 9-14.

S.E. Elmaghraby, "The knapsack problem with generalized upper bounds," *European Journal of Operations Research* 38 (1989), 242-254.

J. Etcheberry, "The set covering problem: a new implicit enumeration algorithm," *Operations Research* 25 (1977), 760-772.

M.L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management Science* 27 (1981), 1-18.

M.L. Fisher and P. Kedia, "Optimal solution of set covering/partitioning problem using dual heuristics," *Management Science* 36 (1990), 674-688.

M.L. Fisher, W.D. Northup, and J.F. Shapiro, "Using duality to solve discrete optimization problems: theory and computational experience," *Mathematical Programming Study* 3 (1975), 56-94.

M.R. Garey and D.S. Johnson, "Two-processor scheduling with start times and deadlines," *SIAM J. Computing* 6 (1977), 416-426.

M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman and Company, New York, 1979).

R.S. Garfinkel and G.L. Nemhauser, "The set partitioning problem: set covering with equality constraints," *Operations Research* 17 (1969), 848-856.

R.S. Garfinkel and G.L. Nemhauser, "Optimal set covering: a survey," in Geoffrion, ed. *Perspectives on Optimization* (Addison-Wesley, Reading, Mass, 1972).

J. Grabowski, E. Nowicki, and S. Zdrzalka, "A block approach for single machine schedule with release dates and due dates," *European Journal of Operational Research* 26 (1986), 278-285.

E.S. Gottlieb, "Facets defining inequalities for the dichotomous knapsack problem," *Operations Research Letters* 9 (1990), 21-29.

E.S. Gottlieb and M.R. Rao, "Facets of the knapsack polytope derived from disjoint and overlapping index configurations," *Operations Research Letters* 7 (1988), 95-100.

E.S. Gottlieb and M.R. Rao, "The generalized assignment problem: valid inequalities and facets," *Mathematical Programming* 46 (1990), 31-52.

E.S. Gottlieb and M.R. Rao, "(1,k)-configuration facets for the generalized assignment problem," *Mathematical Programming* 46 (1990), 53-60.

P.L. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation, and persistency in quadratic 0-1 optimization," *Mathematical Programming* 28 (1984), 121-155.

P.L. Hammer, E.L. Johnson, and U.N. Peled, "Facets of regular 0-1 polytopes," *Mathematical Programming* 8 (1975), 179-206.

P. Hansen, "Methods of nonlinear 0-1 programming," *Annals of Discrete Mathematics* 5 (1979), 53-71.

M. Held, P. Wolfe, and H.P. Crowder, "Validation of subgradient optimization," *Mathematical Programming* 6 (1974), 62-88.

K. Hoffman and M. Padberg, "LP-based combinatorial problem solving," *Annals of Operations Research* 4 (1985), 145-194.

K. Hoffman and M. Padberg, "Improving LP-representation of zero-one linear programs for branch-and-cut," *ORSA Journal on Computing* 3 (1991), 121-134.

E.L. Johnson and M. Padberg, "A note on the knapsack problem with special ordered sets," *Operations Research Letters* 1 (1981), 18-22.

E.L. Johnson, M.M. Kostreva, and U.H. Suhl, "Solving 0-1 integer programming problems arising from large scale planning models," *Operations Research* 33 (1985), 803-819.

M. Laurent, "A generalization of antiwebs to independence systems and their canonical facets," *Mathematical Programming* 45 (1989), 97-108.

E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds, *The traveling salesman problem: a guided tour of combinatorial optimization* (Wiley, New York, 1985).

V.J. Leon and S.D. Wu, "On scheduling with ready-times, due-dates, and vacations," *Naval Research Logistics* 39 (1992), 53-65.

L. Lovasz and A. Schrijver, "Cones of matrices and set functions, and 0-1 optimization," *Working Paper BS-R8925,* Centrum voor wiskunde en Informatica (1989).

R.E. Marsten, "An algorithm for large set partitioning problems," *Management Science* 20 (1974), 774-787.

R.E. Marsten, "XMP: a structured library of subroutines for experimental mathematical programming," *ACM Transactions on Mathematical Software* 7 (1981), 481-497.

R.E. Marsten, M.R. Muller, and C.L. Killion, "Crew planning at flying tiger: a successful application of integer programming," *Management Science* 25 (1981), 1175-1183.

R.E. Marsten and F. Shepardson, "Exact solution of crew scheduling problems using the set partitioning model: recent successful applications," *Networks* 11 (1981), 165-177.

S. Martello and P. Toth, *Knapsack Problems* (John Wiley and Sons, 1990).

R.K. Martin and L. Schrage, ″Subset coefficient reduction cuts for 0-1 mixed integer programming,″ *Operations Research* 33 (1985), 505-526.

R.M. Nauss, ″The 0-1 knapsack problem with multiple choice constraints,″ *European Journal of Operational Research* 2 (1979), 125-131.

G.L. Nemhauser and L.E. Trotter, ″Properties of vertex packing and independence system polyhedra, ″ *Mathematical Programming* 6 (1974), 48-61.

G.L. Nemhauser and L.E. Trotter, ″Vertex packings: structural properties and algorithms,″ *Mathematical Programming* 8 (1975), 232-248.

G.L. Nemhauser and G.M. Weber, ″Optimal set partitioning, matchings and lagrangean duality,″ *Naval Logistics Quarterly* 26 (1979), 553-563.

G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (John Wiley and Sons, New York, 1988).

M. Padberg, ″On the facial structure of set packing polyhedra,″ *Mathematical Programming* 5 (1973), 199-215.

M. Padberg, ″A note on zero-one programming,″ *Operations Research* 23 (1975), 833-837.

M. Padberg, ″Covering, packing and knapsack problems,″ *Annals of Disc. Math.* 4 (1979), 265-287.

M. Padberg, ″(1,k)-configurations and facets for packing problems,″ *Mathematical Programming* 18 (1980), 94-99.

M. Padberg, ″The boolean quadric polytope; some characteristics, facets and relatives,″ *Mathematical Programming* 45 (1989), 139-172.

M. Padberg and G. Rinaldi, ″Optimization of a 532-city traveling salesman problem by branch-and-cut,″ *Operations Research Letters* 6 (1987),1-8.

M. Padberg and G. Rinaldi, ″A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems,″ *SIAM REVIEW* 33 (1991), 60-100.

R.G. Parker and R.L. Rardin, *Discrete Optimization* (Academic Press, New York, 1988).

J. Picard and M. Queyranne, ″On the integer-valued variables in the linear vertex packing problem,″ *Mathematical Programming* 12 (1977), 97-101.

C.N. Potts, ″Analysis of a heuristic for one machine sequencing with release dates and delivery times,″ *Operations Research* 28 (1980), 1436-1441.

W.R. Pulleyblank, "Polyhedral combinatorics," in Bachem, Grotschel and Korte, (1983), 312-345.

H.M. Salkin, *Integer Programming* (Addison-Wesley, Reading, Mass, 1975).

S.C. Sarin, S. Ahn, and A.B. Bishop, "An improved branching scheme for the branch and bound procedure of scheduling n jobs on m parallel machines to minimize total weighted flowtime," *International Journal of Production Research* 26 (1988), 1183-1191.

A. Schrijver, "Polyhedral proof methods in combinatorial optimization," *Discrete Applied Mathematics* 14 (1986), 111-133.

A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, 1986).

J.F. Shapiro, "A survey of Lagrangian techniques for discrete optimization," *Annals of Discrete Mathematics* 5 (1979), 113-138.

H.D. Sherali, "Bounds on penalties for dummy arcs in transportation networks," *European Journal of Operational Research* 36 (1988), 353-359.

H.D. Sherali and W.P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM J. DISC. MATH* 3 (1990), 411-430.

H.D. Sherali and W. Adams, "A hierarchy of relaxation and convex hull characterizations for mixed-integer zero-one programming problems," *Research Report* Virginia Polytechnic Institute and State University Department of Industrial and Systems Engineering, 1989.

H.D. Sherali and Y. Lee, "Models and algorithms for scheduling illuminators in surface-to-air defense strategies," *Research Report* Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University (September 1990).

H.D. Sherali and Y. Lee, "Models and algorithms for passive homing device operations in battle group anti-air warfare scenarios," *Research Report* Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University (September 1991).

H.D. Sherali and D.C. Myers, "Dual formulations and subgradient optimization strategies for linear programming relaxations of mixed-integer programs," *Discrete Applied Mathematics* 20 (1989), 51-68.

H.D. Sherali and O. Ulular, "A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems," *Applied Mathematics and Optimization* 20 (1989), 193-223.

C.D. Simone, "The cut polytope and the boolean quadric polytope," *Discrete Mathematics* 79 (1989), 71-75.

C.D. Simone, ″Lifting facets of the cut polytope,″ *Operations Research Letters* 9 (1990), 341-344.

C.D. Simone, M. Deza, and M. Laurent, ″Collapsing and lifting for the cut polytope,″ *Research Report R. 265 CNRS* (1989).

B. Simons, ″On scheduling with release times and deadlines,″ in M.A.H. Dempster et al., ed., *Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approach to Scheduling Problems* (D. Reidel Co. Hingham, 1981) 75-88.

B. Simons, ″Multiprocessor scheduling of unit-time jobs with arbitrary release times and deadlines,″ *SIAM J. Computing* 12 (1983), 294-299.

B. Simons and M.K. Warmuth, ″A fast algorithm for multiprocessor scheduling of unit-length jobs,″ *SIAM J. Computing* 18 (1989), 690-710.

P. Sinha and A.A. Zoltners, ″The multiple-choice knapsack problem,″ *Operations Research* 27 (1979), 503-515.

J.P. Sousa and L.A. Wolsey, ″A time index formulation on non-preemptive single machine scheduling problem,″ *Mathematical Programming* 54 (1992) 353-367.

L.E. Trotter, ″A class of facet producing graphs for vertex packing polyhedra,″ *Discrete Mathematics* 12 (1975), 373-388.

T.J. Van Roy and L.A. Wolsey, ″Solving mixed integer programs by automatic reformulation,″ *Operations Research* 35 (1987), 45-57.

L.A. Wolsey, ″Faces for a linear inequality in 0-1 variables,″ *Mathematical Programming* 8 (1975), 165-178.

L.A. Wolsey, ″Facets and strong valid inequalities for integer program,″ *Operations Research* 24 (1976), 367-372.

L.A. Wolsey, ″Strong formulations for mixed integer programming: a survey,″ *Mathematical Programming* 45 (1989), 173-191.

L.A. Wolsey, ″Valid inequalities for 0-1 knapsacks and MIPs with generalized upper bound constraints,″ *Discrete Applied Mathematics* 29 (1990), 251-261.

E. Zemel, ″Lifting the facets of zero-one polytopes,″ *Mathematical Programming* 15 (1978), 268-277.

E. Zemel, ″The linear multiple choice knapsack problem,″ *Operations Research* 28 (1980), 1412-1423.

E. Zemel, ″An O(n) algorithm for the linear multiple choice knapsack problem and related problems,″ *Information Processing Letters* 18 (1984), 123-128.

E. Zemel, ″Easily computable facets of the knapsack polytope,″ *Mathematics of Operations Research* 14 (1989), 760-764.

Vita

Youngho Lee, born on August 30, 1960 in Seoul, Korea, received B.S. and M.S. degrees in Industrial Engineering from Seoul National University, Seoul, Korea in 1984 and 1986, respectively. He worked for Samsung Hewlett-Packard Company for two years as a principal consultant on management information systems. In August 1988, he entered Virginia Polytechnic Institute and State University and began his Ph.D. program under the advice of Professor H.D. Sherali in the area of mathematical programming. Throughout four years of study, he has written several papers and worked as a research assistant to solve naval defense problems funded by the Naval Surface Warfare Center, Dahlgren, Virginia, He received the Ph.D. degree in Industrial and Systems Engineering in 1992. His major areas of research interests are integer programming, combinatorial optimization, network programming, and the implementation of mathematical programming for information systems.

Youngho Lee married Sonya L. Pak on May 23, 1988. They have a son Benjamin SeungBin, born on September 12, 1989. They have enjoyed out-door life in Blueridge Mountains while in Blacksburg, Virginia.

Youngho Lee