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

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. It 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 quadratic 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 ............................................. 1

The Polyhedral Structure of Combinatorial Optimization Problems .... 7

1. Sequential and Simultaneous Liftings of Minimal Cover Inequalities for
   GUB Constrained Knapsack Polytopes ....................................... 7
   1.1. Introduction ........................................................................ 7
   1.2. Valid Inequalities from Minimal GUB Covers ...................... 9
   1.3. Sequentially Lifted Facets from Minimal GUB Covers ............ 16
   1.4. Simultaneously Lifted Facets from Minimal GUB Covers .......... 27
   1.5. A Special Case: the Zero-One Knapsack Polytope ............... 39
   1.6. A Generalization of the GUB Cover ................................... 44
   1.7. Summary and Recommendations for Further Research .......... 46

2. The Polyhedral Structure of the Set Partitioning Polytope .......... 48
   2.1. Introduction ...................................................................... 48
   2.2. A Hierarchy of Relaxations for the Set Partitioning Polytope ... 50
   2.3. A Family of Valid Inequalities for the Set Partitioning Polytope 58
   2.4. A Family of Cutting Planes for the Set Partitioning Polytope .... 75
   2.5. Summary and Recommendations for Further Research .......... 83

3. A New Class of Lifted Facets for the Boolean Quadric Polytope .... 84
   3.1. Introduction ...................................................................... 84
   3.2. Polyhedral Structure of the Boolean Quadric Polytope .......... 86
7.2.4. Node Selection Rule and Fathoming Process .................................. 150
7.2.5. Algorithm GPBB ................................................................. 151

8. Computational Results ............................................................... 154
8.1. Computational Results for Heuristics HGP and HGPLP .................. 156
8.2. Computational Results for Heuristic HGPLD ............................... 157
8.3. Computational Results for Algorithm GPBB ............................... 158


10. Summary and Recommendations for Further Research ................... 170

References ................................................................. 172

Vita ................................................................. 180
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

Introduction: Motivation and Objective
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 combina-
torial 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.

\[
\text{GKP: Minimize } \sum_{j \in N} c_j x_j : \sum_{i \in M} \sum_{j \in N_i} a_{ij} x_j \geq b, \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M, \ x_j \in \{0,1\} \ \forall j \in N,
\]

where the data are all integer, \(N = \{1, \ldots, n\}\), \(M = \{1, \ldots, m\}\), and where \(\bigcup_{i \in M} N_i = N\), with \(N_i \cap N_j = \emptyset\) for \(i, j \in M, i \neq j\). Johnson and Padberg (1981) show that any GUB knapsack problem with arbitrarily signed coefficients \(b\) and \(a_{ij}, j \in N\), can be equivalently transformed into a form with \(b > 0\), and with \(0 < a_{ij} \leq b \ \forall j \in N\). Hence, without loss of generality, we will also assume that \(b > 0\) and that \(0 < a_{ij} \leq b \ \forall j \in N\). 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. Balintfy 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

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

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 $j(i) \in \arg\max_{j \in N_i} (a_{ij})$. Similarly, given any $B \subseteq N$, for each $i \in M_{B}$, define a key index $j_B(i)$ such that $j_B(i) \in \arg\max_{j \in N_i \cap B} (a_{ij})$. 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 $B_- = B - B_+$. 

The Polyhedral Structure of Combinatorial Optimization Problems
Let us suppose that for each \( k \in \mathbb{N} \)

\[
a_k + \sum_{i \in (M - M_0)} a_{i(0)} \geq b.
\] (1.1)

Otherwise, \( x_k = 0 \) in every feasible solution to \( X \). Denoting the convex hull operation by \( \text{conv}(\cdot) \), let \( \text{GUBKP} = \text{conv}(X) \), and let \( \dim(\text{GUBKP}) \) be the dimension of \( \text{GUBKP} \), which is the maximum number of affinely independent points in \( \text{GUBKP} \) minus one.

**Proposition 1.1** \( \dim(\text{GUBKP}) = n - |M_0| \), where \( M_0 = \{ i \in M : \sum_{p \in (M - i)} a_{ip} < b \} \).

**Proof.** By the definition of \( M_0 \), we must have \( \sum_{j \in M} x_j = 1 \) for each \( i \in M_0 \). Hence, it follows that \( \dim(\text{GUBKP}) \leq n - |M_0| \). To prove that \( \dim(\text{GUBKP}) = n - |M_0| \), it suffices to show that there exist \( n - |M_0| + 1 \) affinely independent points in \( \text{GUBKP} \).

For each \( i \in (M - M_0) \), we construct a set of feasible points in \( \text{GUBKP} \) as follows. For each \( k \in N_i \), construct \( x^{(k,0)} \equiv \{ x_k = 1, x_j = 1 \text{ for } j = j(p), \forall p \in (M - i), x_j = 0 \text{ otherwise } \} \), and let \( x^{(k,0)} \equiv \{ x_k = 1 \text{ for } j = j(p), \forall p \in (M - i), x_j = 0 \text{ otherwise } \} \). Similarly, for each \( i \in M_0 \), we construct a set of feasible points in \( \text{GUBKP} \) as follows. For each \( k \in N_i \), construct \( x^{(k,0)} \equiv \{ x_k = 1, x_j = 1 \text{ for } j = j(p), \forall p \in M - i, x_j = 0 \text{ otherwise } \} \).

Then, the total number of distinct feasible points thus constructed is \( n - |M_0| + 1 \). Let \( \tilde{X} \) be the set of these distinct points \( x_i \), indexed by \( j = 1, \ldots, n - |M_0| + 1 \). Without loss of generality, let \( x^{n - |M_0| + 1} \equiv \{ x_j = 1 \text{ for } j \in N^+, x_j = 0 \text{ otherwise} \} \).

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

Suppose that \(\text{dim}(\text{GUBKP})\) 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 \(x_{j, \min(i)} = 1 - \sum_{j \in \{N_i - \{\min(i)\}\}} x_j\) where \(\min(i) \in \text{argmin}(a_i)\). By replacing the variable \(x_{j, \min(i)}\) for \(i \in M_0\) and retaining the inequality constraints of the form \(\sum_{j \in \{N_i - \{\min(i)\}\}} x_j \leq 1\) 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 \(H \subseteq \mathbb{N}\), define \(X(H) = X \cap \{x \in \{0,1\}^n : x_{j,0} = 1, \forall i \in M_H\}\). 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 \(A \subseteq \mathbb{M}\),

\[
\text{dim}(\text{conv}(X(\bigcup_{i \in A} N_i))) = |\bigcup_{i \in (\mathbb{M} - A)} N_i|.
\]

A facet of an \(n\)-dimensional polytope is a maximal proper face, or, equivalently, a face of dimension \(n-1\). The inequality \(\sum_{j \in \mathbb{N}} \pi_j x_j \geq \pi_0\), denoted as \(\pi x \geq \pi_0\), is a facet defining inequality if (i) \(x \in \text{GUBKP}\) 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 $\sum_{i \in N_i} x_i \leq 1$, $i \in M$, 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 $\sum_{i \in M_K} a_{i(j)} \leq b - 1$, where $K = N - K$. A GUB cover $K$ is called a minimal GUB cover of $X$ if $\sum_{i \in M_K} a_{i(j)} + \min_{i \in M_K}(a_{i(j)}) \geq b$. Accordingly, we define a minimal GUB cover inequality as

$$\sum_{j \in K} x_j \geq 1.$$  \hspace{1cm} (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 $R = \{ j \in K : a_i \geq \max_{i \in K}(a_{i(j)}) \}$. An extension of the minimal GUB cover $K$ of $X$, denoted by $E(K)$, is defined as $E(K) = K \cup S$, where $S = \bigcup_{i \in M_K} N_i$.

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

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

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

*The Polyhedral Structure of Combinatorial Optimization Problems*
Proof. Suppose that there exists a \( \bar{x} \in X \) such that \( \sum_{j \in E(K)} \bar{x}_j \leq |M_0| \). From the definition of \( E(K) \), the largest \( |M_0| \) coefficients \( a_i \) among \( \{a_i : j \in E(K)\} \) are the \( \{a_i : j \in S_+\} \). But \( \sum_{i \in M_\bar{K}} a_{ij} \leq b \) implies that even with such a solution \( \bar{x} = \{\bar{x}_j = 1 \text{ for } j \in S_+, \bar{x}_j = 1 \text{ for } j \in (\bar{K} - S)_+, \bar{x}_j = 0 \text{ otherwise}\} \), 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 \( i \in M_\eta \), the inequality (1.3) dominates the minimal GUB cover inequality if \( R \neq \phi \), and this completes the proof. \( \blacksquare \)

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 \( S = \bigcup_{i \in (M_{K_0} - M_{K_0})} N_i \) for any \( p \in M_{K_0} \) is a cover, where \( j_\text{max}(a_i) \). 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 \( a_{i_1} + \sum_{i \in (M_{K_0} - p)} a_{ij} \geq b \forall p \in M_{K_0} \). 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

\[
\max \left\{ \sum_{j \in K_2} a_j x_j : \sum_{j \in N_i \cap K_2} x_j \leq 1 \ \forall \ i \in M_{K_2}, \ x_j \in (0,1) \ \forall \ j \in K_2 \right\} \leq b - \sum_{i \in M_{K_2}} a_{j(i)} - 1, \tag{1.4}
\]
then the following inequality

$$\sum_{j \in K_1} x_j \geq 1$$  \hspace{1cm} (1.5)$$

is valid for GUBKP and dominates the minimal GUB cover inequality \(\sum_{j \in K} x_j \geq 1\). Moreover, if \(\min_{j \in K} (a_j) + \sum_{j \in K_0} a_{j(0)} \geq b\), then the inequality (1.5) is a facet of \(\text{conv}(X(K_2, K))\) where \(X(K_2, K) = X \cap \{x \in (0,1)^n : x_j = 0 \ \forall \ j \in K_1, \ x_{j(0)} = 1 \ \forall \ i \in M_K\} \).

**Proof.** (i) It is readily shown that the inequality (1.5) is valid for GUBKP. (ii) Since \(\min_{j \in K_1} (a_j) + \sum_{i \in M_K} a_{i(0)} \geq b\), the unit vectors \(e_i\), for \(j \in K_1\), are feasible to \(\text{conv}(X(K_2, K))\), and moreover, they satisfy (1.5) as an equality, and are linearly independent. Hence, the inequality (1.5) is a facet of \(\text{conv}(X(K_2, K))\). This completes the proof. ■

Note that the condition (1.4) of Proposition 1.5 can be restated as \(\sum_{i \in M_K} \max_{j \in K_1 \cap K_2} (a_j) + \sum_{i \in M_K} a_{i(0)} < b\). Furthermore, if no such partition with \(K_2 \neq \phi\) exists, i.e., if \(\min_{j \in K} (a_j) + \sum_{i \in M_K} a_{i(0)} \geq b\), 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 \(\text{conv}(X(K))\) if and only if \(\min_{j \in K} (a_j) + \sum_{i \in M_K} a_{i(0)} \geq b\).

**Proof.** Suppose that the minimal GUB cover inequality is a facet of \(\text{conv}(X(K))\). Then, by the structure of (1.2), there must exist \(|K|\) unit vectors \(e_i\), one for each \(j \in K\), which belong to \(\text{conv}(X(K))\). Hence, we have that \(a_j \geq b - \sum_{i \in M_K} a_{i(0)}\) for all \(j \in K\), and so, \(\min_{j \in K} (a_j) + \sum_{i \in M_K} a_{i(0)} \geq b\). Conversely, if \(\min_{j \in K} (a_j) + \sum_{i \in M_K} a_{i(0)} \geq b\), then the unit vectors \(e_i\), \(j \in K\), are feasible to \(\text{conv}(X(K))\), satisfying (1.2) as an equality, and are linearly independent. Hence, the valid inequality (1.2) is a facet of \(\text{conv}(X(K))\). This completes the proof. ■
Example 1.1 Consider the following example, where $X = \{x \in (0,1)^8 : x_1 + 5x_2 + x_3 + 5x_4 + x_5 + 3x_6 + x_7 + 3x_8 \geq 9, x_1 + x_2 \leq 1, x_3 + x_4 \leq 1, x_5 + x_6 \leq 1, x_7 + x_8 \leq 1\}$.

Since for all $i \in M$, $\sum_{p \in (M - 0)} a_{ij(p)} \geq 9$, 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 $\text{conv}(X(K_2, \overline{K}))$, where $X(K_2, \overline{K}) = X \cap \{x \in (0,1)^8 : x_1 = x_3 = 0, x_2 = x_4 = 1\}$. 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 $x_1 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 2$. This minimal GUB cover $K = \{3, 4, 5, 6\}$ can be verified to be strong. Moreover, since $\min_{i \in K} a_i + \sum_{i \notin K} a_{ij} = 9 = b$, by Proposition 1.6, the corresponding minimal GUB cover inequality $x_3 + x_4 + x_5 + x_6 \geq 1$ is a facet of $\text{conv}(X(\overline{K})) = X \cap \{x \in (0,1)^8 : x_2 = 1, x_4 = 1\}$.

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 $\sum_{i \in M_H} \max_{j \in H} a_{ij} + \sum_{i \notin M_H} a_{ij(p)} < b$. 

The Polyhedral Structure of Combinatorial Optimization Problems 15
Proof. Since \( \sum_{j \in H} x_j \geq 1 \) is valid for GUBKP, we have that \( \sum_{j \in M} a_j \leq b - 1 \), which can be restated as \( \sum_{i \in M - M_H} a_{j(i)} + \sum_{i \in M_H} \max_{j \in (N_i - k)} (a_j) \leq b - 1 \). 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 \( \tilde{H} \) is not minimal, then there exists an \( s \in H \) such that \( \tilde{H} = H - s \) satisfies the condition of the proposition. Then, by Proposition 1.5, \( \sum_{j \in \tilde{H}} x_j \geq 1 \) is valid for GUBKP. Since \( \sum_{j \in \tilde{H}} x_j \geq 1 \) 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 \( p \in M_K \), where \( K \) is a minimal GUB cover of \( X \). Noting that the minimal GUB cover inequality (1.2) is valid for \( \text{conv}(X(K)) \), 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 \( p \in M_K \), define \( \eta^t \), for \( t = j(p) \), as

\[
\eta^t \equiv \min \{ \sum_{j \in K} x_j : x \in X, \quad x_j = 0 \quad \forall \ j \in N_p \} \\
= \min \{ \sum_{i \in M_K} x_{(i)} : \sum_{i \in M_K} a_{(i)} x_{(i)} \geq b - \sum_{i \in (M_K - p)} a_{(i)}, \ x_{(i)} \in (0,1) \quad \forall \ i \in M_K \},
\]

and define \( \xi^s \), for each \( s \in (N_p - j(p)) \), as
\[ \zeta^s \equiv \min \{ \sum_{j \in K} x_j : x \in X, \ x_s = 1 \} \]
\[ \equiv \min \{ \sum_{i \in M_K} x_{i(j)} : \sum_{i \in M_K} a_{i(j)} x_{i(j)} \geq b - a_s - \sum_{i \in (M_K - p)} a_{i(j)}, \ x_{i(j)} \in \{0,1\} \ \forall \ i \in M_K \}. \] (1.7)

Since \( K \) is a minimal GUB cover of \( X \), and since GUBKP is full dimensional, and by (1.1), note that \( 1 \leq \eta' \leq |M_K| \) and \( 1 \leq \zeta^s \leq \eta' \ \forall \ s \in (N_s - t) \).

**Proposition 1.8** (i) For a given minimal GUB cover \( K \) of \( X \), the following inequality, defined for some \( p \in M_K \) as

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

where \( t = j(p) \), is a valid inequality for GUBKP for any \( \alpha_t \leq \eta' - 1 \) and for any \( \alpha_s \geq -\zeta^s + \alpha_t + 1 \ \forall \ s \in (N_p - t) \).

(ii) Moreover, if \( \alpha_t = \eta' - 1 \), \( \alpha_s = \eta' - \zeta^s \ \forall \ s \in (N_p - t) \), and \( \min_{j \in K} (a_j) + \sum_{i \in M_K} a_{i(j)} \geq b \), then the inequality (1.8) is a facet of \( \text{conv}(X(K - N_s)) \).

**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 \( s \in (N_s - t) \), 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 \( s \in (N_s - t) \), or if \( \bar{x} \in X \) and has \( \bar{x}_j = 0 \ \forall \ j \notin N_s \), then such \( \bar{x} \) and \( \bar{x} \) satisfy (1.8). That is, \( \sum_{j \notin K} x_j + \alpha_t \geq 1 + \alpha_t \), and \( \sum_{j \notin K} x_j \geq 1 + \alpha_t \). By the definition of the quantities \( \alpha_t \) and \( \alpha_s \), in the first case, we have, using (1.7), that \( \sum_{j \notin K} x_j + \alpha_t \geq \zeta^s - \zeta^s + \alpha_t + 1 \geq 1 + \alpha_t \), and in the second case,
we have, using (1.6), that \( \sum_{j \in \mathbb{K}} \hat{x}_j \geq \eta^t \geq 1 + \alpha_t \). Hence, the inequality (1.8) is valid for GUBKP.

(ii) Since the minimal GUB cover inequality (1.2) is a facet of \( \text{conv}(X(\mathbb{K})) \) by Proposition 1.6, and since \( \text{conv}(X(\mathbb{K})) \) is of full dimension \(|\mathbb{K}|\) by Corollary 1.1, there exist \(|\mathbb{K}|\) linearly independent vertices of \( \text{conv}(X(\mathbb{K})) \), indexed by \( x^t, j = 1, \ldots, |\mathbb{K}| \), which satisfy the inequality (1.2) as an equality. Since \( x_{i|0} = 1 \), for \( i \in \mathbb{M}_\mathbb{K}, \ j = 1, \ldots, |\mathbb{K}| \), these vertices also satisfy the inequality (1.8) as an equality and moreover, these vertices belong to \( X(\mathbb{K} - N_p) \). Now, for each \( s \in (N_p - t) \), let \( \hat{x}^t \) be a solution of (1.7) such that \( \hat{\xi}^t = \sum_{j \in \mathbb{K}} \hat{x}^t_j \). Note that \( \hat{x}^t \in X(\mathbb{K} - N_p) \). Also, for \( t = j(p) \), let \( \hat{x}^t \) be a solution of (1.6) such that \( \eta^t = \sum_{j \in \mathbb{K}} \hat{x}^t_j \), and note that \( \hat{x}^t \in X(\mathbb{K} - N_p) \). Moreover, \( \hat{x}^* \) and \( \hat{x}^1 \) satisfy the inequality (1.8) as an equality when \( \alpha_s = \eta^t - \xi^t \), and \( \alpha_t = \eta^t - 1 \).

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

\[
\sum_{j = 1}^{\mathbb{K}_1} \lambda_j x^t_j + \delta_t \hat{x}^t + \sum_{s \in (N_p - t)} \mu_s \hat{x}^s = 0 \quad \text{and} \quad \sum_{j = 1}^{\mathbb{K}} \lambda_j + \delta_t + \sum_{s \in (N_p - t)} \mu_s = 0. \tag{1.9}
\]

Since for each \( s \in (N_p - t) \), \( \hat{x}^s_t = 0 \), \( x^t_s = 0 \) for \( j = 1, \ldots, |\mathbb{K}| \), and \( \hat{x}^* = 1 \), it follows that \( \mu_s = 0 \) \( \forall \ s \in (N_p - t) \). Now, if \( \delta_t = 0 \), then by the linear independence of \( x^t, j = 1, \ldots, |\mathbb{K}| \), we would have \( \lambda_j = 0, \forall j = 1, \ldots, |\mathbb{K}| \), a contradiction. Hence, without loss of generality,
suppose that $\delta_i = -1$, 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 $\hat{x}^t = 0$, $x_j = 1$ for $j = 1, \ldots, [K]$, it follows that $\sum_{j=1}^{[K]} \lambda_j = 0$, which is a contradiction. Hence, the vectors $\{(x^t, \text{ for } j = 1, \ldots, [K]) \}$, $\hat{x}^t$, ( $\hat{x}^s$ for $s \neq (N_s - t)$) 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 $\text{conv}(X(\overline{K}))$. Let $M_k = \{i_1, \ldots, i_k\}$ be arbitrarily ordered, where $k = |M_k|$. Let $M(q) = \{i_1, \ldots, i_k\} \subseteq M_k$, and let $N(q) = \bigcup_{i \in M(q)} N_i$ for $q = 1, \ldots, k$. For $q = 0$, let $M(q) = N(q) = \emptyset$. Let $q \in \{0, \ldots, k-1\}$ and suppose that $\sum_{j \in K \cup N(q)} \alpha x_j \geq \alpha_0$ is valid for GUBKP and is a facet of $\text{conv}(X(\overline{K} - N(q)))$. Consider $i_{q+1}$.

Denote $t = j(i_{q+1})$, and compute

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

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

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

Then,

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

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

The proof of Proposition 1.9 follows the same argument used in the proof of Proposition 1.8. Note that if $q = 0$, then $\eta(q) = \eta^t$ and $\zeta(q) = \zeta^s$, 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 $\sum_{j \in K} x_j + \sum_{j \in K_-} \alpha^j x_j - \sum_{j \in K_+} \alpha^j (1 - x_j) \geq 1$. This is of the form

$$\sum_{j \in K} x_j + \sum_{j \in K_-} \alpha^j x_j + \sum_{j \in K_+} \alpha^j x_j \geq 1 + \sum_{j \in K_+} \alpha^j. \quad (1.13)$$

The derivation of the coefficients $\alpha^j$ 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 $i \in M_K$ are considered. For each $i \in M_K$, let $\alpha^i$ denote the value of $\alpha^j$ when $i = j$, i.e., when $i$ is taken to be the first index in $M_K$. In other words, $\alpha^i = \eta^t - 1$ for $t = j(i)$, and $\alpha^i = \eta^t - \zeta^t$ for $\forall t \in N_0 - x(i)$. 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 \( \eta' \) and \( \zeta^* \) 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 \( t \in \overline{K}_- \), \( \eta' = h \), where \( h \) is defined by

\[
\sum_{k \in K_h} a_k \geq b - \sum_{l \in (M_K^+ - M)} a_{j(l)} > \sum_{k \in K_h - 1} a_k.
\]

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

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 \( s \in \overline{K}_- \), \( \zeta^* = h \), where \( h \) is defined by

\[
\sum_{k \in K_h} a_k \geq b - a_s - \sum_{l \in (M_K^+ - M)} a_{j(l)} > \sum_{k \in K_h - 1} a_k. \]

Note that for a given \( p \in M_K^+ \), \( \zeta'^{11} \geq \zeta'^{12} \) whenever \( a_{j1} \leq a_{j2} \), for \( j1, \ j2 \in N_p \cap \overline{K}_- \).

**Corollary 1.2**  For a given \( p \in M_K^+ \), if \( \eta' = 1 \), then \( \zeta^* = 1 \quad \forall \ s \in (N_p - t) \), where \( t = j(p) \).

**Proof.** Follows from the fact that \( 1 \leq \zeta^* \leq \eta' \quad \forall s \in N_p - t \). ■
Corollary 1.3  Let \( t, e \arg\max(a_{ij}) \). If \( \eta^n = 1 \), then \( \xi = 1 \ \forall \ s \in (N, - j(i)), \ i \in M_K \), and \( \eta^{(j)} = 1 \ \forall \ i \in M_K \).

Proof. From (1.6), it follows that \( 1 \leq \eta^{(j)} \leq \eta^n \ \forall \ i \in M_K \). 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 \( \text{conv}(X(K)) \), and if \( \max(a) + \sum_{i \in K} a_{ij} \geq b \), where \( t, e \arg\max(a_{ij}) \), then the minimal GUB cover inequality is a facet of GUBKP.

Proof. Since \( \max(a) + \sum_{i \in K} a_{ij} \geq b \), we have that \( \eta^n = 1 \). 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 \( X = \{x \in (0, 1)^7 : 2x_1 + 4x_2 + x_3 + 2x_4 + x_5 + 2x_6 + x_7 \geq 4, \ x_1 + x_2 \leq 1, \ x_3 + x_4 \leq 1, \ x_5 + x_6 \leq 1, \ x_7 \leq 1 \} \). For a minimal GUB cover \( K = \{1, 2, 3, 4\} \), the minimal GUB cover inequality is \( x_1 + x_2 + x_3 + x_4 \geq 1 \), which is a facet of \( \text{conv}(X(K)) \). Since \( t, e = 6 \) and \( \eta^e = 1 \), by Proposition 1.12, \( x_1 + x_2 + x_3 + x_4 \geq 1 \) is a facet of \( \text{conv}(X) \). Note that \( K \) is a strong GUB cover.
Finally, we show that the readily obtained coefficients \( \alpha_i \), \( j \in \mathcal{K} \), provides bounds on the coefficients \( \beta_i \), \( j \in \mathcal{K} \), of arbitrary valid inequalities (not necessarily sequentially lifted inequalities) that have unit coefficients for all \( j \in \mathcal{K} \).

**Proposition 1.13**  If \( \sum_{j \in K} \sum_{j \in K_-} \beta_j x_j \geq \sum_{j \in K_+} \beta_j (1 - x_j) \geq 1 \) is valid for GUBKP, then \( \beta_i \leq \alpha'_i \) \( \forall j \in \mathcal{K}_+ \), and \( \beta_i \geq \alpha'_i - (\alpha'_i - \beta_i) \), where \( t = j(p) \) such that \( j \in \mathcal{N}_p \cap \mathcal{K}_- \), \( \forall p \in \mathcal{M}_\mathcal{K} \).

**Proof.** Assume that for some \( t \in \mathcal{K}_- \), \( \beta_i > \alpha'_i \). From Proposition 1.8, we have that \( \alpha' = \eta - 1 \). Let \( \bar{x} \) be an optimal solution of (1.6) such that \( \eta \bar{x} = \sum_{j \in K} \bar{x}_j \). Then, \( \bar{x} \in X \), but

\[
\sum_{j \in K} \bar{x}_j + \sum_{j \in K_-} \beta_j \bar{x}_j - \sum_{j \in K_+} \beta_j (1 - \bar{x}_j) = \eta - \beta_i = \alpha'_i + 1 - \beta_t < 1,
\]

and so, the inequality in the proposition is violated. Hence, \( \beta_i \leq \alpha'_i \), \( \forall t \in \mathcal{K}_- \). For the case of \( \beta_i \), \( j \in \mathcal{K}_- \), suppose that for some \( p \in \mathcal{M}_\mathcal{K} \), and \( s \in \mathcal{N}_p \), we have \( \beta_s < \alpha'_s - (\alpha'_s - \beta_s) \) where \( t = j(p) \). Consider problem (1.7), and let \( \bar{x} \in X \) solve this problem. Then, since \( \alpha'_s - \alpha'_s = 1 - \zeta_s \), we have that

\[
\sum_{j \in K} \bar{x}_j + \sum_{j \in K_-} \beta_j \bar{x}_j - \sum_{j \in K_+} \beta_j (1 - \bar{x}_j) = \zeta_s + \beta_s - \beta_t < \zeta_s + \alpha'_s - \alpha'_t = 1,
\]

and so the inequality in the proposition is again violated. This completes the proof. \( \blacksquare \)

We now consider the task of efficiently computing the coefficients \( \alpha_i \), \( j \in \mathcal{K} \), of a sequentially lifted facet (1.13). As mentioned earlier, the task of computing \( \alpha_i \) for \( j \in \mathcal{K} \)
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 $\alpha_i$ for all $j \in K$ within a time complexity of $O(n|\mathcal{M}_x|)$ 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.

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

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

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

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

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

**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, $v(b) = \min \{ z : w(z) \geq b \}$. 

The Polyhedral Structure of Combinatorial Optimization Problems 24
Proof. Problem $P(b)$ seeks the value $v(b) = \min \{ z : \sum_{j \in N} c_j x_j \leq z, \sum_{j \in N} a_j x_j \geq b, \sum_{j \in N} x_j \leq 1 \forall i \in M, x_j \in (0,1) \forall j \in N \}$. Equivalently, $v(b) = \min \{ z : \text{there exists } x \text{ satisfying} \sum_{j \in N} c_j x_j \leq z, \sum_{i \in N} a_j x_j \geq b, \sum_{j \in N} x_j \leq 1 \forall i \in M, x_j \in (0,1) \forall j \in N \}$. Hence, we have that $v(b) = \min \{ z : w(z) \geq b \}$. ■

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

$$\sum_{j \in K} x_j + \sum_{j \in T^-} a_j x_j - \sum_{j \in T^+} x_j(1 - x_j) \geq 1. \tag{1.14}$$

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

$$1 + \alpha_t = \min \{ \sum_{j \in K} x_j + \sum_{j \in T^-} a_j x_j - \sum_{j \in T^+} \alpha_j(1 - x_j) : \sum_{j \in (K \cup T)} a_j x_j \geq b - \sum_{i \in (M_R - M_T - p)} a_j(0), \sum_{j \in N_i} x_j \leq 1 \forall i \in M, x_j \in (0,1) \forall j \in N \} ,$$

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

$$1 + \alpha_t - \alpha_s = \min \{ \sum_{j \in K} x_j + \sum_{j \in T^-} a_j x_j - \sum_{j \in T^+} \alpha_j(1 - x_j) : \sum_{j \in (K \cup T)} a_j x_j \geq b - a_s - \sum_{i \in (M_R - M_T - p)} a_j(0), \sum_{j \in N_i} x_j \leq 1 \forall i \in M, x_j \in (0,1) \forall j \in N \} .$$

Let $L = K \cup T, \bar{L} = N - L$, and define
\[ w_L(z) = \max \left\{ \sum_{j \in L} a_j x_j : \sum_{j \in K} x_j + \sum_{j \in T_-} \alpha_j x_j - \sum_{j \in T_+} \alpha_j (1 - x_j) \leq z, \right. \]
\[ \left. \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M, \ x_j \in (0,1) \ \forall j \in N \right\}. \tag{1.15} \]

By Proposition 1.15, we have that

\[ 1 + \alpha_t = \min \{ z : w_L(z) \geq b - \sum_{i \in (M^- \setminus p)} a_{i(j)} \}, \text{ and} \]

\[ 1 + \alpha_t - \alpha_s = \min \{ z : w_L(z) \geq b - a_s - \sum_{i \in (M^- \setminus p)} a_{i(j)} \}. \]

Hence, we can efficiently obtain the coefficients \( \alpha_j \) for \( j \in \mathbb{N}_p \), 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

\[ w_{L \cup \mathbb{N}_p}(z) = \max \left\{ \sum_{j \in L} a_j x_j + \sum_{j \in \mathbb{N}_p} a_j x_j : \right. \]
\[ \left. \sum_{j \in K} x_j + \sum_{j \in T_-} \alpha_j x_j - \sum_{j \in T_+} \alpha_j (1 - x_j) + \sum_{s \in (\mathbb{N}_p \setminus t)} \alpha_s x_s - \alpha_t (1 - x_t) \leq z, \right. \]
\[ \left. \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M, \ x_j \in (0,1) \ \forall j \in N \right\}. \]

Equivalently, this yields,

\[ w_{L \cup \mathbb{N}_p}(z) = \max \{ \max_{s \in (\mathbb{N}_p \setminus t)} [a_s + w_L(z + \alpha_t - \alpha_s)], [a_t + w_L(z)] \}. \tag{1.16} \]
Hence, we can compute the coefficients $\alpha_i$ 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 $\max_{z} w_i(z)$. Then, it follows that $w_i(z) = w_i(\hat{z}) \forall z \geq \hat{z}$. For any minimal GUB cover $K$, to begin with, since $w_i(z) = w_i(\lfloor M_k \rfloor)$ for all $z \geq \lfloor M_k \rfloor$, we only need to compute $w_i(z)$ for $z = 1, \ldots, \lfloor M_k \rfloor$. Moreover, since $\alpha_i - \alpha_* \geq 0 \forall s \in (N_\ast - 1)$, we have recursively that the value in (1.16), for each $L$ and $N_\ast$ thereafter, also remains a constant for $z \geq \lfloor M_k \rfloor$. Therefore, each function $w_i(z)$ needs to be evaluated (recursively) via (1.16) only for $z = 1, \ldots, \lfloor M_k \rfloor$. Hence, the time complexity of computing the lifted coefficients $\alpha_i$, for $j \in \overline{K}$, is $O(n|\lfloor M_k \rfloor|)$.

### 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

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

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

$$\text{GUBKP} \equiv \text{conv}(\mathcal{X}) = \{ x : \quad x_j = \sum_{J \in \mathcal{J}} y_J \quad \forall \; j \in N, \quad \sum_{J \in \mathcal{J}} y_J = 1, \quad y_J \geq 0, \quad \forall \; J \in \mathcal{F}, y_J = 0 \quad \forall \; J \in \overline{F} \}$$

The Polyhedral Structure of Combinatorial Optimization Problems 27
Using the standard projection operation, we obtain

\[
\text{GUBKP} \equiv \{ x : \sum_{j \in N} \pi_j^k x_j \geq \pi_0^k \}
\]

where \((\pi_j^k, \pi_0^k), k = 1, \ldots, K\), are the extreme directions of \(\Pi\).

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

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 \(\sum_{j \in K} x_j \geq 1\) is a valid inequality for \(\text{conv}(X(\overline{K}))\), and a facet for \(\text{conv}(X(\overline{K}))\) if \(\min_{j \in K} (a_j) + \sum_{i \in M_K} a_{ji} \geq b\). We are interested in finding a lifted inequality, which is a facet of GUBKP, and is of the form \(\sum_{j \in K} x_j + \sum_{j \in K_-} \pi_j x_j - \sum_{j \in K_+} \pi_j x_j \geq 1\)

where \(\bar{x}_j \equiv (1 - x_j) \ \forall \ j \in N\), and \(\pi_j \ \forall \ j \in \overline{K}\) are unrestricted in sign. This is of the form

\[
\sum_{j \in K} x_j + \sum_{j \in K_-} \pi_j x_j + \sum_{j \in K_+} \pi_j x_j \geq 1 + \sum_{j \in K_-} \pi_j.
\]

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

\[
\Pi_{\overline{K}} \equiv \{ \pi_{\overline{K}} : \sum_{j \in J \cap \overline{K}} \pi_j \geq 1 + \sum_{j \in \overline{K}_-} \pi_j - |J \cap K|, \ \forall \ J \in F\}.
\]

Equivalently, we have,
\[ \Pi_K \equiv \{ \pi_K : \sum_{j \in (K_+ - J)} \pi_j - \sum_{j \in J \cap K_-} \pi_j \leq |J \cap K| - 1, \ \forall \ J \in F \}. \]

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

\[ \Pi_K \equiv \{ \pi_K : \sum_{j \in \overline{K_+} - J} \pi_j - \sum_{j \in J \cap \overline{K_-}} \pi_j \leq |J \cap K| - 1, \ \forall \ J \in F \equiv \overline{KJ} \neq \emptyset \}. \quad (1.20) \]

Furthermore, note that we need to examine only the most restrictive constraints in (1.20). Toward this end, we define \( \text{GUB}(\overline{K}) = \{ T \subseteq \overline{K} : |T \cap N_i| \leq 1, \ \forall \ i \in M_\overline{K} \} \). Now for any \( T \in \text{GUB}(\overline{K}) \), define the feasible extension of \( T \) as \( J(T) = \{ J \in F : J = \overline{J_i} \cup T \text{ for some } J_i \subseteq K \} \), and let \( \overline{K_T} = (T \cap \overline{K_-}) \cup (\overline{K_+} - T) \). Accordingly, define

\[ F_K = \{ T \in \text{GUB}(\overline{K}) : J(T) \neq \emptyset, \text{ and } \overline{K_T} \neq \emptyset \}. \]

Then, we can restate (1.20) as follows.

\[ \Pi_K \equiv \{ \pi_K : \sum_{j \in \overline{K_+} - T} \pi_j - \sum_{j \in T \cap \overline{K_-}} \pi_j \leq \min \{ |J \cap K| : J \in J(T) \} - 1, \ \forall \ T \in F_K \} \quad (1.21) \]

We now consider an explicit representation of the minimization problem in (1.21).

Note that \( T \in F_K \) if and only if (i) \( |T \cap N_i| \leq 1 \) for \( i \in M_\overline{K} \), i.e., \( T \in \text{GUB}(\overline{K}) \), (ii) \( \sum_{i \in K_+} a_i \geq b \), i.e., \( J(T) \neq \emptyset \), and (iii) \( \overline{K_T} \neq \emptyset \). Hence, for each \( T \in F_K \), we can represent the minimization problem in (1.21), denoted by \( \text{AGUBKP}(T) \), as follows.
AGUBKP(T): Minimize \( \sum_{j \in K_+} y_j : \sum_{j \in K_+} a_j y_j \geq b - \sum_{j \in T} a_j, \quad y_j \in \{0,1\} \quad \forall j \in K_+ \)

Note that \( 1 \leq \nu(\text{AGUBKP}(T)) \leq |M| \), 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 \( \tilde{b} = b - \sum_{j \in T} a_j \). If \( \tilde{b} \) is less than or equal to \( \max(a_j) \), then \( \nu(\text{AGUBKP}(T)) = 1 \). Otherwise, if \( \tilde{b} \) 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

\[
\Pi_{\overline{K}} = \{ \pi_{\overline{K}} : \sum_{j \in \overline{K}_+ - T} \pi_j - \sum_{j \in T \cap \overline{K}_-} \pi_j \leq N(T) \quad \forall \ T \in F_{\overline{K}} \}.
\]  

(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 \( \pi_{\overline{K}} \in \Pi_{\overline{K}} \), where \( \pi_{\overline{K}} = \{ \pi_j : j \in \overline{K} \} \), and \( \Pi_{\overline{K}} \) 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

\[
\min \{ \sum_{j \in N} \pi_j (\sum_{j \in J} y_j) : \sum_{j \in N} y_j = 1, \ y_j \geq 0 \ \forall \ J \in F, \ y_J = 0 \ \forall \ J \in \overline{F} \} \geq \pi_0.
\]

The above condition is equivalent to requiring that

\[
\min \{ \sum_{j \in J} y_j (\sum_{j \in J} \pi_j) : \sum_{j \in J} y_j = 1, \ y_j \geq 0 \ \forall \ J \in F \} \geq \pi_0,
\]
which in turn is equivalent to

\[ \sum_{J \in \mathcal{J}} \pi_J \geq \pi_0, \quad \forall J \in \mathcal{F}. \]

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 \( \pi \in \Pi_K \), where \( \Pi_K \) is given by (1.22). This completes the proof. \( \blacksquare \)

**Proposition 1.17** Let \( K \) be a minimal GUB cover such that \( \min_{i \in K} (a_i) + \sum_{i \in M_K} a_{i0} \geq b \), and let \( \Pi_K \) be given by (1.22). Then, the inequality (1.18) having \( 1 + \sum_{J \in \mathcal{K}_+} \pi_J > 0 \) is a facet of GUBKP if and only if \( (\pi, j \in K) \) is a vertex of \( \Pi_K \) with \( 1 + \sum_{j \in \mathcal{K}_+} \pi_j > 0 \).

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

\[ \Pi_K = \{ \pi : \sum_{J \in \mathcal{J}} \pi_J \geq 1, \quad \forall J \in \mathcal{F} \}. \tag{1.23} \]

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

\[ \pi = \{ (\hat{\pi}_j = \frac{1}{1 + \sum_{j \in \mathcal{K}_+} \pi_j}, \quad j \in K), \quad \hat{\pi}_j = \frac{\pi_j}{1 + \sum_{j \in \mathcal{K}_+} \pi_j}, \quad j \in K \} \tag{1.24} \]

is an extreme point of (1.23).

The Polyhedral Structure of Combinatorial Optimization Problems 31
Now, for each \( j \in K \), the set \( J(j) \equiv \{ j \} \cup K_+ \in F \) 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

\[
\pi_j = 1 - \sum_{t \in K_+} \pi_t \quad \text{for} \quad j \in K 
\]

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

\[
\{ \pi_K : \sum_{j \in J \cap \bar{K}} \pi_j \geq 1 + |J \cap K| \left( \sum_{t \in K_+} \pi_t - 1 \right) \forall J \in F \}. \tag{1.26}
\]

This holds if and only if \( \{ \hat{\pi}_j, j \in \bar{K} \} \) is feasible to (1.26), and there exist some \( |K| \) linearly independent hyperplanes that are binding at \( \{ \hat{\pi}_j, j \in \bar{K} \} \). Feasibility of \( \hat{\pi} \) to (1.26) requires from (1.24) that

\[
\sum_{j \in J \cap \bar{K}} \frac{\pi_j}{1 + \sum_{t \in K_+} \pi_t} \geq 1 + |J \cap K| \left( \frac{\sum_{t \in K_+} \pi_t}{1 + \sum_{t \in K_+} \pi_t} - 1 \right) \forall J \in F.
\]

That is, we must have...
\[ \sum_{j \in J \cap K} \bar{\pi}_j \geq 1 + \sum_{i \in K^-} \bar{\pi}_i - |J \cap K| \quad \forall j \in F. \quad (1.27) \]

Note by (1.19) that (1.27) is equivalent to requiring that \( \bar{\pi}_K \) belongs to \( \Pi_K \). 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 \( |K| \) linearly independent equations from (1.26) give \( \hat{\pi} \) as the unique solution if and only if the corresponding \( |K| \) equations from (1.27) give \( \bar{\pi} \) as the unique solution, since from (1.24), there is a 1-1 correspondence between \( \hat{\pi} \) and \( \bar{\pi} \) according to

\[ \begin{align*}
(\hat{\pi}_j &= \frac{\bar{\pi}_j}{1 + \sum_{j \in K^c} \bar{\pi}_j} \quad \forall j \in K} \text{ and } (\bar{\pi}_j &= \frac{\hat{\pi}_j}{1 - \sum_{j \in K^c} \hat{\pi}_j} \quad \forall j \in K}.\end{align*} \]

Hence, \( \bar{\pi}_K \) is an extreme point of (1.26) if and only if \( \bar{\pi}_K \) is an extreme point of \( \Pi_K \) with \( 1 + \sum_{j \in K^c} \bar{\pi}_j > 0 \), and this completes the proof. \( \blacksquare \)

\textbf{Example 1.3} Consider the following example to illustrate the above simultaneous lifting procedure. Let \( X = \{ x \in (0,1)^9 : x_1 + x_2 + 2x_3 + x_4 + x_5 + 2x_6 + x_7 + x_8 + 3x_9 \geq 4, x_1 + x_2 + x_3 \leq 1, x_4 + x_5 + x_6 \leq 1, x_7 + x_8 + x_9 \leq 1 \} \). A minimal GUB cover inequality is \( x_1 + x_2 + x_3 + x_4 + x_5 + x_8 \geq 1 \), which is a facet of \( \text{conv}(X(K)) \), where \( K = \{7, 8, 9\} \). For this minimal cover, we have that \( F_K = \{ \phi, \{7\}, \{8\} \} \), thereby leading to the following computations.
<table>
<thead>
<tr>
<th>$T$</th>
<th>$\overline{KT}$</th>
<th>$\sum_{j \in T} a_j$</th>
<th>$\nu(\text{AGUBKP}(T))$</th>
<th>Inequalities of $\Pi_\bar{K}$ in (1.22)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi)</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>$\pi_0 \leq 1$</td>
</tr>
<tr>
<td>7</td>
<td>7, 9</td>
<td>1</td>
<td>2</td>
<td>$\pi_0 - \pi_1 \leq 1$</td>
</tr>
<tr>
<td>8</td>
<td>8, 9</td>
<td>1</td>
<td>2</td>
<td>$\pi_0 - \pi_2 \leq 1$</td>
</tr>
</tbody>
</table>

The point \((0, 0, 1)\) is the only vertex of $\Pi_\bar{K}$ with $1 + \sum_{j \in \bar{K}_+} \pi_j = 2 > 0$. Hence, $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \geq 2$ is the only facet obtainable from the minimal GUB cover inequality by the lifting procedure.

**Lower and Upper Bounds on the Lifted Coefficients $\pi_j$ for $j \in \bar{K}$ 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 $\pi_t$ for $t \in \bar{K}_+$. Since GUBKP is a full dimensional polytope, it follows that for any $t \in \bar{K}_+$, we have $T = (\bar{K}_+ - t) e_{\bar{K}_+}$ and from (1.22), we directly have that $\pi_t \leq N(\bar{K}_+ - t)$. Hence, from (1.6), an upper bound $UB_t$ on $\pi_t$ is given by

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

Now, for given lower bounds $LB_t$ on $\pi_t \forall s \in (N_+ - t)$, where $p = M_t$, let us derive a lower bound $LB_t$ on $\pi_t$. Toward this end, examine any $T \in F_K$ such that $t \not\in T$. Then, from (1.22), we have that
\[
\sum_{j \in \overline{K}_n - T} \pi_j - \sum_{j \in T \cap \overline{K}_n} \pi_j \leq N(T) \quad \forall T \in F_K \ni t \notin T. \quad (1.28)
\]

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

\[
\pi_t \leq N(T) - \left( \sum_{j \in \overline{K}_n - T - t} \pi_j - \sum_{j \in T \cap \overline{K}_n} \pi_j \right) \quad \forall T \in F_K \ni t \notin T. \quad (1.29)
\]

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

\[
\pi_t = \min \{ N(T) - \left( \sum_{j \in \overline{K}_n - T - t} \pi_j - \sum_{j \in T \cap \overline{K}_n} \pi_j \right) : T \in F_K \ni t \notin T \}. \quad (1.30)
\]

For all \( T \in F_K \ni t \notin T \), define \( \nu(T) = \sum_{j \in (\overline{K}_n - T)} \pi_j - \sum_{j \in T \cap \overline{K}_n} \pi_j \), \( \overline{v}(T) = \sum_{j \in (\overline{K}_n - T - t)} \pi_j - \sum_{j \in T \cap \overline{K}_n} \pi_j \). Then equation (1.30) reads

\[
\pi_t = \min \{ N(T) - \overline{v}(T) : T \in F_K, t \notin T \} \quad (1.31)
\]

For any \( T \in F_K \ni t \notin T \), if \( (T + t) \in F_K \), then we have that \( \overline{v}(T) = \nu(T + t) \leq N(T + t) \) by (1.28).

On the other hand, if \( (T + t) \notin F_K \), then, there exists some \( s \in T \cap N_s \) and \( \overline{v}(T) = \nu(T + t - s) - \pi_s \leq N(T + t - s) - \pi_s \). Hence, from (1.31), we have

\[
\pi_t = \min \{ \min_{T \in T_1} \{ N(T) - N(T + t) \}, \min_{T \in T_2} \{ N(T) - N(T + t - s) + \pi_s \} \}
\]
where \( T_1 = \{ T e F_k : t \notin T \text{ and } (T + t) e F_k \} \), \( T_2 = \{ T e F_k : t \notin T \text{ and there exists some } s e (N_{p} - t) \cap T \} \text{ where } p \equiv M_i \). Hence, for given lower bounds \( \text{LB}_s \) on \( \pi_i \) for \( s e (N_{p} - t) \), a lower bound \( \text{LB}_t \) on \( \pi_i \) for \( t e \overline{K}_+ \) is given by

\[
\text{LB}_t = \min \left( \min_{T e T_1} [N(T) - N(T + t)], \min_{T e T_2} [N(T) - N(T + t - s) + \text{LB}_s] \right) \tag{1.32}
\]

Next, let us derive lower and upper bounds on the lifted coefficients \( \pi_i \), for any \( s e \overline{K}_+ \). From (1.22), we have that when \( T = \{ s \} \cup \{ \overline{K}_+ - t \} \), where \( t \equiv j(M_i) \), \( \pi_i - \pi_s \leq N(T) \). Hence, we have \( \pi_i \geq \pi_s - N(T) \). Consequently, for a given lower bound \( \text{LB}_t \) on \( \pi_i \), \( t e \overline{K}_+ \), a lower bound \( \text{LB}_s \) for any \( s e \overline{K}_+ \) is given by

\[
\text{LB}_s = \text{LB}_t - N(\overline{K}_+ - t + s) \text{ where } t \equiv j(M_s). \tag{1.33}
\]

**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 \( \pi_i \) or the \( \pi_s \) coefficients. Otherwise, we need to derive unconditional lower bounds \( \text{LB}_t \) \( \forall j e \overline{K}_+ \). Toward this end, we derive an unconditional lower bound of zero on \( \pi_i \), \( \forall t e \overline{K}_+ \), 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, \( \dim(\text{GUBKP}) < n \)), it can be readily shown that the following inequality

\[
\sum_{j e K_+} z_j + \sum_{j e K_+} \pi_j z_j + \sum_{i e M_k} \sum_{j e (N_i - j(0))} (\pi_{j(0)} - \pi_j) z_j \leq |M_k| - 1 \tag{1.34}
\]

is a facet of the convex hull of the polytope.
\[ Z \equiv \text{conv}\{z \in (0,1)^n : \sum_{i \in M} \sum_{j \in N_i} \tilde{a}_{ij} z_j \leq \tilde{b}, \sum_{j \in N_i} z_j \leq 1 \ \forall i \in M\} \]

where for all \( i \in M \), \( \tilde{a}_{ij} = a_{ij}, \tilde{a}_{i0} = a_{i0} - a_i \ \forall j \in \{N_i - j(i)\} \), and \( \tilde{b} = b + \sum_{j \in N_i} a_j \). 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 \( t \in \bar{\kappa}_+ \), we have

\[
\pi_t \geq 0, \quad \text{and} \quad \pi_{t^+} - \pi_s \geq 0, \quad \forall s \in (N_p - t), \quad \text{where} \ p = M_t. \tag{1.35}
\]

From (1.35), we have a valid lower bound of zero on \( \pi_{s_t} \), \( \forall t \in \bar{\kappa}_+ \). Consequently, from (1.33), we have that \( \text{LB}_s = -N(\bar{\kappa}_+ - t + s) \forall s \in (N_p - t), \) where \( p \equiv M_t. \]

Finally, let us derive an upper bound \( \text{UB}_s \) for any given \( s \in \bar{\kappa}_- \). Note from (1.22) that the collection of constraints defining \( \Pi_\kappa \) that contain the coefficient \( \pi_s \) are given by

\[
\pi_s \geq \sum_{j \in (\bar{\kappa}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \bar{\kappa}_-} \pi_j - N(T) \quad \forall T \in F_\kappa \equiv s \in T. \tag{1.36}
\]

Again, at any vertex of \( \Pi_\kappa \), since at least one of (1.36) must be binding, we have that

\[
\pi_s = \max \{ \sum_{j \in (\bar{\kappa}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \bar{\kappa}_-} \pi_j - N(T) : \ T \in F_\kappa \equiv s \in T \}. \tag{1.37}
\]

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

\[
\sum_{j \in (\bar{\kappa}_+ - T)} \pi_j - \sum_{j \in (T - s) \cap \bar{\kappa}_-} \pi_j \leq N(T - s).
\]
On the other hand, if \((T - s) \notin F_k\), then \(N(T - s) = \infty\), 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 \(t = j(M_s)\). Consequently, from (1.22), the corresponding constraint for \(T' = (T - s + t) \in F_k\) yields

\[
\left[ \sum_{j \in (K_+ - T)} \pi_j - \sum_{j \in (T - s) \cap K_-} \pi_j \right] = \left[ \sum_{j \in (K_+ - T')} \pi_j - \sum_{j \in T' \cap K_-} \pi_j \right] + \pi_t \leq N(T') + \pi_t \leq N(T') + UB_t.
\]

Note that if \(\overline{K_T} \equiv (K_+ - T') \cup (T' \cap K_-) = \emptyset\), we simply have \(N(T') = 0\) in that case.

Combining the last two inequalities, we may write for any \(T \in F_k \ni s \in T\), and \(t = j(M_s)\),

\[
\left[ \sum_{j \in (K_+ - T)} \pi_j - \sum_{j \in (T - s) \cap K_-} \pi_j \right] \leq \min\{N(T - s), N(T - s + t) + UB_t\}.
\]

Substituting this in (1.37) above, we obtain the following upper bound \(UB_s\) on \(\pi_s\) for any \(s \in K_-\).

\[
UB_s = \max_{T \in F_k \ni s \in T} \left\{ \min\{N(T - s), N(T - s + t) + UB_t\} - N(T) \right\}, \quad (1.38)
\]

where \(t = j(M_s)\).

**Example 1.4** Consider the following constraints of a GUB constrained knapsack problem, where \(X \equiv \{x \in (0, 1)^n\} \ni 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 \( \pi_2 \), where \( \{2\} \in \overline{K}_- \). Since \( \eta^2 = 4 \), we have that \( UB_2 = \eta^2 - 1 = 3 \). Furthermore, the set \( T_1 \equiv \{T: \exists T \in F_k \in \{2\} \not\in T \text{ and } (T + \{2\}) \in F_k \} \), is given by \( T_1 = \{(3, 6, 8), (4, 6, 8)\} \). Also the set \( T_2 \equiv \{T: \exists T \in F_k \in \{2\} \not\in T \text{ and } \{1\} \in T \} \) is given by \( T_2 = \{(1, 4, 6), (1, 4, 8), (1, 6, 8), (1, 3, 5, 8), (1, 3, 6, 7), (1, 6, 8), (1, 4, 5, 8), (2, 4, 6, 7), (1, 4, 6, 8)\} \). 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 \( \pi_1 \) in any lifted facet (1.18). Now, the set \( T' \equiv \{T: \exists T \in F_k \in \{1\} \in T \} \) 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),

\[
UB_1 = \max_{T \in T'} \{ \min_{T \in T'} [N(T - \{1\}), N(T - \{1\} + \{2\}) + 3] - N(T)\} = 2.
\]

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 \( \pi_1 \) and \( \pi_2 \) 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 \( |N| = 1 \ \forall i \in M \), which represents the ordinary knapsack polytope, denoted by KP. That is, \( KP \equiv \text{conv}\{x \in \{0,1\}^n : \sum_{i \in N} a_i x_i \geq b\} \), where...
the data are all integer, \( N = \{1, \ldots, n\} \), \( b > 0 \), \( 0 < a_i \leq b \quad \forall i \in N \), and \( \sum_{j \in K} a_i \geq b \) for all \( k \in N \). Recall that the minimal (GUB) cover inequality \( \sum_{j \in K} x_j \geq 1 \) is a facet of \( \text{conv}(KP(K)) \), where \( KP(K) = KP \cap \{ x \in \{0, 1\}^n : x_i = 1, \quad \forall i \in K \} \). Our interest is in characterizing a (simultaneously) lifted facet, as in Balas and Zemel (1978), of the form

\[
\sum_{j \in K} x_j + \sum_{i \in K} \pi_i x_j \geq 1 + \sum_{i \in K} \pi_i, \quad (1.39)
\]

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

\[
f(\theta) = \min \left\{ \sum_{j \in K} y_j : \sum_{j \in K} a_j y_j \geq b + \theta \right\} - 1,
\]

where \( b = b - \sum_{j \in K} a_j \).

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

\[
\Pi_K \equiv \{ (\pi) : \sum_{j \in (K - T)} \pi_j \leq f(\sum_{j \in (K - T)} a_j) \quad \forall T \in F_K \} \quad (1.40)
\]

and where \( F_K = \{ T \subseteq K : \sum_{j \in K \setminus T} a_j \geq b \} \).

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 \( \pi_i, j \in K \), for such facets of \( KP \).
**Upper bound UBₜ on πₜ, teK**

From (1.40), by examining T = K - {t} eFₖ, we directly have that πₜ ≤ f(aₜ). Hence, an upper bound UBₜ on πₜ is given by

\[ UBₜ = f(aₜ) \quad \text{for each teK}. \]

Note that UBₜ is the same as Balas and Zemel’s upper bound on πₜ, teK.

**Lower bound LBₜ on πₜ, teK**

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

\[ LBBZₜ = h \forall teSₜ, \]

where letting E(K) denote the extension of K as before, we have Sₜ = E(K), and

\[ Sₜ = \{te(E(K) - K) : \sum_{j \in Kₜ} a_j \leq aₜ < \sum_{j \in Kₜ+1} a_j \} , \]

where Kₜ is the index set of the h largest aⱼ for j \in K. Note that UBₜ = LBBZₜ or LBBZₜ + 1.

We now construct a tighter lower bound on πₜ. Consider any teK, and let us examine any TeFₖ such that t \notin T. From (1.40), we have that

\[ πₜ \leq f( \sum_{j \in (K - T)} aⱼ) - \sum_{j \in (K - (T - \{t\})} πⱼ \forall TeFₖ, \{t\} \notin T. \] (1.41)

But since at an extreme point of Πₓ, at least one of (1.41) must be binding, we have that
\[ \pi_t = \min \{ f( \sum_{j \in (K - T)} a_j ) - \sum_{j \in (K - \{t\})} \tau_j : T \in F_K \not= \{t\} \} \]

given other \( \pi_j \) values. Consequently, we get

\[ \text{LB}_t = \min \{ f( \sum_{j \in (K - T)} a_j ) - f( \sum_{j \in (K - \{t\})} a_j ) : T \in F_K \not= \{t\} \} \]  \hspace{1cm} (1.42)

**Proposition 1.18**  \( \text{LB}_t \geq \text{LBBZ}_t \), \( \forall t \in K \).

**Proof.** By the monotone increasing nature of the function \( f \), it follows that \( \text{LB}_t \geq 0 \) \( \forall t \in K \). Since \( \text{LBBZ}_t = 0 \) for \( t \in E(K) \), the result holds trivially for this case. Hence, suppose that \( t \in (E(K) - K) \). Consider any \( T \in F_K \) with \( t \not\in T \), and examine two cases.

**Case (i):** \( a_i = \sum_{j \in \mathbb{N}_K} a_j \). It follows that \( \text{LBBZ}_t = h \equiv f(a_t - \hat{b}) + 1 \). But we have, defining \( \hat{b} = b - \sum_{i \in T} a_i - a_t > 0 \), that

\[ f( \sum_{j \in (K - T)} a_j ) - f( \sum_{j \in (K - \{t\})} a_j ) = \min \{ \sum_{j \in K} y_j : \sum_{j \in K} a_j y_j \geq \hat{b} + a_t \} - \min \{ \sum_{j \in K} y_j : \sum_{j \in K} a_j y_j \geq \hat{b} \} \]

\[ \geq \min \{ \sum_{j \in K} y_j : \sum_{j \in K} a_j y_j \geq a_t \} = f(a_t - \hat{b}) + 1. \]  \hspace{1cm} (1.43)

This implies from (1.42) that \( \text{LB}_t \geq \text{LBBZ}_t \).
Case (ii) : \( \sum_{j \in K_h} a_j < a_t < \sum_{j \in K_0, \ldots, 1} a_j \). In this case, we have, \( \text{LBB}_i = h \equiv f(a_i - \overline{b}) \). Let \( \Delta \) 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

\[
\begin{align*}
&f(\sum_{j \in (K - T)} a_j) - f\left(\sum_{j \in (K - T) \setminus \{t\}} a_j\right) \\
&\geq f(\sum_{j \in (K - T)} a_j - \Delta) - f(\sum_{j \in (K - T) \setminus \{t\}} a_j) \\
&\geq f(a_t - \Delta - \overline{b}) + 1 = f(a_t - \overline{b}).
\end{align*}
\]

Hence from (1.42) and (1.44), we have that \( \text{LB}_t \geq \text{LBB}_i \). Therefore, the result holds for any \( t \in (E(K) \setminus K) \) as well, and this completes the proof. \( \blacksquare \)

**Example 1.5** Consider \( KP = \text{conv}\{xe(0, 1)^4 : 3x_1 + 3x_2 + 3x_3 + 2x_4 \geq 7\} \). Let \( K = \{1, 2\} \), so that \( E(K) = \{1, 2, 3\} \) and \( \overline{b} = 2 \). Consider \( t = \{4\} \in \overline{E(K)} \). Note that \( \{T \in F_K : t \not\in T\} = \{3\} \). Hence, from (1.42), we have that \( \text{LB}_4 = f(a_4) - f(0) = f(a_4) = UB_4 = 1 > \text{LBB}_4 = 0 \).

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 \( \text{LB}_t = 0 \equiv \text{LBB}_t \) as well.

**Example 1.6** Consider \( KP = \text{conv}\{xe(0, 1)^4 : 3x_1 + 3x_2 + 3x_3 + 4x_4 \geq 7\} \). The minimal cover \( K = \{1, 2, 3\} \) is a strong cover for \( KP \). Since \( \overline{b} = 3 \), \( \overline{K} = \{4\} \), and for \( t = \{4\} \in (E(K) - K) \), we have that \( \{T \in F_K : t \not\in T\} = \phi \), this gives \( \text{LB}_4 = f(a_4) - f(0) = f(a_4) = UB_4 = 2 \). However, \( \text{LBB}_4 = 1 < \text{LB}_4 \).
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 \( \sum_{j \in C} a_j \leq b - 1 \), where \( \overline{C} \equiv N - C \). 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 \( D \equiv \overline{C} + \{k\} \). Accordingly, we define a minimal cover inequality as

\[
\sum_{j \in C} x_j \geq 1
\]  

(1.45)

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

**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
\[ \sum_{j \in E(C)} x_j \geq 1 + |M_R| \]  \hspace{1cm} (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 \( \sum_{j \in C_+} a_j < b \) implies that even with such a solution \( \bar{x} \equiv \{ \bar{x}_j = 1, \forall j \in R_+ \}, \bar{x}_j = 1, \forall j \in (\bar{C} - \bar{S}) \}, \) 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 \( \sum_{j \in C} x_j \geq 1 + \sum_{i \in M_R} (1 - \sum_{j \in (N - C)} x_j) \) Since \( \sum_{j \in (N - C)} x_j \leq 1 \) 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 \( \sum_{i \in M_R} a_i \leq b - 1 \). This implies that \( H \) is a cover set. If \( H \) is not a minimal cover, then there exists some \( k \in H \) 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 \( X \equiv \{ x \in (0, 1)^N : x_1 + 5x_2 + x_3 + 5x_4 + x_5 + 2x_6 + x_7 + 2x_8 \geq 9, x_1 + x_2 \leq 1, x_3 + x_4 \leq 1, x_5 + x_6 \leq 1, x_7 + x_8 \leq 1 \}. \) Since \( \sum_{i \in \{1, \ldots, N\}} a_i \geq b \) 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.

\[ \text{SP} : \text{Minimize } \{c^T x : Ax = e, x_j = 0 \text{ or } 1 \ \forall j \in \mathbb{N}\} \]

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

Problem \( \text{SP} \) 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 \( \text{SP} \) are given by Garfinkel and Nemhauser (1972), Salkin (1975), Fisher and Kedia (1990), and Chan and Yano (1992). Two well-known approaches to problem \( \text{SP} \) 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 \( \text{SP} \) 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

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

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

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

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 \(d \in \{0, \ldots, n\}\), 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.

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

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

Using this notation, problem $SPP$ can be restated as follows.

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

subject to

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

$$x_j \text{ binary} \quad \forall \ j \in N \quad (2.2)$$

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

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

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 $x_1 x_2$, $x_1(1 - x_2)$, $(1 - x_1)x_2$, $(1 - x_1)(1 - x_2)$. For con-
venience, we will take $F_d(\phi, \phi) = 1$, 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 $\text{SPP}_\delta$ of $\text{SPP}$, 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 $x_i^2 \equiv x_i$ and so $x_i(1 - x_i) = 0$ for each binary variable $x_i$, $j = 1, \ldots, n$, this gives the following set of constraints.

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

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

**STEP 2.** Linearize the constraints in (2.4) and (2.5) by substituting the following variables for the corresponding nonlinear terms for each $J \subseteq N$, $w_j \equiv \prod_{i \in J} x_i$, where we assume the notation that $w_j \equiv x_j$ for $j = 1, \ldots, n$, and $w_j \equiv 1$. Furthermore, denoting by $f_d(J)$ and $f_d(J, J_2)$ 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 $\text{SPP}_\delta$, where

\[ \text{SPP}_\delta = \{ (x, w) : \]
\[ \left[ N_i \cap J \right] f_d(J) + \sum_{j \in (N_i - J)} f_{d+1}(J + j) = f_d(J) \quad \forall i \in M \right], \quad \forall J \subseteq N \ni |J| = d, \quad d = 0, \ldots, \delta \quad (2.6) \]

\[ f_d(J_1, J_2) \geq 0, \quad (J_1, J_2) \) of order d, \quad d = 1, \ldots, \min(\delta + 1, n) \right) \right). \quad (2.7) \]

**Remark 2.1** Note that for the case $\delta = 0$, using the fact that $f_0(\phi, \phi) \equiv 1$, and that $f_i(j, \phi) \equiv x_i$ and $f_i(\phi, j) \equiv (1 - x_i)$ for $j = 1, \ldots, n$, it follows that $\text{SPP}_0$ given by (2.6) and (2.7) is $\overline{\text{SPP}}$. Also note that for any $d \in \{1, \ldots, \delta\}$,

\[ f_d(J_1, J_2) \equiv f_{d+1}(J_1 + k, J_2) \quad \text{where } k \in N - (J_1 \cup J_2). \]

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

\[ \text{SPP}_{P_\delta} = \{ x : (x, w) \in \text{SPP}_\delta \} \]

the projection of the set $\text{SPP}_\delta$ 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 $\text{SPP}_{P_\delta}$ represent a sequence of nested, valid relaxations leading up to the convex hull representation. That is,

\[ \text{SPP} \equiv \text{SPP}_{P_0} \subseteq \text{SPP}_{P_{n-1}} \subseteq \ldots \subseteq \text{SPP}_{P_1} \subseteq \text{SPP} \equiv \overline{\text{SPP}}. \]
Due to the structure of the set partitioning polytope, we can further reduce the size of the constraints (2.7) of SPP$_{d}$, $d \in \{0,\ldots,n\}$. Toward this end, consider the following set of constraints associated with SPP$_{d}$, $d \in \{0,\ldots,n\}$.

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

**Proposition 2.1** For any $d \in N$, the constraints (2.6) and (2.8) imply the constraints (2.7) in SPP$_{d}$.

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

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

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

where $[F(x)]_L$ denotes the linearized term of the polynomial $F(x)$ as defined at Step 2 of the RLT procedure. For some $i \in M_n$, we have a constraint in (2.6) as follows.

$$[F_{d-1}(J)] \left( x_k + \sum_{j \in \mathcal{N}_i - \{k\}} x_j \right) - F_{d-1}(J_1) = 0 \text{ or,}$$

$$f_{d}(J_1 + \{k\}) + \left[ \sum_{j \in \mathcal{N}_i - \{k\}} x_j F_{d-1}(J_1) \right]_L = f_{d-1}(J_1). \quad (2.10)$$

From (2.8), (2.9) and (2.10), it follows that
\[ f_d(J_1, J_2) = \left[ \sum_{j \in (N_1 - \{k\})} x_j f_{d-1}(J_1) \right]_L \]
\[ = \sum_{j \in (N_1 - \{k\} - J_1)} f_d(J_1 + \{j\}) + \sum_{j \in (N_1 - \{k\}) \cap J_1} f_{d-1}(J_1) \geq 0. \]

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

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

For some \( i \in \mathcal{M}_k \), we have the set partitioning constraint, \( x_i + \sum_{j \in (N_1 - \{k\})} x_j = 1 \). Since (2.6) includes constraints obtained by multiplying the foregoing constraint with all factors \( F_d(J), |J| = 0, 1, \ldots, \delta \), and since \( \prod_{j \in J_1} \prod_{j \in (J_2 - \{k\})} (1 - x_j) \) is a linear combination of such factors, it follows that (2.6) gives

\[ \left[ \left( \sum_{j \in (N_1 - \{k\})} x_j \prod_{j \in J_1} \prod_{j \in (J_2 - \{k\})} (1 - x_j) \right)_L \equiv \left[ \prod_{j \in J_1} x_j \prod_{j \in (J_2 - \{k\})} (1 - x_j) \right]_L \equiv f_d(J_1, J_2). \]

Letting \( J'_2 = (J_2 - \{k\}) \), the left-hand-side of the above equation is comprised of terms of the type \( f_{\text{min}_{d+1}}(J_1 \cup \{j\}, J') \) for \( j \in N_1 - \{k\} \equiv j \notin J_1 \cup J'_2 \), and of the type \( f_d(J_1, J'_2) \) for \( j \in N_1 - \{k\} \equiv j \in J_1 \), and zeroes in case \( j \in (N_1 - \{k\}) \cap J'_2 \). Since \( |J'_2| = (p - 1) \), the induction hypothesis implies that all these terms are nonnegative. Hence, \( f_d(J_1, J_2) \geq 0 \) is also implied, and this completes the proof. \( \blacksquare \)
Remark 2.3  By Remark 2.1, note that we only need \( f_0(J_1, J_2) \geq 0 \ \forall (J_1, J_2) \) of order \( D = \min(\delta + 1, n) \) in (2.7) to obtain the same representation of SPP\(_\delta\). 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 \( f_0 \geq 0 \) and \( f_{\delta-1}(J) \geq 0 \), where \( D = \min(\delta + 1, n) \), in (2.8) for obtaining the same representation of SPP\(_\delta\). 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\(_\delta\), \( \delta \in \{0, \ldots, n\} \), can be restated as follows.

\[
\text{SPP}_\delta = \{ (x,w) : \sum_{J \subseteq \mathbb{N}} f_d(J) + \sum_{J \subseteq \mathbb{N} \setminus \{J\}} f_{d+1}(J + j) = f_d(J) \ \forall i \in \mathbb{M} \}, \quad \forall J \subseteq \mathbb{N} \ \exists |J| = d, \ d = 0, \ldots, \delta \quad (2.11)
\]

\[
f_d(J) \geq 0 \ \forall J \subseteq \mathbb{N} \ \exists |J| = d, \ d = 1, \ldots, \min(\delta + 1, n) \quad (2.12)
\]

The special structure of SPP\(_\delta\) 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 \( J \subseteq \mathbb{N}, |J| = d \in \{1, \ldots, \delta\} \), and define \( M_d = \bigcup_{J \subseteq \mathbb{N}} M_J \). Then \( f_{\delta+1}(J + j) = 0 \) in (2.11) \( \forall j \in \mathbb{N} \setminus J \ \forall i \in \mathbb{M}_J \). Moreover, if \( |J \cap N_i| > 1 \) for any \( i \in \mathbb{M}_J \), then \( f_d(J) = 0 \) also.

**Proof.**  Consider the constraint (2.11) for any \( i \in \mathbb{M}_J \), and any \( J \subseteq \mathbb{N}, |J| = d \in \{0, \ldots, \delta\} \). Since \( |J \cap J| \geq 1 \), the nonnegativity constraints (2.12) imply that \( f_{\delta+1}(J + j) = 0 \), \( \forall j \in \mathbb{N} \setminus J \ \forall i \in \mathbb{M}_J \). 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. 

The Polyhedral Structure of Combinatorial Optimization Problems 55
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

\[
\tilde{M}_J = M - \bigcup_{j \in J} M_j \equiv \bigcap_{j \in J} \bar{M}_j \equiv \{ i \in M : N_i \cap J = \emptyset \}
\]

\[
N_{ij} = \{ j \in N_i : a_j^T a_k = 0 \}, \ i \in \bar{M}_k.
\]

\[
\tilde{N}_{ij} = \bigcap_{j \in J} N_{ij}, \ i \in \bar{M}_k.
\]

Hence, \( \tilde{M}_J \) represents the set of rows in \( M \) not containing any variables from the set \( J \), and \( \tilde{N}_{ij} \) 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

\[
SPP_\delta = \{ (x, w) : \sum_{j \in N_i} x_j = 1 \quad \forall i \in M \quad (2.13) \}
\]

\[
[ \sum_{t \in \tilde{N}_{ij}} w_{J \cup \{t\}} = w_J \quad \forall i \in \tilde{M}_J ] \quad \forall |J| = d, \quad d = 1, \ldots, \delta \quad (2.14) \}
\]

\[
[w_J \geq 0, \quad w_{J \cup \{t\}} \geq 0 \quad \forall t \in \tilde{N}_{ij}, \quad \forall i \in \tilde{M}_J ] \quad \forall |J| = d, \quad d = 1, \ldots, \delta \quad (2.15) \}
\]

where \( w_j \) and \( w_{J \cup \{t\}} \) are the linearized terms for the products \( \prod_{j \in J} x_j \) and \( \prod_{j \in J \cup \{t\}} x_j \), 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 \(a_i^\top a_j \neq 0\). 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, \(SPP \equiv SPP_{\alpha(G)}\).

**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 \(SPP_{|M_0|}\), where \(|M_0|\) 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 SPP₁. 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 SPP₁ of SPP, given by (2.13) - (2.15), as follows. Note that in this relaxation, \( w_{jk} \) is the linearized term for the product \( x_j x_k \). We will denote \( w_{jk} \) to be \( w_k \) if \( j < k \) and \( w_i \) otherwise.

\[
\text{SPP₁} = \{ (x, w) : \quad \sum_{j \in N_i} x_j = 1 \quad \forall \ i \in M \tag{2.16} \\
\sum_{j \in N_{i,k}} w_{jk} = x_k \quad \forall \ i \in \overline{M}_k, \quad \forall \ k \in N \tag{2.17} \\
[x_k \geq 0, \ w_{jk} \geq 0 \ \forall \ j \in N_{i,k}, \ \forall \ i \in \overline{M}_k, \ \forall \ k \in N] \}. \tag{2.18}
\]

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 SPP₁ is feasible to SPP, we have that SPP₁ ⊆ SPP. Hence, letting \( v(P) \) denote the optimal objective function value for any problem \( P \), \( v(SPP) \geq v(SPP₁) \), where SP₁ : Minimize \( \{ cx : (x, w) \) satisfies (2.16), (2.17), (2.18) \}, and where SPP is the linear programming relaxation of SP.

Similarly, by (2.13) and (2.14), we can write the second-order (\( \delta = 2 \)) RLT formulation SPP₂ of SPP as follows. Note here that \( w_{jk} \) is the linearized term for the product \( x_j x_k \), for \( j < k < l \), and where the indices are not necessarily so arranged, we simply write this product term as \( w_{jkl} \).

\[
\text{SPP₂} = \{ (x, w) : \quad \sum_{j \in N_i} x_j = 1 \quad \forall \ i \in M \tag{2.19} \\
\sum_{j \in N_{i,k}} w_{jk} = x_k \quad \forall \ i \in \overline{M}_k, \quad \forall \ k \in N \tag{2.20} \\
[x_k \geq 0, \ w_{jk} \geq 0 \ \forall \ j \in N_{i,k}, \ \forall \ i \in \overline{M}_k, \ \forall \ k \in N] \}. \tag{2.21}
\]
\[
\sum_{j \in N_i,k} w_{(j,k)} = x_k \quad \forall \ i \in \overline{M}_k, \ \forall \ k \in N \tag{2.20}
\]
\[
\sum_{j \in (N_i,k \cap N_i,l)} w_{(j,l)} = w_{(k,l)} \quad \forall \ i \in \overline{M}_k \cap \overline{M}_l, \ \forall \ k < l \in N \tag{2.21}
\]
\[
[x_i \geq 0, \ w_{(k,l)} \geq 0 \ \forall \ j \in (N_i,k \cap N_i,l), \ \forall i \in (\overline{M}_k \cap \overline{M}_l)] \quad \forall k < l \in N \quad \}. \tag{2.22}
\]

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

\[
x_k - \sum_{j \in N_i,k} x_j \leq 0
\]

are satisfied by all \(x \in \text{SPP}\).

**Proof.** For each \(k \in N\), consider the constraint (2.17). By the nonnegativity constraint (2.18), we have that \(w_{(j,k)} \leq x_k \ \forall \ j \in N_i,k\). Moreover, given any \(j \in N_i,k\) for \(i \in \overline{M}_k\), consider the constraint (2.17). Since \(j \in N_i,k\), we have that \(k \in N_i,j\) for some \(i \in \overline{M}_j\). Hence, we have that \(w_{(j,k)} \leq x_j\) all \(j \in N_i,k\) for all \(i \in \overline{M}_k\). Hence, the constraint (2.17) implies that \(x_k = \sum_{j \in N_i,k} w_{(j,k)} \leq \sum_{j \in N_i,k} x_j\). Hence, it follows that \(x_k - \sum_{j \in N_i,k} x_j \leq 0\). This completes the proof. \(\blacksquare\)
Remark 2.5 The inequality (2.23) asserts that when $x_k = 1$ at least one of the variables $x_i, i \in N_k$, 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.,

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

By Proposition 2.2, we have that $w_{(k)} = 0$ 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 $w_{(k)} = 0$ 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^* = \{ j \in L(k) : 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 \( N_k \) is as hard as finding an optimal solution to \( \text{SP} \). However, we can easily construct a subset of \( N_k \) for some \( k \in \mathbb{N} \) as follows.

Recall the following constraints (2.21) of \( \text{SPP}_k \):

\[
w_{(kl)} = \sum_{j \in (N_{i,k} \cap N_{i,j})} w_{(jl)} \quad \forall \ i \in \overline{M}_k \cap \overline{M}_l, \quad \forall \ k < l \in \mathbb{N}.
\]

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

\[
z_{(ko)} = \max \{ w_{(ko)} : w_{(ko)} = \sum_{j \in (N_{i,k} \cap N_{i,j})} w_{(jl)}, \quad \forall \ i \in \overline{M}_k \cap \overline{M}_l, \quad w_{(jl)} \geq 0 \ \forall (j,l), \ \ w_{(ko)} \leq 1 \ \forall (k,l) \}.
\]

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

The above procedure for detecting a set of zero \( w_{(ko)} \)'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 \( k \in \mathbb{N} \), \( i \in \overline{M}_k \), and \( Q \subseteq N_{ik} \), the following inequality

The Polyhedral Structure of Combinatorial Optimization Problems
\[ x_k - \sum_{j \in Q} x_j \leq 0 \]  

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

**Proof.** Suppose that (2.25) is valid for SPP. Then, the inequality \( x_k - \sum_{j \in Q} w_{(k)} \leq 0 \) is valid for SPP. From equation (2.17), we have that \( w_{(k)} = 0 \) for all \( j \in (N_{i,k} - Q) \) because \( w_{(k)} \geq 0 \ \forall \ j, k \). On the other hand, if \( w_{(k)} = 0 \) for all \( j \in (N_{i,k} - Q) \), then the equality (2.17) becomes \( \sum_{j \in Q} w_{(k)} = x_k \). 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 \( Q_k \subseteq N_{i,k}, \ i \in \bar{M}_k \), be such that the inequalities \( x_k - \sum_{j \in Q_k} x_j \leq 0 \), \( i \in \bar{M}_k \) are satisfied by all \( x \in P \). For each \( j \in \bigcup_{h \in M_k} Q_{h} \), define \( Q ( j ) = \bigcup_{h \in \bar{M}_k, j \in Q_h} Q_{h} \) and for \( i \in \bar{M}_k \), let \( T_i = \{ j \in Q_i : Q_{k} \subseteq Q ( j ) \} \) for some \( h \in \bar{M}_k \). Then, the inequalities \( x_k - \sum_{j \in Q(T_i)} x_j \leq 0 \ \forall \ i \in \bar{M}_k \) are satisfied by all \( x \in SPP \). ■

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

\[
\begin{array}{cccccccccccccccc}
& 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Consider the strengthening procedure of Proposition 2.6 to strengthen the inequality \( x_i - x_j - x_{ij} \leq 0 \) associated with \( N_{j,i} \), where for \( k = 1 \), \( \bar{M}_1 = \{3,4,5\} \), and \( N_{1,j} = \{3,12\} \), \( N_{4,1} = \{3,4\} \), and \( N_{3,4} = \{3,4,5,12\} \). 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 \( x_i - x_j \leq 0 \).

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

\[
\begin{align*}
x_i &= w_{1,3} + w_{1,12} & \text{for } i = 3 \\
x_i &= w_{1,3} + w_{4,4} & \text{for } i = 4 \\
x_i &= w_{1,3} + w_{1,4} + w_{1,5} + w_{1,12} & \text{for } i = 5 
\end{align*}
\]

Consider the form of equality (2.21) for \( l = 12 \) in the corresponding second-order RLT formulation. Since \( \bar{M}_{12} = \{1,2,4\} \), we have that \( \bar{M}_1 \cap \bar{M}_{12} = \{4\} \) and \( N_{k,12} = \{1,9,11,13,14\} \), \( N_{4,4} = \{3,4\} \). Therefore, \( N_{k,12} \cap N_{4,4} = \phi \). Hence, the inequality of type (2.21) for \( k = 1 \) and \( l = 12 \) is
\[ w_{1,12} = \sum_{l \in N_{k,4} \cap N_{k,12}} w_{1,12,l} = 0 \]  

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

\textbf{Example 2.2} (Example 3.2 in Balas, 1977) Consider the following numerical example.

\[
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

For \( k = 1 \), we have that \( \overline{M}_1 = \{3, 4, 5, 6, 7\} \), and \( N_{k,1} = \{2, 5, 7\} \), \( N_{k,1} = \{2, 6, 8\} \), \( N_{k,1} = \{3, 5, 8\} \), \( N_{k,1} = \{3, 4, 6\} \), and \( N_{k,1} = \{4, 5, 7\} \).

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 \( N_{5,1} \), \( x_1 - x_2 - x_6 - x_7 \leq 0 \) can be strengthened to \( x_1 - x_2 - x_7 \leq 0 \). ■

We now consider Balas’ second strengthening procedure.

**Proposition 2.7** (Proposition 3.2, Balas, 1977) Let the index sets \( Q_k \subseteq N_k, \ i \in \overline{M}_k, \ k \in \mathbb{N} \), be such that the inequalities \( x_k - \sum_{i \in Q_k} x_i \leq 0, \ i \in \overline{M}_k, \ k \in \mathbb{N} \) are satisfied by all \( x \in \text{SPP} \). For \( i \in \overline{M}_k, \ k \in \mathbb{N} \), define

The Polyhedral Structure of Combinatorial Optimization Problems 65
\begin{equation}
U_{ik} = \{ j \in Q_{ik} : Q_{hk} \cap Q_{kj} = \phi \text{ for some } h \in \overline{M}_k \cap \overline{M}_j \}. \tag{2.26}
\end{equation}

Then, the inequalities

\[ x_k - \sum_{j \in Q_{ik} \cup U_{ik}} x_j \leq 0, \text{ for } k \in \mathbb{N}, \text{ are satisfied by all } x \in \text{SPP}. \]

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 \( w_{qk} = 0 \) 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 \( w_{qk} = w_{q(k)} \) for some \( l \) and that \( w_{q(k)} = \sum_{i \in N_{ik} \cap N_{kl} \cap N_{ll}} w_{qli} \), for \( i \in \overline{M}_i \cap \overline{M}_k \cap \overline{M}_l \). If \( N_{ik} \cap N_{kl} \cap N_{ll} = \phi \), then \( w_{q(k)} = 0 \) and consequently \( w_{q(k)} = 0 \). In other words, for a given \( \delta \), we can tighten the formulation of SPP by using the logical implications of the structure of SPP, for \( \delta = 1, \ldots, \alpha(G) - 1 \). Also note that Balas’ strengthening procedures use only partial information regarding the logical implications of SPP for tightening SPP. On the other hand, if one has the facility to handle SPP itself directly, then stronger relaxations can be enforced via such an explicit representation of SPP.

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

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

(2.27)

where \(a_i a_j = 0 \; \forall \; j \in S\). This subclass of elementary inequalities is distinguished by the additional property that \(S \subseteq \mathbb{N}_k\) for some \(i \in \mathbb{M}_k\). 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 \(k, h \in \mathbb{N}\), let \(S_r \subseteq L(r)\), \(r = k, h\), be such that \(h \in S_k, \; k \notin S_h\), and that the inequalities

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

(2.28)

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

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

(2.29)

where \(S = (S_h \setminus \{h\}) \cup [S_h \cap L(k)]\). Furthermore, (2.29) is stronger than the sum of the two inequalities (2.28) if and only if \(S_h \cap [S_h \cup L(k)] \neq \phi\). □
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 $w_{ik} = 0$ for $j \in N_k - S_k$. We then have the following first-order RLT formulation associated with the constraints (2.28).

\[ x_k = \sum_{j \in S_k} w_{(jk)} = w_{(kh)} + \sum_{j \in S_k \setminus \{h\}} w_{(jk)} \quad (\text{since } i \in S_k) \quad (2.30) \]

\[ x_h = \sum_{j \in S_n} w_{(jh)} \quad (2.31) \]

Hence, we have that

\[ w_{(jk)} = 0 \quad \forall j \in S_k \setminus \{h\}. \quad (2.32) \]

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

\[ w_{(kh)} = \sum_{j \in S_n} w_{(jkh)}. \]

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

\[ w_{(kh)} = \sum_{j \in (S_n \cap L(k) - [S_k \setminus \{h\}])} w_{(jkh)}. \]

By substituting $w_{(kh)}$ in (2.30), we have that
\[
x_k = \sum_{j \in (S_n \cap L(k) - [S_n \setminus \{h\}])} w_{j(kn)} + \sum_{j \in (S_n \setminus \{h\})} w_{j(k)}.
\]

Since \( w_{j(kn)} \leq x_j \) and \( w_{j(k)} \leq x_j \), (2.33) implies that

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

Equivalently,

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

where \( S = (S_n \setminus \{h\}) \cup (S_n \cap L(k)) \).

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 \( k \in \mathbb{N} \) from the set of elementary inequalities (2.23) corresponding to the same index \( k \).

**Proposition 2.9** (Proposition 5.2, Balas, 1977) For some \( k \in \mathbb{N} \), let the sets \( Q_i \subseteq N_{ik}, \ i \in M_k \), be such that the inequalities

The Polyhedral Structure of Combinatorial Optimization Problems
\[ x_k - \sum_{j \in Q_i} x_j \leq 0, \ \forall i \in \overline{M}_k \]  \hspace{0.5cm} (2.34)

are satisfied by all \( x \in \text{SPP} \), and let \( Q_0 = \bigcup_{i=0}^{k} Q_i \). Then the inequality

\[ x_k - \sum_{j \in S} x_j \leq 0, \]  \hspace{0.5cm} (2.35)

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

\[ \sum_{j \in Q} a_j \neq e - a_k, \ \forall Q \subseteq Q_0 \setminus S. \]  \hspace{0.5cm} (2.36)

Since (2.34) is valid for \( \text{SPP} \), by Proposition 2.5, it follows that \( w_{i0} = 0 \ \forall j \in N_{i,k} \setminus Q_i \), \( \forall i \in \overline{M}_k \). 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 \( \text{SPP} \) whenever \( x_k = 0 \). If \( x_k = 1 \), the inequality (2.35) is valid to \( \text{SPP} \) if and only if

\[ \text{maximum} \{1 - \sum_{j \in S} x_j : \sum_{j \in N_{i,k}} x_j = 1, \ \forall i \in \overline{M}_k, \ x_j \in (0,1)\} \leq 0. \]  \hspace{0.5cm} (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 \( \text{SPP} \) if \( x_k = 1 \) and \( x_i = 0 \ \forall i \in S \). 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 \( \text{SPP} \).

In \( \text{SPP}_k \), we have the following (implied) constraints for some \( k \in \mathbb{N} \).
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 $\mu_i$ and $\beta_j$ 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.

$$\sum_{i \in \overline{M}_k} \mu_i x_k - \sum_{i \in M_k} \mu_i \sum_{j \in Q_1} w_{(jk)} = 0$$

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

Hence, it follows that

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

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

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

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

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 $S \subseteq N \setminus \{k\}$ such
that there exists a feasible solution \((\mu, \beta)\) 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 \(\sum_{i \in M_k} \mu_i = 1\). Alternatively, using a surrogate of (2.38) and (2.39), we want to claim that there exists a feasible solution to the following system \(M\).

\[
M \equiv \{ \mu_i \text{ unrestricted } \forall i \in \bar{M}_k : \sum_{i \in (M_k \cap M_j)} \mu_i = 0 \forall j \in S, \sum_{i \in (M_k \cap M_j)} \mu_i = 0 \forall j \in (Q_0 - S), \sum_{i \in M_k} \mu_i = 1 \}.
\]

The above conjecture is illustrated by the following example.

**Example 2.3** Consider the following set-partitioning problem.

\[
\text{SP: Maximize } \{ \ x_1 + x_2 + x_3 + x_4 + x_5 + x_6: x_1 + x_2 + x_3 + x_7 = 1, x_1 + x_3 + x_4 + x_8 = 1, \\
x_1 + x_4 + x_5 + x_9 = 1, x_1 + x_5 + x_6 + x_{10} = 1, x_1 + x_9 + x_{11} = 1, x_j \in (0,1) \ j = 1, \ldots, 11 \}
\]

The optimal solution \(\bar{x}\) of \(\overline{SP}\) is

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

Note that no elementary inequality cuts off this optimal basic feasible solution \(\bar{x}\).

However, the above procedure generates a valid inequality which cuts off \(\bar{x}\). We can construct \(SP_1\) with respect to \(x_2\) by adding the constraints (2.38) and (2.39) to \(SP\). That is,

\[
\text{SP}_1(x_2) : \quad \text{Maximize } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
\text{subject to}
\]

The Polyhedral Structure of Combinatorial Optimization Problems
\[
\begin{align*}
x_1 + x_2 + x_3 + x_7 &= 1 \\
x_1 + x_3 + x_4 + x_8 &= 1 \\
x_1 + x_4 + x_5 + x_9 &= 1 \\
x_1 + x_5 + x_6 + x_{10} &= 1 \\
x_1 + x_2 + x_6 + x_{11} &= 1
\end{align*}
\]

\[
\begin{align*}
x_2 &= w_{2,4} + w_{2,8} \\
x_2 &= w_{2,4} + w_{2,5} + w_{2,9} \\
x_2 &= w_{2,5} + w_{2,10}
\end{align*}
\]

\[
\begin{align*}
w_{2,4} &\leq x_4 \\
w_{2,5} &\leq x_5 \\
w_{2,8} &\leq x_8 \\
w_{2,9} &\leq x_9 \\
w_{2,10} &\leq x_{10} \\
w_{2,k} &\geq 0 \quad k = 4, 5, 8, 9, 10 \\
x_j &\in \{0, 1\}, \quad j \in \{1, \ldots, 11\}
\end{align*}
\]

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

\[
x_2 - x_8 - x_{10} \leq 0
\] (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).

The Polyhedral Structure of Combinatorial Optimization Problems
\[ 2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2. \] (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.

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

\[ x_k - \sum_{j \in Q_0} \alpha_j x_j - \sum_{j \in Q_1} x_j \leq 0 \] (2.45)

where \( Q_F = \{ j \in Q_0 : 0 < \alpha_j \leq 1 \} \), and \( Q_1 = \{ j \in Q_0 : \alpha_j > 1 \} \).

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

\[ x_k - \sum_{j \in Q_F} x_j - \sum_{j \in Q_1} x_j \leq 0 \] (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 ($Q_r \cup C_n$ 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

$$\pi x \leq \pi_0$$

(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 $x \in SPP \setminus SPP$.

**Proposition 2.10** Let $\overline{x} = (\overline{x}_r, \overline{x}_f)$, where $F = \{ j \in N : \overline{x}_j$ is fractional $\}$ and $\overline{F} = N - F$, be a basic feasible solution of $\overline{SPP}$ with $F \neq \emptyset$. Then $SPP_i$ cuts off $\overline{x}$.

**Proof.** Suppose on the contrary that $\overline{x}$ remains feasible to $SPP_i$, along with some associated value $\overline{w}$ of $w$. Since $\overline{x}$ is a basic feasible solution, there exists a basis $B$ such that $B\overline{x}_B = e$, 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 $k$th element of $B^{-1}e$.

Since $(\overline{x}, \overline{w})$ must satisfy (2.17), and since $x_i = 0$ implies that $w_{(i)} = 0 \forall j$, we have $B\overline{w}_k = e\overline{x}_k \forall k \in B$, where $W_x = \{ w_{(i)}, j \in B \}$, and where $j \in B$ means that $x_j$ is basic. For each $k \in B \cap F$, since $\overline{w}_k = \overline{x}_k \neq 0$, and since $\overline{w}_k = (B^{-1}e)\overline{x}_k$, the $k$th component of
B⁻¹e is equal to one, i.e., \((B⁻¹e)_x = 1 \forall k \in B \cap F \neq \phi\), a contradiction. This completes the proof. ■

Proposition 2.10 shows that there exists a family of cutting planes inherent in SPP₁. We consider a procedure to generate a class of cutting planes, which cut off the fractional basic feasible solutions of SPP. For this purpose, we may write (2.17) and (2.18) as follows.

\[
Dw = Ex, \ w \geq 0, \ x \geq 0.
\] (2.48)

Associating Lagrangian multiplier vectors \(\pi\) with respect to (2.48), we have by linear programming duality that \(x \in \text{SPP}_1(x) = \{ x : (x,w) \text{ is feasible to SPP} \} \) if and only if (2.16) holds, \(x \geq 0\), and \(0 = \min\{0w : Dw = Ex, \ w \geq 0\}\), i.e.,

\[
0 = \max \{ \pi Ex : \pi \in PC \}
\] (2.49)

where \(PC\) is a polyhedral cone defined as

\[
PC = \{ \pi : \pi D \leq 0, \ \pi \text{ unrestricted} \}
\]

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 \(\bar{x}_i, \ i = 1, ..., L\) be the set of generators of the polyhedral cone \(PC\). Then (2.49) holds if and only if

\[
\bar{x}_i Ex \leq 0 \ \forall \ i = 1, ..., L.
\] (2.50)

In particular, for any \(\bar{x} \in PC\), the inequality

The Polyhedral Structure of Combinatorial Optimization Problems
\[ \pi E x \leq 0 \]  

(2.51)

is satisfied by all \( x \in SPP \). 

Consequently, any inequality (2.51) is a valid inequality for SPP. Note that the inequality (2.23) can be obtained by selecting a particular \( \pi \) 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 \( x \in \overline{SPP} \setminus SPP \).

**Separation Problem**: Given a fractional basic feasible solution \( \bar{x} = (\bar{x}_r, \bar{x}_f) \in \overline{SPP} \), 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.

\[ \text{SPP}_{\text{projection}}: \text{maximize} \{ (\pi^+ - \pi^-)E \bar{x} : (\pi^+ - \pi^-)D \leq 0, \]
\[ \sum_{j} (\pi^+_j + \pi^-_j) \leq 1, \]
\[ \pi^+ \geq 0, \pi^- \geq 0 \} \]

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

\[ \bar{\pi}Ex \leq 0. \]  

(2.52)

The Polyhedral Structure of Combinatorial Optimization Problems

77
Conversely, if there exists a valid inequality \( \pi^x \leq 0 \) that cuts off \( \bar{x} \), then the inequality (2.52) generated from an optimal solution to \( \text{SPP}_{\text{Projection}} \) will also cut off \( \bar{x} \). Since the size of problem \( \text{SPP}_1 \) 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 \( \text{SPP}_1 \), while maintaining the properties of the strong formulation. This issue will be addressed next.

We now consider another RLT formulation of \( \text{SPP} \). Let \( \bar{x} \) be an optimal basic feasible solution to \( \text{SPP} \). 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 \( x_i \) \( i \in F \). Denote \( w_{ii} = x_i \), \( w_{ij} = w_{jk} \) if \( j < k \) and \( w_{ii} \) otherwise, \( \forall j, k \in F \) and \( v_{jk} = x_i x_k \) \( \forall j e F, k \in F \). Then a first-order RLT formulation with this restricted set of products can be described as follows.

\[
\sum_{j \in N_i} x_j = 1 \quad \forall i \in M
\]

\[
\sum_{j \in N_i \cap F} w_{ij} + \sum_{j \in N_i \cap F} v_{jk} = x_k \quad \forall i \in \tilde{M}_k, \quad \forall k \in F
\]

\[
w_{ij} \geq 0 \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F
\]

\[
w_{ij} \leq x_j \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F
\]
\[ w_{jk} \leq x_k \quad \forall j \in \tilde{N}_k \cap F, \quad k \in F \] (2.57)

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

\[ v_{jk} \geq 0 \quad \forall j \in \tilde{N}_k \cap \bar{F}, \quad k \in F \] (2.59)

\[ v_{jk} \leq x_j \quad \forall j \in \tilde{N}_k \cap \bar{F}, \quad k \in F \] (2.60)

\[ v_{jk} \leq x_k \quad \forall j \in \tilde{N}_k \cap \bar{F}, \quad k \in F \] (2.61)

\[ v_{jk} \geq x_j + x_k - 1 \quad \forall j \in \tilde{N}_k \cap \bar{F}, \quad k \in F \] (2.62)

\[ x_j \geq 0, \forall j \in \mathbb{N} \] (2.63)

where \( \tilde{N}_k \equiv \bigcup_{i \in M_k} N_{ik} \).

**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 \( i \in M_k \) \( \exists j \in N_k \), by (2.54),

\[ w_{jk} + \sum_{i \in M_k \cap F} w_{ik} + \sum_{i \in N_k \cap \bar{F}} v_{sk} = x_k. \] (2.64)
Since \( w_{nk} \leq x_i \) and \( v_{ik} \leq x_i \), (2.64) implies that

\[
 w_{(kj)} + \sum_{t \in N_{i,k} \cap F - j} x_t + \sum_{s \in N_{i,k} \cap \bar{F}} x_s \geq x_k. 
\]  

(2.65)

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

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

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 \( \{ \sum_{j \in N} c_j x_j : (x,w) \text{ satisfies (2.53), (2.54), (2.55), (2.59), (2.60), (2.63)} \} \).

**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 \( \bar{M}_k \). 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
\( x_i \in \mathcal{S}_i \) 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 \( \bar{x} = (\bar{x}_{\bar{r}}, \bar{x}_{\bar{p}}) \) solves \( \overline{SP} \). If \( \bar{x}_{\bar{p}} \) is fixed as above, the resulting surviving (nontrivial) constraint sets are of the type

\[
\left( \sum_{i \in \mathcal{S}_t \cap \bar{N}_i} x_i = 1 \text{ for } i \in \mathcal{S}_t \cap \bar{N}_i \neq \emptyset \right), \text{ for } t = 1, \ldots, p.
\] (2.66)

Applying RLT on SPP by multiplying each \( x_i, j \in \mathcal{S}_i \) 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.

\[
\sum_{j \in \bar{N}_i} x_j = 1 \quad \forall \ i \in \mathcal{M}
\]

\[
\left( \sum_{j \in \bar{i}, k \cap F} w_{jk} = x_k \forall k \in \mathcal{S}_t \right), \text{ for } i \in \bar{i}, k \cap F \neq \emptyset , \forall t = 1, \ldots, p
\]

\( w_{jk} \geq 0 \text{ for } j \in \bar{N}_k \cap F, \ k \in \mathcal{S}_t, \ t = 1, \ldots, p \) (2.67)

\( v_{jk} \leq x_j \text{ for } j \in \bar{N}_k \cap F, \ k \in \mathcal{S}_t, \ t = 1, \ldots, p \)

\( v_{jk} \geq 0 \text{ for } j \in \bar{N}_k \cap F, \ k \in \mathcal{S}_t, \ t = 1, \ldots, p \)

\( x_j \geq 0, \ \forall j \in \mathcal{N} \)

where \( \bar{N}_k = \bigcup_{i \in \bar{i}, k \cap \mathcal{S}_t \neq \emptyset} \mathcal{N}_{i, k} \ \forall k \in \mathcal{S}_t, \ t = 1, \ldots, p. \)
Proposition 2.13 If $\bar{x} = (\bar{x}_r, \bar{x}_F)$ 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, $\bar{w}_{ij} = 0 \forall i \in F$ and $\bar{v}_i = 0 \forall i \in F$ if $\bar{x}_i = 0$, and $\bar{w}_{ij} = \bar{x}_i \forall i \in F$, and $\bar{v}_i = \bar{x}_i \forall i \in F$. if $\bar{x}_i = 1 \forall j \in F$, after fixing $x_i = \bar{x}_i \forall j \in F$, 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 $x_R = \{x_i : j \in S_t\}$. Since $\bar{x}$ is a basic feasible solution, this system is of the form $B_i\bar{x}_R = e$ where $e = (1, \ldots, 1)^t$, and $B_i$ is nonsingular. Letting $W_i = (w_{ij}, j \in S_t)$, we have by RLT that $B_i\bar{W}_i = e \bar{x}_i$ for all $i \in S_t$. Hence, $\bar{W}_i = B_i^{-1}e\bar{x}_i$ for all $i \in S_t$.

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

Corollary 2.14 If $\bar{x}$ is a unique optimal fractional basic feasible solution to $\overline{SPP}$, then $v(SPP_{RLT}) > v(\overline{SPP})$, 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 $SPP$. 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 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 \( v(SP_\delta) = \min \{ cx : x \in \text{SPP}_\delta \} \), where \( \text{SPP}_\delta \) is a projection of the \( \delta \)-th-order relaxation \( \text{SPP}_\delta \) onto the original \( x \) space, the lower bound \( v(SP_\delta) \) approaches the optimal objective value of the problem SP when \( \delta \) approaches \( n \). Moreover, for a fixed \( \delta \), \( v(SP_\delta) \) can (in principle) be computed in polynomial time, as the optimum of some linear programming problem. However, the computational effort of evaluating \( v(SP_\delta) \) grows exponentially with \( \delta \). 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

\[
\text{QP: Maximize } \left\{ \sum_{j \in \mathbb{N}} q_j x_j + \sum_{1 \leq i < j \leq n} q_{ij} x_i x_j : x_i \in \{0, 1\}, \forall j \in \mathbb{N} \right\},
\]

where all the data is rational and \(\mathbb{N} = \{1, \ldots, n\}\).

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 (1984a, 1984b) 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.

\begin{align}
  w_{ij} - x_i - x_j &\leq 0 \quad \forall 1 \leq i < j \leq n \\
  w_{ij} - x_i &\leq 0 \quad \forall 1 \leq i < j \leq n \\
  -w_{ij} + x_i + x_j &\leq 1 \quad \forall 1 \leq i < j \leq n \\
  x_i &\in (0, 1), \quad \forall i \in \mathbb{N}, \quad w_{ij} \geq 0, \quad \forall 1 \leq i < j \leq n.
\end{align}

(3.1) (3.2) (3.3) (3.4)

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

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

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

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

as the \textit{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*. 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*, 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).

$$NQP(n) = \{x \in \mathbb{R}^n : F_n(J, \bar{J}) \geq 0 \ \forall J \subseteq N, \ \bar{J} = N - J\}$$  \hspace{1cm} (3.5)
where $F_n(J, \overline{J}) = \prod_{t \in J} \prod_{s \in \overline{J}} (1 - x_s)$.

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

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

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

$$x_i x_j \geq \sum_{r = 3}^{n} \sum_{|J| = r \cup j \in J}^{n} \prod_{t \in J} x_t \prod_{s \in \overline{J}} (1 - x_s) \quad \text{for } 1 \leq i < j \leq n$$

$$\prod_{t \in J} x_t \prod_{s \in \overline{J}} (1 - x_s) \geq 0 \quad \text{for } J \subseteq N, |J| = 3, \ldots, n.$$
Now, let us use the transformation \( y_j = f_\ast(J, \bar{J}) \), \( \forall J \subseteq \mathbb{N}, |J| = 3, \ldots, n \), whose inverse is given by \( w_j = \sum_{\bar{J} \subseteq J} y_{\bar{J}} \), \( \forall J \subseteq \mathbb{N}, |J| = 3, \ldots, n \). Writing the linearized set of constraints for NQP(n), and substituting out \( w_j \), for \( 3 \leq |J| \leq n \), in terms of \( y_j \), we obtain an equivalent linearized reformulation of (3.5), given by \( \text{QP}^n = \{(x, w) \in \mathbb{R}^n : (x, w, y) \in Z^*\} \), where the constraints of \( Z^* \) are as follows.

\[
1 - \sum_{j \in \mathbb{N}} x_j + \sum_{1 \leq i < j \leq n} w_{ij} \geq \sum_{r = 3}^{n} \sum_{|J| = r}^{n} \frac{(r-1)(r-2)}{2} y_j \tag{3.6}
\]

\[
x_j - \sum_{1 \leq i < j \leq n} w_{ij} \geq \sum_{r = 3}^{n} \sum_{|J| = r \cap J \neq \emptyset}^{n} (2 - r)y_j \quad \text{for } j = 1, \ldots, n \tag{3.7}
\]

\[
w_{ij} \geq \sum_{r = 3}^{n} \sum_{|J| = r \cap J \neq \emptyset}^{n} y_j \quad \text{for } 1 \leq i < j \leq n \tag{3.8}
\]

\[
y_j \geq 0 \quad \text{for } J \subseteq \mathbb{N}, |J| = 3, \ldots, n.
\]

Let \( \pi_0, \pi_i, \text{ and } \pi_{ij} \) represent dual multipliers associated with constraint (3.6), (3.7), and (3.8), respectively. Since \( \text{QP}^n \) is a full dimensional polytope, using the projection operation (see, for example, Nemhauser and Wolsey, 1988 or Balas, 1992), we have

\[
\text{QP}^n = \{(x, w) \in \mathbb{R}^n : \sum_{j \in \mathbb{N}} (\pi_0^{k} - \pi_j^{k})x_j + \sum_{1 \leq i < j \leq n} (\pi_i^{k} + \pi_j^{k} - \pi_{ij}^{k} - \pi_0^{k})w_{ij} \leq \pi_0^{k}, \forall k \in K\} \tag{3.9}
\]

where \( \pi^k, k \in K \), are extreme directions of the following polyhedral cone \( \Pi \):

The Polyhedral Structure of Combinatorial Optimization Problems
\[ \Pi \equiv \{ \pi \geq 0 : \frac{-(r-1)(r-2)}{2} \pi_0 + (r-2) \sum_{j \in J} \pi_j - \sum_{i < j \in J} \pi_{ij} \leq 0 \ \forall J \subseteq \mathbb{N} \ \exists |J| = r = 3, \ldots, n \}, \]

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

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.

\[ \text{QP}^n = \{ (x, w) \in \mathbb{R}^n : \sum_{j \in \mathbb{N}} \lambda_j^k x_j + \sum_{1 \leq i < j \leq n} \lambda_{ij}^k w_{ij} \leq \lambda_0^k, \ \forall k \in K \} \quad (3.10) \]

where \( \lambda^k, k \in K, \) are extreme directions of the polyhedral cone \( \Lambda, \) given by

\[ \Lambda \equiv \{ \lambda : \sum_{j \in \mathbb{J}} \lambda_j + \sum_{i < j \in \mathbb{J}} \lambda_{ij} \leq \lambda_0, \ \forall J \subseteq \mathbb{N} \}. \quad (3.11) \]

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

**Proposition 3.1** The inequality \( \sum_{j \in \mathbb{N}} \lambda_j x_j + \sum_{1 \leq i < j \leq n} \lambda_{ij} w_{ij} \leq \lambda_0 \) is valid for \( \text{QP}^n \) if and only if \( \lambda \equiv (\lambda_j, \lambda_{ij}, \lambda_0) \) belongs to \( \Lambda, \) where \( \Lambda \) is a polyhedral cone given by (3.11). Furthermore, this valid inequality is facet defining for \( \text{QP}^n \) if and only if \( \lambda \equiv (\lambda_0, \lambda_{ij}, \lambda_0) \) is an extreme direction of the polyhedral cone \( \Lambda. \) ■
Remark 3.1 Given \( \lambda \equiv (\lambda_i, \lambda_{ij}, \lambda_j) \in \Lambda \) that defines a facet of \( \text{QP}^n \), let the support graph \( G(\lambda) \) have nodes \( N(\lambda) \) corresponding to indices of nonzero coefficients \( \lambda_i \), and have edges \( A(\lambda) \) corresponding to indices of nonzero \( \lambda_{ij} \) 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 \( \text{QP}^{n'} \) \( (n' < n) \) is a facet for \( \text{QP}^n \), and conversely, if \( |N(\lambda)| = n' < n \), then the given facet of \( \text{QP}^n \) is a facet for \( \text{QP}^{n'} \). 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 \( \text{QP}^n \) 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 \( \text{QP}^n \) 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 \( a_i \) 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 \( \text{QP}^n \). 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 \( \Lambda \), 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 \( \text{QP}^n \).

Motivated by Remark 3.1, let \( J_1 \subseteq \mathbb{N} \) and \( J_2 \equiv \mathbb{N} - J_1 \), and let \( \alpha \in \{1, \ldots, n - 2\} \), and consider the following quadratic function, where \( \bar{x}_i = (1 - x_i) \forall i \in \mathbb{N} \).

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

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 \textit{(nonhomogeneous) product-form inequality},

\[
(a - t) \sum_{j \in J_1} x_j + (t - \alpha - 1) \sum_{j \in J_2} \bar{x}_j - \sum_{i < j \in J_1} w_{ij} - \sum_{i < j \in J_2} w_{ij} + \sum_{(i,j) \in (J_1 \times J_2)} w_{ij} \leq \frac{(t - \alpha)(t - \alpha - 1)}{2} \quad (3.14)
\]

where for any \( J_1 \subseteq \mathbb{N} \) and \( J_2 \equiv \mathbb{N} - J_1 \), let \( t = |J_1| \) 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 \( \text{QP}^n \).
**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 \(\lambda\) is an extreme direction of \(\Lambda_0\) given below, having all the indices in \(N\) accounted among the nonzero coefficients in (3.15).

\[
\Lambda_0 = \{ \lambda : \sum_{j \in J} \lambda_j + \sum_{1 < j \in J} \lambda_{ij} \leq 0 \ \forall J \subseteq N \}. \tag{3.16}
\]

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.

\[
- \sum_{j \in J} x_j - \sum_{1 < j \in J} w_{ij} - \sum_{1 < j \in \bar{J}} w_{ij} + \sum_{(i, j) \in (J : \bar{J})} w_{ij} \leq 0 \ \forall J \subseteq N, |J| = 1, \ldots, n - 2, \bar{J} \equiv N - J \tag{3.17}
\]
Proposition 3.3. For any \( J_1 \subseteq N \) and \( J_2 \equiv N - J_1 \), with integer \( \alpha, 1 \leq \alpha \leq n - 2 \), and \( t = |J_2| \), the product-form inequality of type (3.14) is a facet-defining inequality for \( \text{QP}^n \), denoted by a product-form facet. The number of distinct product-form facets equals \( (n - 3)(\alpha - 1) + 2n^2 - n + 1 \).

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

\[
f_n = \binom{n}{2} \tilde{f}_2 + \binom{n}{3} \tilde{f}_3 + \ldots + \binom{n}{n} \tilde{f}_n.
\]

(3.18)

In order to avoid duplications in counting \( \tilde{f}_k \) for any \( k \geq 3 \), note that by interchanging \( J_1 \) and \( J_2 \), and replacing \( \alpha \) by \( \alpha' \) 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 \( \alpha \) 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 \( \alpha \), we get

\[
\tilde{f}_k = (\frac{k}{2} - 1)2^k.
\]

(3.19)
From (3.18) and (3.19) observing that \( \tilde{t}_1 = 4 \), we obtain
\[
\tilde{f}_s = \sum_{k=3}^{n} \binom{n}{k} 2^k (k/2 - 1) + \binom{n}{2} 4
= \sum_{k=0}^{n} \binom{n}{k} k 2^{k-1} - \sum_{k=6}^{n} \binom{n}{k} 2^k + (2n^2 - n + 1) = (n - 3)3^{n-1} + 2n^2 - n + 1,
\]
and this completes the proof. ■

We now recall three classes of facets from Padberg (1989). For any \( S \subseteq \mathbb{N} \) with \( |S| \geq 2 \) and integer \( \beta, 1 \leq \beta \leq |S| - 1 \), 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 \( S \subseteq \mathbb{N} \) with \( |S| \geq 1 \) and \( T \subseteq \mathbb{N} - S \) 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 \( S \subseteq \mathbb{N} \) with \( s = |S| \geq 1 \) and \( T \subseteq \mathbb{N} - S \) with \( t = |T| \geq 2 \), the generalized cut facet is given by
\[
(s - 1) \sum_{j \in S} x_j + (t - s - 1) \sum_{j \in T} x_j - \sum_{i < j \in S} w_{ij} - \sum_{i < j \in T} w_{ij} + \sum_{(i,j) \in (S : T)} w_{ij} \leq \frac{(t - s)(t - s - 1)}{2}.
\]

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

**Proposition 3.4** The clique facet for \( \text{QP}^n \) 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 \( \beta \in \{1, \ldots, |S| - 2\} \), \( S \subseteq \mathbb{N}, |S| \geq 3 \).

The cut facet for \( \text{QP}^n \) 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 \( S \subseteq \mathbb{N}, |S| \geq 1, T \subseteq \mathbb{N} - S, |T| \geq 2 \).
The generalized cut facet for $QP^n$ belongs to the class of the product-form facets with $J_1 = S$, $J_2 = T$, $a = |S|$, 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 $x_j$, $j \in N' = \{1, ..., n - 1\}$, of the form

$$
\sum_{j \in N'} \lambda_j x_j + \sum_{i < j \in N'} \lambda_{ij} w_{ij} \leq 0.
$$

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

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

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 $N' \equiv \{1, ..., n - 1\}$, the inequality

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

is a facet for $QP^n$ if and only if $\bar{\lambda}(n) \equiv (\bar{\lambda}_n, \bar{\lambda}_j, j \in N')$ is a vertex of the following polyhedron:
\[ \{ \lambda : \lambda_n + \sum_{j \in J} \lambda_{jn} \leq -\sum_{j \in J} \bar{\lambda}_j - \sum_{i < j \in J} \bar{\lambda}_{ij} \quad \forall J \subseteq N' \}. \] (3.22)

**Proof.** Let us partition the constraints of \( \Lambda_0 \) as follows.

\[ \text{N' - Block: } \sum_{j \in J} \lambda_j + \sum_{i < j \in J} \lambda_{ij} \leq 0 \quad \forall J \subseteq N' \] (3.23)

\[ \text{n - Block: } \lambda_n + \sum_{j \in J} \lambda_{jn} \leq -\sum_{j \in J} \lambda_j - \sum_{i < j \in J} \lambda_{ij} \quad \forall J \subseteq N' \] (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 \( \bar{\lambda}(n) \equiv (\bar{\lambda}_n, \bar{\lambda}_j, \bar{\lambda}_{ij} \text{ for } j \in N') \) 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 \( \bar{\lambda}(N) \equiv (\bar{\lambda}_j \text{ for } j \in N, \bar{\lambda}_{ij} \text{ for } i < j \in N) \) is feasible to \( \Lambda_0 \) 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 \( \Lambda_0 \), and so from Proposition 3.1, is a facet of QP*.

**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 \( \lambda(n) \equiv \)
\( (\lambda_n, \lambda_{jn} \text{ for } j \in \mathbb{N}') \) are being simultaneously determined, as opposed to being sequentially determined.

Consider the homogenous product-form facet of the form (3.17) in \( \mathbb{N}' \)-space defined for some \( J_1 \subseteq \mathbb{N}', \{J_1\} \in \{1, \ldots, n - 3\}, \overline{J_1} = \mathbb{N}' - J_1. \)

\[
- \sum_{j \in J_1} x_j - \sum_{i < j \in J_1} w_{ij} - \sum_{i \in J_1} w_{ij} + \sum_{(i,j) \in (J_1 : \overline{J_1})} w_{ij} \leq 0
\]  

(3.25)

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

\[
\lambda_n + \sum_{j \in J} \lambda_{jn} \leq \frac{(p - q)(p - q + 1)}{2} \quad \forall J \subseteq \mathbb{N}' \text{, where } |J \cap J_1| = p, |J \cap \overline{J_1}| = q.
\]  

(3.26)

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

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

(ii) \( \lambda_n = -1, \lambda_{jn} = -1 \ \text{for } j \in J_1, \lambda_{jn} = 1 \ \text{for } j \in \overline{J_1} \) and

(iii) \( \lambda_n = 0, \lambda_{jn} = 1 \ \text{for } j \in J_1, \lambda_{jn} = -1 \ \text{for } j \in \overline{J_1} \).
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 $J = J_i \cup \{n\}$ and $\overline{J} = \overline{J}_i$, in (3.17), and (iii) corresponds to $J = J_i$ and $\overline{J} = \overline{J}_i \cup \{n\}$ in (3.17).

**Proof.** First of all, note that the given values of $(\lambda_{in}, \lambda_{in}$ for $j \in N')$ 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 $(\lambda_{in}, \lambda_{in}$ for $j \in N')$ 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 $n = 5$, $J_i = \{1,2\}$, $\overline{J}_i = \{3,4\}$, $\lambda_4 = -1$, $\lambda_{12} = 2 \forall j \in J_i$, $\lambda_{12} = -2 \forall j \in \overline{J}_i$. Let us verify that this $\lambda(n)$ 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., $(p-q)^2 - 3(p-q) + 2 \geq 0$ for $p = 0,1,2$, $q = 0,1,2$. 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 $\lambda(n)$, and so yield $\lambda(n)$ 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, \( \overline{\lambda}(n) \) is a vertex of (3.26). The corresponding facet is

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

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

\[
\lambda_{jn} = \theta \quad \forall j \in J_1 \quad \text{and} \quad \lambda_{jn} = -\theta \quad \forall j \in \overline{J}_1.
\]  \hspace{1cm} (3.27)

Then, system (3.26) becomes

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

where \( P = |J_1| \). This is equivalent to the system

\[
\lambda_n + \theta r \leq \frac{r(r + 1)}{2} \quad \forall r = P + 1 - n, \ldots, P.
\]  \hspace{1cm} (3.26)

**Proposition 3.7** The vertices \( (\lambda_n, \theta) \) of the two dimensional polytope (3.26) are all of the type

\[
\lambda_n = \frac{-R(R - 1)}{2} \quad \text{and} \quad \theta = R, \quad \text{for} \ R = P + 2 - n, \ldots, P.
\]  \hspace{1cm} (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

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

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 \( P + 1 - n \) to \( P \), we get vertices of (3.28) as
\( \theta = R \) for \( R = P + 2 - n, \ldots, P \), and \( \lambda_n = R(R + 1)/2 - R^2 = -R(R - 1)/2 \). This completes the proof. \( \blacksquare \)

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 \( \vec{\lambda}(n) \equiv \{ \vec{\lambda}_n = -R(R - 1)/2, \vec{\lambda}_{jn} = R \forall j \in J_1, \vec{\lambda}_{jn} = -R \forall j \in \overline{J}_1 \} \), for each \( R = P + 2 - n, \ldots, P \), 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

\[
\lambda_n = -\frac{R(R - 1)}{2}, \quad \lambda_{jn} = R \forall j \in J_1, \quad \lambda_{jn} = -R \forall j \in \overline{J}_1,
\]

\[
(3.30)
\]
is feasible to (3.26) for all $R = P + 2 - n, \ldots, P$, where $|J_1| = P$. 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 $(p-q)(p-q+1)/2$. Hence, the constraint is binding whenever these are equal, i.e., whenever $(p-q-R)(p-q-R+1) = 0$, i.e., $(p-q) = R$ or $(p-q) = R-1$.

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 $|\overline{J}_1| \geq 2$ when $n \geq 4$.) We show below that the system of binding constraints implies that $\lambda_s = -\lambda_t$. Since the choice of $s$ and $t$ is arbitrary, this in turn means that $\lambda_s = 0$, $\forall j \in J_1,$ and $\lambda_t = -\lambda \forall j \in \overline{J}_1,$ is implied by this system. But this reduces the system (3.26) to the system (3.28). Examining the constraints in (3.28) for $r = (p-q) = R$ and $r = (p-q) = R-1$, which are binding at the given solution (3.30) as noted above, we get $\lambda_s + \theta R = R(R + 1)/2$ and $\lambda_t + \theta(R - 1) = R(R - 1)/2$. These equation uniquely yield $\lambda_s = -R(R - 1)/2$ and $\theta = R$, and so this implies that (3.30) is a vertex of (3.26).

Hence, to show that $\lambda_s = -\lambda_t$, consider a set $J \subseteq N'$ such that $|J \cap J_1| = p$, $|J \cap \overline{J}_1| = q$, $\{s, t\} \cap J = \phi$, and $(p-q) = (R-1)$ if $R = P$, and $(p-q) = R$ if $R < P$. (Note that if $R = P$, we must have $J = J_1 - \{s\}$, and if $R < P$, then since $R \geq P + 2 - n$ such a $J$ exists for which $p \leq P - 1 = |J_1| - 1$, $q \leq n - 2 - P = |\overline{J}_1| - 1$, and $(p-q) = R$.) 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

---

The Polyhedral Structure of Combinatorial Optimization Problems 101
\[ \lambda_n + \sum_{j \in J} \lambda_{jn} = \lambda_n + \sum_{j \in J} \lambda_{jn} + \lambda_{sn} + \lambda_{ln}. \]

This implies that \( \lambda_{sn} = -\lambda_{ln}, \) and hence, the proof is complete. \( \blacksquare \)

Consequently, we have the following result.

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

\[
- \sum_{j \in J_1} x_j - \sum_{i < j \in J_1} w_{ij} - \sum_{i < j \in J_1} w_{ji} - \sum_{(i,j) \in (J_1 \cup J_2)} w_{ij} - \frac{R(R - 1)}{2} x_n + R \sum_{j \in J_1} w_{jn} - R \sum_{j \in J_1} w_{jn} \leq 0
\]

where \( J_1 \subseteq N, |J_1| = P \geq 1, \overline{J}_1 = N - J_1, \) with \( |\overline{J}_1| \geq 2, \) and \( R \in \{ P + 2 - n, ..., P \}. \)

**Proof.** Follows from Proposition 3.7 and 3.8. \( \blacksquare \)

### 3.5. Summary and Recommendations for Further Research

We have developed a class of facets obtained by lifting homogenous product-form facets of QP*. 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).
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 \( i \in \mathbb{N} = \{1,\ldots,n\} \), and that there are \( m \) illuminators, indexed \( j \in \mathbb{M} = \{1,\ldots,m\} \). Each target \( i \) has a specified engageability duration \( [r_{ij}, d_{ij}] \), 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 \( \Delta_{ij} \), prior to its scheduled engagement time \( t_{ij} \), 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 \( j \in M \), each job \( i \in N \) has a release time \( r_{ij} \), a completion time (due-date or deadline) \( d_{ij} \), and a processing time \( \Delta_{ij} \). 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 \( \varepsilon \)-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. \textit{\Delta-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 $\Delta$, say. Moreover, suppose that for each $i \in \mathbb{N}$, $r_{i,j} = r_i$ and $d_{i,j} = d_i$, for all $j \in \mathcal{M}$, where $r_i$ and $d_i$ are all multiples of $\Delta$, and that the start and end times of the unavailability intervals on each machine are all also multiples of $\Delta$. We denote this problem a $\Delta$-multiple duration scheduling problem. Note that this situation may have been achieved by conservatively perturbing the data, given roughly equal and small processing times $\Delta_i$, for $i \in \mathbb{N}$. In other words, we assume that the problem data has been perturbed so that each processing time is set at $\Delta = \max_{i \in \mathbb{N}} \{\Delta_i\}$, the values $r_i$, $i \in \mathbb{N}$, have been increased and the values $d_i$, $i \in \mathbb{N}$, 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 $r_i, d_i$, $i \in \mathbb{N}$, and the start and end times of machine unavailability durations are all nonnegative integers.

Under this assumption, let $T = \max_{i \in \mathbb{N}} \{d_i\}$ 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 $k \in \mathcal{K} = \{1,\ldots,K\}$ 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 $\tau_k$ denote the upper interval end-point of slot $k$, for $k \in \mathcal{K}$. 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 $\tau_k$. The corresponding cost with respect to the objective function of making an assignment $x_{ik}$ is given by

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

(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 $t = 0, \ldots, T-1$, let us define $N(t) = \{i \in N : r_i \leq t\}$, $n(t) = |N(t)|$, and let $\{k = 1, \ldots, a(t)\}$ 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 $\{1, \ldots, n(\hat{t})\}$. 
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 \( 1 \leq \hat{t} \leq \hat{t} \), suppose that all the slots \( 1,...,a(\hat{t}-1) \) are occupied by jobs in \( N(\hat{t}-1) \) for the given optimal solution. Flag the jobs assigned to these slots. If \( \hat{t} = \hat{t} \), then the unflagged jobs in \( N(\hat{t}) \) must occupy the slots \( \{a(\hat{t}-1) + 1,...,a(\hat{t})\} \), since \( n(\hat{t}) \leq a(\hat{t}) \), and so these jobs can be made to occupy the slots \( \{a(\hat{t}-1) + 1, ..., n(\hat{t})\} \) while preserving the optimal objective value. On the other hand, if \( \hat{t} > \hat{t} \), then as before, some \( a(\hat{t}) - a(\hat{t}-1) \) jobs from the unflagged ones in \( N(\hat{t}) \) must occupy the slots \( a(\hat{t}-1) + 1, ..., a(\hat{t}) \) 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. \( \blacksquare \)

Proposition 5.1 prompts the following algorithm for the \( \Delta \)-multiple duration scheduling problem.

Algorithm ASP

STEP 1. Find \( \hat{t} = \min\{t: n(t) \leq a(t)\} \), 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 \( \infty \) otherwise.

\[
\text{AP: } \quad \text{Minimize } \sum_{i \in N(\hat{t})} \sum_{k = 1}^{n(\hat{t})} c_{ik} x_{ik}
\]
subject to

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

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

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

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 \(N = \emptyset\), 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 \(\Delta\)-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 \(1, \ldots, n\) 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 \(B_i\) blocked-off durations were encountered in this process, this part is of complexity \(O(B_i + n(\hat{t}))\), 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 $n_p \times n_p$ can be solved in time $O(n_p^4)$ (see, for example, Bazaraa et al., 1990). Since $\sum_{p=1}^q n_p^4 \leq \left( \sum_{p=1}^q n_p \right)^4 = n^4$, 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. \qed

**Remark 5.1** At Step 1 of Algorithm ASP, we have taken $c_k = \infty$ for $k \notin S_k$, for each $i \in \mathbb{N}$. 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, usable 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 $i \in \mathbb{N}$, $j \in \mathbb{M}$. 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 endpoints of each unavailability interval duration on each of the machines $j \in \mathbb{M}$, are all multiples of some quantity $\Delta$. (Possibly, $\Delta = 1$.) Scaling by $\Delta$, 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 \( \tau_k \) denotes the upper interval end-point of slot \( k \). for \( k \in K \), let us define for each job \( i \in N \), a set

\[
S_i = \{ k \in K : [\tau_k - \Delta_i, \mu_k, \tau_k] \subset \tau_i, [\mu_i, \mu_k] \text{ and } \\
(\tau_k - \Delta_i, \mu_k, \tau_k) \cap \{ \text{unavailability intervals for the machine corresponding to slot } k \} = \phi \}
\]

where \( \mu_k \) 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 \( \tau_k \). 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 \( \mu_k \). 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 \( k \in K \), the set \( J_k = \{(i, \rho) : i \in \mathbb{N}, \rho \in S_i, \text{ and } [\tau_k - 1, \tau_k] \subseteq [\tau_{\rho} - \Delta_{i, \rho}, \tau_{\rho}]\} \). Note that for each \( k \in K \), \( J_k \) is the set of combinations \((i, \rho)\) such that slot \( k \) will be occupied if \( x_\rho = 1 \). Then, problem GP can be formulated as follows.

\[
\text{GP:} \quad \text{Minimize} \quad \sum_{i \in \mathbb{N}} \sum_{k \in S_i} c_{ik} x_{ik}
\]

subject to

\[
\sum_{k \in S_i} x_{ik} = 1 \quad \text{for } i \in \mathbb{N} \quad (6.1)
\]

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

\[
x_{ik} \in \{0, 1\} \quad \text{for } k \in S_i, \ i \in \mathbb{N} \quad (6.3)
\]

### 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

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

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.

\[
\text{GPGN:} \quad \text{Minimize} \quad \sum_{i \in \mathbb{N}} \sum_{k \in S_i} c_{ik} x_{ik}
\]

subject to
\[
\sum_{i \in S_i} x_{ik} = 1 \quad \text{for } i \in N
\]

\[
\sum_{k \in J(i, \rho)} y_{(i, \rho)k} = \Delta_{i, \mu_k} x_{i, \rho} \quad \text{for } \rho \in S_i, \ i \in N
\]

\[
\sum_{(i, \rho) \in J_k} y_{(i, \rho)k} \leq 1 \quad \text{for } k \in K
\]

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

\[0 \leq y_{(i, \rho)k} \leq 1, \ \text{for } (i, \rho) \in J_k, \ k \in K\]

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

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

**Proposition 6.1** \(v(\bar{G}P) \geq v(\bar{GP}GN)\). ■

Hence, although the underlying linear programming relaxation of \(GP\) can be solved using special generalized network flow techniques (see Bazzaraa 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,

\[z_{ik} = 1 \text{ if slot } k \text{ is occupied by job } i \text{ and } 0 \text{ otherwise},\]
\( R_k = \{ i : (i, \rho) \in J_k \text{ for some } \rho \geq k \} \equiv \{ i : z_{ik} \text{ is defined} \}, \text{ for each } k \in K, \)

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

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

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

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

\[
\sum_{i \in \mathbb{N}} \xi_i
\]

subject to

\[
\xi_i \geq w_{ik} \tau_k z_{ik} \text{ for } k \in F_i, \text{ } i \in \mathbb{N} \tag{6.5}
\]

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

\[
\sum_{i \in M} \frac{1}{\Delta_{i,j}} \sum_{k \in F_i, \rho \equiv j} z_{ik} = 1 \text{ for } i \in \mathbb{N} \tag{6.7}
\]

\[
\sum_{rho \in P_{ik}} \tau_{\rho} \leq \max_{i \in \mathbb{N}}(\Delta_{i,j})(1 - z_{ik}) \text{ for } k \in F_i, \text{ } i \in \mathbb{N} \tag{6.8}
\]
\[ z_{ik} \in \{0, 1\}, \quad \text{for } k \in K, \ i \in N \]
\[ \xi_i \geq 0, \quad \text{for } i \in N \]

**Proposition 6.2** Suppose that \( \Delta_{ij} = \Delta_i \), \( \forall j \in M \), for each \( i \in N \), and that \( w_{ik} = w_{ik'} \), \( \forall k \in S_i, \ i \in N \). Then, for every feasible solution to \( \overline{GP} \) having an objective function value \( v_1 \), there corresponds a feasible solution to \( \overline{GP_AF} \) 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 \( \overline{GP} \). Define \( Z_{ik} = \{ \rho : (i, k) \in \mathcal{A} \} \) 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 \( \sum_{i \in K_k \rho \in Z_{ik}} x_{i\rho} \leq 1 \), for \( k \in K \), we have, \( \sum_{i \in K_k} \bar{z}_{ik} \leq 1 \), 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 \( j \in M \),

\[
\sum_{k \in F_i \mu_k \equiv j} \bar{z}_{ik} = \sum_{k \in F_i \mu_k \equiv j} \sum_{\rho \in Z_{ik}} \bar{x}_{i\rho} = \sum_{\rho \in \mathcal{S}_i} \bar{x}_{i\rho}.
\]

Hence, from (5.1), we get,

\[
\sum_{j \in M} \frac{1}{\Delta_{ij}} \sum_{k \in F_i \mu_k \equiv j} \bar{z}_{ik} = \sum_{j \in M} \sum_{\rho \in \mathcal{S}_i} \bar{x}_{i\rho} = \sum_{\rho \in \mathcal{S}_i} \bar{x}_{i\rho} = 1 \quad \forall i \in N,
\]

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

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

Hence, we obtain,
\[
\sum_{q \in F_i} \bar{z}_{q} = \sum_{q \in F_i} \bar{z}_{q} - \sum_{q \in A_{ik}} \bar{z}_{q} = \sum_{q \in F_i} \bar{z}_{q} - \sum_{q \in A_{ik}} \sum_{p \in Z_{iq}} \bar{z}_{ip}, \quad (6.9)
\]

Using (6.7), we get,

\[
\sum_{q \in F_i} \bar{z}_{q} = \sum_{j \in M} \sum_{k \in F_i, \mu_k \equiv j} \bar{z}_{ik} \leq \left[ \max_{j \in M} (\Delta_{ij}) \right] \sum_{j \in M} \frac{1}{\Delta_{ij}} \sum_{k \in F_i, \mu_k \equiv j} \bar{z}_{ik} = \max_{j \in M} (\Delta_{ij}) \bar{z}_{ik}. \quad (6.10)
\]

Furthermore, since \( \Delta_{ij} = \Delta_i \), \( \forall j \in M \)

\[
\sum_{q \in A_{ik}} \sum_{p \in Z_{iq}} \bar{x}_{ip} \geq \sum_{q \in A_{ik}} \sum_{p \in Z_{iq} \cap Z_{ik}} \bar{x}_{ip} = \sum_{q \in Z_{ik}} \sum_{p \in A_{ik}} \bar{x}_{ip} = \Delta_{i} \sum_{p \in Z_{ik}} \bar{x}_{ip} = \max_{j \in M} (\Delta_{ij}) \bar{z}_{ik}. \quad (6.11)
\]

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

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

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 \( \overline{GP} \) and \( \overline{GPAF} \), respectively, where \( \xi_i = \max \{ w_i \tau_i \bar{z}_{ik} : k \in F_i \} \geq 0 \) for \( i \in N \). Accordingly, we have that \( v_1 = \sum_{i \in N} \sum_{p \in Z_{ik}} w_i \tau_i \bar{x}_{ip} \), and that

\[
v_2 = \sum_{i \in N} \xi_i = \sum_{i \in N} \max \{ w_i \tau_k \sum_{p \in Z_{ik}} \bar{x}_{ip} : k \in F_i \} \leq \sum_{i \in N} \max \{ w_i \sum_{p \in Z_{ik}} \tau_p \bar{x}_{ip} : k \in F_i \}.
\]
Since for each $i \in \mathbb{N}$, the foregoing maximum is realized for some slot $k \in F_i$, and since the weights $w_i$ are the same for all slots $l$ on that machine $\mu_i$, we have that $v_2 \leq \sum_{s \in \mathbb{N} \setminus S_i} w_i \tau_{l} \tilde{X}_{l} = v_1$. 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.

$$(r_{11}, d_{11}, \Delta_{11}, w_{11}) = (0, 5, 1, 1)$$
$$(r_{21}, d_{21}, \Delta_{21}, w_{21}) = (0, 3, 1, 3)$$
$$(r_{31}, d_{31}, \Delta_{31}, w_{31}) = (1, 4, 2, 3)$$
$$(r_{41}, d_{41}, \Delta_{41}, w_{41}) = (3, 6, 1, 3)$$

A summary of the results is given below.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of variables</th>
<th>Number of rows</th>
<th>$v(.)$</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPA</td>
<td>13</td>
<td>10</td>
<td>29.0</td>
<td>Integer solution</td>
</tr>
<tr>
<td>GPAF</td>
<td>18</td>
<td>38</td>
<td>15.07</td>
<td>Non integer</td>
</tr>
</tbody>
</table>

**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.

$$(r_{11}, d_{11}, \Delta_{11}, w_{11}) = (0, 5, 1, 1), (r_{12}, d_{12}, \Delta_{12}, w_{12}) = (1, 4, 2, 3)$$
$$(r_{21}, d_{21}, \Delta_{21}, w_{21}) = (0, 3, 1, 3), (r_{22}, d_{22}, \Delta_{22}, w_{22}) = (0, 3, 2, 3)$$

*Models and Algorithms for Scheduling Passive Homing Devices in a Battle-Group Problem* 118
A summary of the results is given below.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of variables</th>
<th>Number of rows</th>
<th>$v(.)$</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>12</td>
<td>11</td>
<td>5.0</td>
<td>integer solution</td>
</tr>
<tr>
<td>GPAF</td>
<td>16</td>
<td>39</td>
<td>1.4967</td>
<td>Non integer</td>
</tr>
</tbody>
</table>

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.
\[ \text{Minimize } \sum_{q \in Q} \sum_{i \in N} \sum_{k \in (T_i \cap S_q)} c_{ik} x_{ik} \]

subject to
\[ \sum_{k \in S_i} x_{ik} = 1 \text{ for } i \in N \] (6.12)
\[ \left( \sum_{(i, \rho) \in J_k} x_{ik} \right) \leq 1 \text{ for } k \in Q \] (6.13)
\[ x_{ik} \in \{0, 1\} \text{ for } k \in S_i, \ i \in N \] (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 \( \Delta_{i,k} \) 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 \( V = \{1, \ldots, K+1\} \), and an arc set \( E = E_s \cup E_r \), where \( E_s = \{(i, i+1): \text{for each slack variable } s_i, i = 1, \ldots, K\} \) and \( E_r = \{(p, q+1): \text{for each column of the interval matrix whose first } 1 \text{ is in row } p \text{ and whose last } 1 \text{ is in row } q, \text{ for each variable } x_{ik}, k \in S, i \in N\} \). 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 \( E_1, E_2, \ldots, E_n \), where \( \bigcup_{i \in N} E_i = E_r \), and where each \( E_i, i \in N \), contains arcs corresponding to the variables \( x_{ik} \) for \( k \in S \). 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 GP\_angular. Consider the following problem that schedules three jobs on two machines with time-window constraints, as given below.
\[(r_{1,1}, \ d_{1,1}, \ \Delta_{1,1}, \ w_{1,1}) = (0, 4, 2, 1), (r_{1,2}, \ d_{1,2}, \ \Delta_{1,2}, \ w_{1,2}) = (0, 4, 2, 1)\]
\[(r_{2,1}, \ d_{2,1}, \ \Delta_{2,1}, \ w_{2,1}) = (1, 4, 2, 2), (r_{2,2}, \ d_{2,2}, \ \Delta_{2,2}, \ w_{2,2}) = (1, 4, 2, 2)\]
\[(r_{3,1}, \ d_{3,1}, \ \Delta_{3,1}, \ w_{3,1}) = (2, 3, 1, 16), (r_{3,2}, \ d_{3,2}, \ \Delta_{3,2}, \ w_{3,2}) = (2, 3, 1, 16)\]

The blocked-off duration of machine 1 is \([1,2]\),
the blocked-off duration of machine 2 is \(\phi\).

<table>
<thead>
<tr>
<th>Machine 1 slots</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine 2 slots</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Time units</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Model GP for this example can be stated as follows.

<table>
<thead>
<tr>
<th>Row</th>
<th>(x_{13})</th>
<th>(x_{15})</th>
<th>(x_{16})</th>
<th>(x_{17})</th>
<th>(x_{23})</th>
<th>(x_{25})</th>
<th>(x_{27})</th>
<th>(x_{32})</th>
<th>(x_{38})</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>48</td>
<td>48</td>
<td>MIN</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>= 1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\leq 1)</td>
</tr>
</tbody>
</table>

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 \(x_i\) and the slack variables in the constraints (6.2) is as follows. \(E_1 = \{x_{13}, x_{15}, x_{16}, x_{17}\}\), \(E_2 = \{x_{23}, x_{25}, x_{27}\}\), \(E_3 = \{x_{32}, x_{38}\}\), 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 $x_{13} = 1$, $x_{23} = 1$, $x_{32} = 1$, and all the other variables are equal to zero. Consider the following description of $\text{GP}_{\text{strong}}$, where $Q = \{1, 2\}$, and $T_i = \{\text{time blocks 3 and 4}, T_2 = \{\text{time blocks 5, 6, 7, and 8}\}$. 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.

<table>
<thead>
<tr>
<th>Row</th>
<th>$x_{32}$</th>
<th>$x_{12}$</th>
<th>$x_{23}$</th>
<th>$x_{33}$</th>
<th>$x_{14}$</th>
<th>$x_{45}$</th>
<th>$x_{17}$</th>
<th>$x_{17}$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST</td>
<td>48</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>48</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>= 1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td>= 1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>= 1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>$\leq 1$</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\leq 1$</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>$\leq 1$</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\leq 1$</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>$\leq 1$</td>
</tr>
</tbody>
</table>

6.3. Tight Formulations Using Strong Valid Inequalities

The initial formulation of model $\text{GP}$ 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 GPP.

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 $\theta$, by writing its objective function as maximize $-\sum_{i \in N} \sum_{k \in S_i} \bar{c}_{ik} x_{ik}$ 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 $\sum_{q \in Q} \sum_{i \in N} \sum_{k \in (T_q \cap S_i)} \bar{c}_{ik} x_{ik} - n\theta$

subject to

$$\sum_{i \in S_k} x_{ik} \leq 1 \quad \text{for } i \in N$$  \hspace{1cm} (6.15)

$$\left[ \sum_{(i, \rho) \in I_k} x_{ik} \leq 1 \quad \text{for each block } q \in Q \right]$$  \hspace{1cm} (6.16)

$$x_{ik} \in \{0,1\} \quad \text{for } k \in S_i, \ i \in N$$  \hspace{1cm} (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,
\[
\text{CHGPP} = \text{conv}\{x \in \mathbb{R}^{\infty} : x \text{ satisfies (6.15), (6.16), (6.17)}\}, \quad \text{where } n_0 = \bigcup_{i \in N} S_i = \sum_{i \in N} |S_i|.
\]

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

\[
\sum_{i \in N} \sum_{k \in T_q \cap S_i} \Delta_{i,k} x_{i,k} \leq |T_q| \quad \text{for each } q \in Q. \tag{6.18}
\]

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 \(\Delta_{i,k} = \Delta_{i,j}\) 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,

\[
\text{GPKP} = \text{conv}\{x \in \mathbb{R}^{n_0} : \sum_{k \in S_i} x_{i,k} \leq 1 \text{ for } i \in N, \quad \sum_{i \in N} \sum_{k \in T_q \cap S_i} \Delta_{i,k} x_{i,k} \leq |T_q| \text{ for each } q \in Q, \quad x_{i,k} \in \{0, 1\} \text{ for } k \in S_i, \ i \in N \}
\]

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 \(\pi x \leq \pi_0\) is a facet defining inequality for GPKP if and only if (i) \(x \in \text{GPKP}\) implies \(\pi x \leq \pi_0\), 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 \( x_k \geq 0 \) 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 \( \sum \sum \pi_{ik} x_{ik} \leq \pi_0 \), denoted as \((\pi, \pi_0)\), 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 \( k \in S_i, \ i \in \mathbb{N} \), there exists a zero-one vertex \( \bar{x} \) of GPKP such that \( \bar{x}_k = 1 \) 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 \( \pi x^j = \pi_0 \), \( j = 1, ..., n_0 \). Let \( R \) be the matrix whose rows represent these vertices. If for some \( k \in S_i, \ i \in \mathbb{N} \), we have \( x^j_k = 0 \) for \( j = 1, ..., n_0 \), then this implies that \( R \) is singular. Hence, for each \( k \in S_i, \ i \in \mathbb{N} \), \( x^j_k = 1 \) for some \( j \in \{1, ..., n_0\} \), 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 \( \Delta_{i,k_1} = \Delta_{i,k_2} \) for all \( k_1, k_2 \in T_q \cap S_i \), \( (k_1 \neq k_2) \), for some \( i \in \mathbb{N} \), then \( \pi_{i,k_1} = \pi_{i,k_2} \) in any facet \( \sum_{i \in \mathbb{N} \cap S_i} \pi_{i,k} x_{i,k} \leq \pi_0 \) of GKP with \( \pi_0 > 0 \).

Proof. Suppose that \( \pi x \leq \pi_0 \) is a facet of GKP such that \( \pi_{i,k_1} \neq \pi_{i,k_2} \). Without loss of generality, let \( \pi_{i,k_1} < \pi_{i,k_2} \). Since \( \pi x \leq \pi_0 \) is a facet of GKP, by Proposition 6.4, there exists a zero-one vertex \( \bar{x} \in \text{GKP} \) such that \( \pi \bar{x} = \pi_0 \) and \( \bar{x}_{i,k_1} = 1 \), so that \( \bar{x}_{i,k_2} = 0 \) by (6.15). Denote \( \bar{x} = \bar{x} \), except that \( \bar{x}_{i,k_1} = 0 \) and \( \bar{x}_{i,k_2} = 1 \). Then \( \bar{x} \) is also feasible to GKP. But \( \pi \bar{x} = \pi_0 - \pi_{i,k_1} + \pi_{i,k_2} > \pi_0 \), leading to a contradiction. This completes the proof. \( \square \)

Proposition 6.5 implies that we can further restrict our investigation of the facial structure of GKP as follows. For each \( q \in Q \), let \( y_{i,q} = \sum_{k \in T_q \cap S_i} x_{i,k} \) for each \( i \in \mathbb{N} \) such that \( T_q \cap S_i \neq \emptyset \). For each \( q \in Q \), denote by \( \mu_q \) the common value of \( \mu_k \) for all \( k \in T_q \). Consider the convex hull of the following generalized assignment polytope, denoted by GPGAP, where \( n_q = \sum_{q \in Q} \left| \{ i \in \mathbb{N} : T_q \cap S_i \neq \emptyset \} \right| \), and where undefined variables are assumed to be zero.

\[
\text{GPGAP} = \text{conv}\{ y_q \in \mathbb{R}^{n_q} : \sum_{q \in Q} y_{i,q} \leq 1 \text{ for each } i \in \mathbb{N}, \\
\sum_{i \in \mathbb{N}} \Delta_{i,\mu_q} y_{i,q} \leq |T_q| \text{ for each } q \in Q, \\
y_{i,q} \in \{0,1\} \}
\]

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 \( \sum_{i \in \mathbb{N} \cap T_q} \pi_{i,q} y_{i,q} \leq \pi_0 \) with \( \pi_0 > 0 \) is a facet of GPGAP. Then
\[
\sum_{i \in N} \sum_{q \in Q} \sum_{k \in T_Q \cap S_i} \pi_{iq} x_{ik} \leq \pi_0
\]  
(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 \((\overline{\pi}_i, \overline{\pi}_j)\) \( j \in J \) of GPKP such that \( \pi = \sum_{j \in J} \lambda_j \overline{\pi}_j \) and \( \pi_0 = \sum_{j \in J} \lambda_j \overline{\pi}_j \) 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

\[
\sum_{i \in N} \sum_{q \in Q} \overline{\pi}_{iq} \left[ \sum_{k \in T_Q \cap S_i} x_{ik} \right] \leq \overline{\pi}_0 \quad \forall j \in J.
\]  
(6.20)

Now, note that for any \( j \in J \), \( \sum_{i \in N} \sum_{q \in Q} \overline{\pi}_{iq} y_{iq} \leq \overline{\pi}_0 \) are also valid inequalities for GPGAP. This follows because if for any \( j \in J \), we have \( \sum_{i \in N} \sum_{q \in Q} \overline{\pi}_{iq} \hat{y}_{iq} > \overline{\pi}_0 \) 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 $\overline{G}$. 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,

\[
( r_{1,1}, d_{1,1}, \Delta_{1,1}, w_{e,1} ) = (0, 9, 1, 1),
\]

\[
( r_{2,1}, d_{2,1}, \Delta_{2,1}, w_{e,1} ) = (0, 9, 1, 1),
\]

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

\[
( r_{4,1}, d_{4,1}, \Delta_{4,1}, w_{e,1} ) = (0, 9, 2, 1).
\]
and has the duration \([3,5]\), blocked-off on the single machine. The available time slots are numbered \(k = 1,\ldots,K = 7\).

<table>
<thead>
<tr>
<th>Machine 1 slots</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time units</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

An optimal solution \(\bar{x}\) to \(\overline{GP}\) is given by \(\bar{x}_{11} = 1\), \(\bar{x}_{22} = 1/2\), \(\bar{x}_{33} = 1/2\), \(\bar{x}_{43} = 1/2\), \(\bar{x}_{44} = 1/2\), 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

\[
x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + 2x_{32} + 2x_{33} + 2x_{42} + 2x_{43} \leq 3. \tag{6.21}
\]

Consider the corresponding continuous solution \(\bar{y}\) translated to \(y\) variables as given by \(\bar{y}_{11} = 1\), \(\bar{y}_{12} = 0\), \(\bar{y}_{13} = 0\), \(\bar{y}_{22} = 0\), \(\bar{y}_{31} = 0\), \(\bar{y}_{32} = 1\), \(\bar{y}_{33} = 1/2\), and \(\bar{y}_{42} = 1/2\). Corresponding to \((6.21)\), the knapsack constraint of \(GP\) 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 \(GP\) 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 \( M_\alpha \) denote indices of the GUB constraints (6.1), and similarly, let \( M_\rho \) denote indices of the set packing constraints (6.2). Note that each constraint \( i \in M_\alpha \) corresponds to a target \( i \), and each constraint \( k \in M_\rho \) corresponds to a slot \( k \), so that we can assume \( M_\alpha = \{1,\ldots,n\} \), and \( M_\rho = \{n+1,\ldots,n+K\} \). Define \( M_{\alpha\rho} = M_\alpha \cup M_\rho \). Let us also refer to the variables as being simply singly subscripted, and given by \( x_j \) for \( j \) belonging to an index set \( N_{\alpha\rho} \). Furthermore, denote by \( N_i \) the set of indices of variables contained in constraint \( i \in M_{\alpha\rho} \), and conversely, let \( M_i \) denote the index set of constraints in \( M_{\alpha\rho} \) that include the variable \( x_i \), for each \( j \in N_{\alpha\rho} \). Note that with this revised compact notation, model GP can be simply restated as

\[
\text{GP: Minimize } \{ \sum_{i \in N_{\alpha\rho}} c_i x_i : \sum_{j \in N_i} x_j = 1 \quad \forall \; i \in M_\alpha, \quad \sum_{j \in N_j} x_j \leq 1 \quad \forall \; i \in M_\rho, \quad x_j \in \{0, 1\} \quad \forall \; j \in N_{\alpha\rho} \},
\]

and similarly, we may restate model GPP as follows.

\[
\text{GPP: Maximize } \{ \sum_{i \in N_{\alpha\rho}} \bar{c}_i x_i : \sum_{j \in N_i} x_j \leq 1 \quad \forall \; i \in M_\alpha, \quad \sum_{j \in N_j} x_j \leq 1 \quad \forall \; i \in M_\rho, \quad x_j \in \{0, 1\} \quad \forall \; j \in N_{\alpha\rho} \}
\]

For simplicity of notation, let us rewrite model GPP as

\[
\text{GPP: Maximize} \{ \bar{c} x : A x \leq \bar{e}, \; x_j \in \{0, 1\} \},
\]

where \( \bar{e} = \{1,\ldots,1\}^T \). 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 \( A_{gi} \) 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 $G'_i$ of $G_i$, corresponding to the block interval $\sigma \in Q$. Hence, $G'_i = \{V', E'\}$, where $V' = \{j \in N_i : i \in T_\sigma\}$, and $E' = \{(i, j) \in E : i, j \in V'\}$. Note that $V = \bigcup_{\sigma \in Q} V'$. Let $R_4$ be a clique of the subgraph $G'_i$. Then,

$$\sum_{j \in R_4} x_j \leq 1 \quad (6.22)$$

is a valid inequality of GPP. Furthermore, we can obtain a complete subgraph corresponding to each GUB constraint of model GPP.

$$\sum_{j \in N_i} x_j \leq 1 \text{ for } i \in M_G \quad (6.23)$$

**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 $a_{j_1}^i a_{j_2} \geq 1,$ if $j_1$ and $j_2$ appear in some common constraint in $M_0,$ and $a_{j_1}^i a_{j_2} = 0,$ 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 $F = \{ j : \bar{x}^j \text{ is fractional } \} \neq \phi.$ 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 $\sum_{j \in H \cap \phi} \bar{x}^j > 1,$ 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.

\[
\begin{align*}
(r_{1,1}, d_{1,1}, \Delta_{1,1}, w_{1,1}) &= (0, 9, 2, 1), \quad (r_{1,2}, d_{1,2}, \Delta_{1,2}, w_{1,2}) &= (0, 9, 2, 1), \\
(r_{2,1}, d_{2,1}, \Delta_{2,1}, w_{2,1}) &= (0, 9, 3, 1), \quad (r_{2,2}, d_{2,2}, \Delta_{2,2}, w_{2,2}) &= (0, 9, 3, 1), \\
(r_{3,1}, d_{3,1}, \Delta_{3,1}, w_{3,1}) &= (0, 9, 4, 1), \quad (r_{3,2}, d_{3,2}, \Delta_{3,2}, w_{3,2}) &= (0, 9, 4, 1)
\end{align*}
\]

The machine blocked-off times are as follows.

Machine 1: [3,5]
Machine 2: [4,6]

\[
\begin{array}{cccccccc}
\text{Machine 1 slots} & 1 & 2 & 3 & & 4 & 5 & 6 & 7 \\
\hline
\text{Machine 2 slots} & 8 & 9 & 10 & 11 & & 12 & 13 & 14 \\
\text{Time units} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\end{array}
\]

An optimal solution \( \bar{x} \) to \( \overline{GP} \) is given by \( \bar{x}_1 = 1/2, \bar{x}_{1,1} = 1/2, \bar{x}_{2,3} = 1, \bar{x}_{3,7} = 1/2, \bar{x}_{3,11} = 1/2, \) and \( \bar{x}_a = 0 \) otherwise. For the solution \( \bar{x} \) to \( \overline{GP} \), variables \( x_{1,3}, x_{1,11}, x_{3,11} \) are fractional in block \( q = 3 \), where \( T_q = \{8, 9, 10, 11\} \). However, the indices of these variables \{\( x_{1,3}, x_{1,11}, x_{3,11} \)\} themselves yield a complete subgraph \( H \) of the subgraph of \( G^f \) induced by the fractional variables of \( V^f \). The node set of the extended complete subgraph of \( G^f \), which includes \( H \) in its node set, corresponds to the indices of the variables \{\( x_{1,3}, x_{1,10}, x_{1,11}, x_{2,10}, x_{2,11}, x_{3,11} \)\}. Hence, the corresponding clique inequality of type (6.22) is \( x_{1,3} + x_{1,10} + x_{1,11} + x_{2,10} + x_{2,11} + x_{3,11} \leq 1 \), 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 \( x_{1,3} = 1, x_{2,3} = 1, x_{3,11} = 1, \) and \( x_a = 0 \) 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 $x_i$ such that $x_i = 1$.

$P_i$: A set of variables corresponding to unassigned target $i$, which are forced to zero due to the partial schedule $PS$, i.e., $P_i = \{ j : j \in N_i \cap N_s, \text{ for all } s \in M_s \cap M_p, \forall k \in PS \}$.

PBS($i$): An index set of variables $x_j$ for $j \in (N_i - P_i)$, corresponding to an unassigned target $i \in M_0$ 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 $PS = \phi$, and $PBS(t) = \phi$ for all $i \in M_0$.

**STEP 1.** Determine the set $P_t$.

**STEP 2.** Find the index $j_t$ of target $t$ such that

$$j_t = \text{argmin}\{ c_j : j \in (N_t - (P_t \cup PBS(t))) \}$$

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 $PS = \phi$), then the given instance of problem GP is infeasible. Otherwise, perform the following steps.
3.1. Set PBS(t) = φ.
3.2. Replace PS ← PS - {j_{t-1}}.
3.3. Replace PBS(t - 1) ← PBS(t - 1) + {j_{t-1}}.
3.2. Replace t ← t - 1.
3.5. Go to STEP 2.

STEP 4. (Stopping criterion) If t = |M_0| = n, then stop; we have found a feasible solution to problem GP. Otherwise, set t ← 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 backtracking 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 \( \tau_{\text{HGP}} \) on the run time for the heuristic HGP. Whenever the procedure backtracks, if the elapsed time exceeds \( \tau_{\text{HGP}} \), 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 \( x_i \) in ascending order of \( \frac{c_i}{|M_i|} \).
**AR2:** Arrange the targets in ascending order of their release times. Break any ties by arranging the targets in descending order of $\Delta C_i$, where $\Delta C_i = c_i - c_{n_i}$, and where $j_1 = \arg\min_{i \in M_s} \{c_i\}$ and $j_2 = \arg\min_{i \in \{N_i, \ldots, 1\}} \{c_i\}$. The value of $\Delta C_i$ 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 $\Delta C_i$, 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 $\overline{GP}$ of problem GP. The prescribed modification of HGP, based on an optimal solution to $\overline{GP}$, operates as follows.
Heuristic HGPLP

**LINEAR PROGRAMMING STEP.** Solve $\overline{GP}$, and let $\overline{x}$ be an optimal solution. If $\overline{x}$ is integral, then stop; $\overline{x}$ also solves problem $GP$. Otherwise, $v(\overline{GP})$ is a lower bound on $v(GP)$. Proceed to the Main Step.

**MAIN STEP (Prioritized Search with HGP).** Partition the set of targets into $M^+ = \{i \in M_0 : \overline{x}_i = 1 \text{ for some } j \in N_i\}$ and $M_0 - M^+$. 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 $PS = \{j : \overline{x}_i = 1\}$, 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 $N_t^- = \{j \in N_t : \overline{x}_i > 0\}$, for $t \in M_0$. 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 $PS = \{j : \overline{x}_i = 1\}$, but now permitting the use of all variables in $N_t$ for $t \in M_0$.

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 \( u_i, i \in M \) be the Lagrange multiplier associated with the GUB constraints of model GP. Then, the corresponding Lagrangian relaxation of model GP, denoted by \( \text{LRGP}(u) \), can be constructed as follows.

\[
\text{LRGP}(u): \quad \text{Minimize} \quad \sum_{i \in M} \sum_{j \in N_i} (c_{ij} - u_i) x_{ij} + \sum_{i \in N} u_i
\]

subject to

\[
\sum_{j \in N_i} x_{ij} \leq 1 \quad \text{for } i \in M
\]

\[
x_{ij} \in \{0, 1\}, \quad j \in N_{GP}
\]

For any \( u \in R^{\text{IMG}} = R^n \), we have that \( \nu(\text{LRGP}(u)) \) is a lower bound on \( \nu(\text{GP}) \). 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:

\[
\text{LDGP}: \quad \text{Maximize} \quad \nu(\text{LRGP}(u)).
\]

Since the constraint set (7.1) has an interval matrix structure by Proposition 6.3, as well as a block-diagonal structure, \( \text{LRGP}(u) \) is a separable shortest path problem on an acyclic digraph \( G \). Consequently, for any \( u \in R^n \) every feasible extreme point solution to \( \text{LRGP}(u) \) is integral. This integrality condition leads to the following result.

**Proposition 7.1** \( \nu(\overline{\text{GP}}) = \nu(\text{LDGP}). \) ■

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.

\textbf{Heuristic HGPLD}

\textbf{INITIALIZATION} Let $N_t$ be the maximum number of iterations permitted, choose $\varepsilon$ 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 $u^1$, and denote by $\hat{z}$ the incumbent solution value of LDGP. (Initially, set $\hat{z} = -\infty$.) Proceed to Step 1.

\textbf{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 $z^k = v(LRGP(u^k))$, and compute a subgradient $\xi^k$ to the Lagrangian dual function $v(LRGP(u))$ at $u = u^k$ as $\xi^k_\ell = 1 - \sum_{i \in M_\ell} \bar{x}^k_\ell$, $i \in M_\ell$. If $z^k > \hat{z}$, then set $\hat{z} = z^k$ and go to Step 3. Otherwise, proceed to Step 2.

\textbf{STEP 2. (Determination of New Dual Solution)} Find a direction of motion $d^k$ and select a step size $\lambda^k$. (Prescribed techniques for making these selections are addressed.) Update the Lagrange multipliers according to $u^{k+1}_i = u^k_i + \lambda^k d^k_i \text{ for each } i \in M_\ell$. If $k \geq N_t$, then terminate. Otherwise increment $k$ by 1 and return to Step 1.
STEP 3. (Application of Heuristic HGP for Problem GP) Let \((\overline{\mathbf{u}}^k, i \in M_{\overline{a}})\) be the complete dual solution corresponding to \((\overline{\mathbf{u}}^*, i \in M_a)\), where the remaining dual values \(\overline{u}_i, i \in M_a\) are obtained from the solution to LRGP(\(\overline{u}^*\)). Define the reduced cost coefficients \(\overline{c}_i = c_i - \sum_{j \in M_{\overline{a}}} \overline{u}_j^*\), \(\forall j \in N_{\overline{a}},\) denote \(M_{\overline{a}} = \{i \in M_a : \sum_{j : x_{ij}^* = 1} \overline{c}_j = 1\}\) and let \(M_{\overline{a}} = M_a - M_{\overline{a}}\). (Recall that \(\overline{x}^k\) is a binary solution to the set packing constraints). If \(M_{\overline{a}} = \phi\), then stop; \(\overline{x}^k\) is an optimal solution for problem GP. Otherwise, order the targets separately within each of the two sets \(M_{\overline{a}}\) and \(M_{\overline{a}}\) according to Remark 7.1, and concatenate these in the stated order. Then, implement heuristic HGP by starting with an advanced starting partial solution \(PS = \{j : \overline{x}_j^k = 1 \text{ and } j \in N , \text{ for some } i \in M_{\overline{a}}\}\) where the targets in \(PS\) are arranged as they are in \(M_{\overline{a}}\). (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 \(\overline{z}\) of the resulting solution is lesser than \(\overline{Z}\), then let this solution be the revised prescribed heuristic solution to GP and set \(\overline{z} = \overline{z}\). Now, if \(\frac{(\overline{z} - \overline{z})}{\overline{Z}} \leq \epsilon\), then stop; the optimal solution value for GP is within \((1 - \epsilon) \times 100\%\) of the current incumbent solution value. Also, if \(||\overline{z}^*||\) is insignificantly small, then stop; \(u^*\) 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 $\text{HDGP}$ given below determines an advanced-start dual feasible solution to $\text{GP}$, that can be used in the Initialization Step of the Lagrangian heuristic $\text{HGPLD}$. Note that defining $u_i$ as the dual variable corresponding to constraint $i$, $i \in M_{\text{GP}}$, the dual to $\text{GP}$ can be written as follows.

$$\text{DGPG: Maximize } \{ \sum_{i \in M_{\text{GP}}} u_i : \sum_{i \in M_j} u_i \leq c_j \text{ for } j \in N_{\text{GP}},$$

$$u_i : \text{unrestricted for } i \in M_G, \ u_i \leq 0 \text{ for } i \in M_P \}.$$  

Dual Heuristic $\text{HDGP}$

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 $R_i = c_j - u_{j(G)}$, where $j(G) = M_j \cap M_G$, for all $j \in N_{\text{GP}}$. Define a violation measure $V_i = \sum_{j \in N_i} |R_j|$, where $N_i = \{ j \in N_i : R_j < 0 \}$, for each $i \in M_P$.

STEP 3. Pick $t = \arg \max \{ V_i : i \in M_P \}$. If $V_t = 0$, go to Step 5. Otherwise, put $u_t = \min \{ R_j : j \in N_i \}$.

STEP 4. Dualize the set packing constraint $teM_P$ using the dual variable $u_t$. Accordingly, replace $R_i$ by $R_i - u_t$ for all $j \in N_i$. Hence, we can now replace $\bar{N}_i$ by $\bar{N}_i - \bar{N}_i \cap \bar{N}_t$ and $V_i$ by $V_i - \sum_{j \in N_j \cap \bar{N}_t} |R_j|$ 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 $R_i = c_i - \sum_{j \in M_p \cap N_j} u_j$, for all $j \in N_{c_p}$. Put $u_i = \min \{ R_i : j \in N_i \}$ for each $i \in M_0$, leave $u_i$ as at present, and stop with the prescribed dual solution at hand.

**Proposition 7.2** Heuristic HDGP generates a dual feasible solution for $\overline{GP}$ with an effort having complexity $O(mnT^2)$.

**Proof.** Whenever $v_i = 0$ 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 $M_0$, 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 $|N_{c_p}|$ 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 $v(GP)$. That is, $\sum_{i \in M_0} \min_{j \in N_i} \{ c_i \} \leq v(GP) \leq \sum_{i \in M_0} \max_{j \in N_i} \{ c_i \}$.

**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):
\[ d^i = \xi^i, \text{ and } d^k_i = \xi^k_i + \frac{||\xi^k||}{||d^{k-1}||} d^{k-1}_i, \quad \forall i \in \mathcal{G}, \quad \text{for } k \geq 2 \] (7.3)

where \( ||.|| \) denotes the Euclidean norm. Note that the direction \( d^i \) simply bisects the angle between \( \xi^i \) 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 \( \lambda^k \) 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

\[ \lambda^k = \beta^k \frac{\tilde{z} - z^k}{||d^k||^2} \] (7.4)

where \( \tilde{z} \) is the current best known objective value for problem GP, and \( \beta^k \) 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 \( N_z \) such that at the top of each block when \( k \mod(N_z) = 1 \), we compute the step length \( \lambda^k \) using the formula (7.3) and with \( \beta^k = \bar{\beta}/2^{n/4} \). Then, within each block, the step size
is held constant, except that it is halved each time the process goes through some $N_i$ 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 $N_i$ 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 $\bar{p} = 0.75$. 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 de-
development of a branching tree, the node selection rule employed, and the fathoming process.

Notationally, let $x^*$ denote the current best primal feasible solution found thus far, and let UB denote the upper bound corresponding to $x^*$. 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 $\min_{j \in M} (r_{ij})$. Define $\overline{n} = \min \{ h : \max_{j \in M} (d_{ij}) \leq \min_{j \in M} (r_{h+1,j}) \},$ for $1 \leq i \leq h$, where $r_{n+1,j} \equiv \infty$. Then, we can decompose the set of jobs into sets $N_1$ and $N_2$, where $N_1 = \{1, \ldots, \overline{n}\}$ 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 \(i \in M_o\), then we can set the corresponding variable \(x_i = 1\), delete \(x_i\) in the variable set, delete the set of constraints \(M_i\) from the model, and also put \(x_k = 0\) for all \(k \in N_o\), \(t \in M_o\), \(k \neq i\). In the same spirit, if for some \(j \in N_i\), there exists an \(i \in M_o\), \(j \neq N_i\), such that \(\bigcup_{k \in N_i} (M_k \cap M_o) \subseteq M_j \cap M_o\), 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 \(j \in N_o\) for which \(\bar{c}_j = c_j - \sum_{i \in N_j} u^i > UB - LB_o\), since \(x_i = 1\) 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 \(k\)th level of the branching tree. At the \(k\)th 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 \(u^i\), \(i \in M_o\). As a second rule, we dynamically select a branching constraint \(i(k)\) at the \(k\)th level of the branching tree such that \(i(k) \in \text{argmax} \{ u^i : i \in \text{unbranched constraints} \}\).
in the chain from node \( k \) to the root node \( j \) 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 \( k \)th level of the branching tree, we create a node for each \( j \in N_{(k)} \), fixing \( x_j = 1 \) and \( x_i = 0 \) for all \( t \in (N_{(k)} - j) \) at each such node. These nodes for level \( (k+1) \) are stored in a linked list of open nodes, denoted by \( \Lambda \), in descending order of the reduced costs \( \overline{c}_i = c_i - \sum_{i \in M_j} u^k \), 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 \( k \)th level of the branching tree, obtained by fixing some variable \( x_j = 1 \), \( j \in N_{(k)} \), for each level \( s < k \) of the branching tree. Specifically, let \( FS(k) \) be the set of indices of variables \( x_j \), \( j \in N_{(k)} \), \( s = 0, \ldots, k-1 \), that are restricted to unity on the unique chain of the branching tree traced from node \( k \) to the root node. At the \( k \)th level of the branching tree, we generate for level \( (k+1) \) a set of subproblems \( P_s^{k+1} \), for \( s \in N_{(k)} \), that can be described as follows.
\[ P_{s+1}^k: \text{Minimize} \left\{ \sum_{j \in N_{GP}} c_j x_j : \sum_{j \in N_i} x_j = 1 \quad \forall i \in M_G, \right. \\
\left. \sum_{j \in N_i} x_j \leq 1 \quad \forall i \in M_p, \right. \\
x_i = 1 \quad \forall t \in FS(k), \quad x_s = 1, \quad x_j \in \{0, 1\} \quad \forall j \in N_{GP} \}\]

Note that each Problem \( P_s \), for \( s \in N_{(k-1)} \), has the same structure as problem GP, but the size is reduced. Problem \( P_s \) 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 \( P_s \) is infeasible if \((M_p \cap M_s) \cap (\bigcup_{i \in FS(k)} M_i) \neq \emptyset\). These trivial infeasible problems are not stored in the list \( \Lambda \) of open nodes. Instead of storing an explicit representation of the formulation \( P_s \), or for that matter, instead of even storing an explicit representation of \( \Lambda \), we record minimal information that can be used to recover the list \( \Lambda \), as well as recover for each node in the list \( \Lambda \), 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 \( \Lambda \), 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 \( \Lambda \) 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 \((u^i, i \in M_\omega)\) 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. \(\varepsilon = \) guaranteed optimality tolerance parameter. Hence, whenever \(\frac{(UB - LB_i)}{UB} \leq \varepsilon\), 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. \(\tau_{bb} = \) the maximum allowed run time. If the total run time of the branch-and-bound algorithm, checked at the top of the loop, exceeds \(\tau_{bb}\), then the algorithm is terminated, and the current best feasible solution is taken as the prescribed solution.
3. $\tau_{\mathrm{HGP}}$ = 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 $x^*$ for problem GP via the heuristic HGP of Section 7.1.1, and let $\text{UB}$ be its objective value. Also select an initial Lagrangian dual vector $u^0$ as $u^0_i = \min \{c_i\} \ \forall \ i \in M_0$, and denote by $\text{LB}_0 = \sum_{i \in M_0} u^0_i$ the incumbent solution value of LDGP. If $\frac{(\text{UB} - \text{LB}_0)}{\text{UB}} \leq \varepsilon$, for some tolerance $0 \leq \varepsilon < 1$, terminate with $x^*$ 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 $u^\phi$. Also denote by $\text{LB}_\phi$ the (near) optimal objective function value of LDGP. If $\frac{(\text{UB} - \text{LB}_\phi)}{\text{UB}} \leq \varepsilon$, then stop with $x^*$ as an $\varepsilon$-optimal solution. Otherwise, execute the variable-fixing procedure using the reduced cost vector $\bar{c}^0$. Put $\text{FS}(k) = \phi$, 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 $J_{\phi_0} = \{j \in N_{\phi_0} : M_j \cap M_i = \emptyset \ \forall \ l \in \text{FS}(k)\}$. (Note that if $J_{\phi_0} = \phi$, then the subproblem at the preceding level would have been infeasible, even in the continuous sense.) Construct subproblems $P^{k+1}_s$ for all $s \in J_{\phi_0}$, and add these subproblems to the linked list $\Lambda$ of open nodes, according to a descending order of the reduced costs $\bar{c}_s$. 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 $\tau_{\text{BB}}$, then stop. Otherwise, proceed to Step 5.
**STEP 5. (Problem Selection and Relaxation)** Select the subproblem \( P_k^t \) which is the most recent member of the list \( \Lambda \), and delete it from \( \Lambda \). Let \( \bar{f}(k) = f(k-1) + \{t\} \). Solve the Lagrangian dual of Problem \( PS_k^t \) 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 \( x^k \) denote an optimal solution to the Lagrangian relaxation subproblem of Problem \( P_k^t \), corresponding to the (near) optimal dual solution \( u^k \) obtained at Step 5. Denote \( M_\delta = \{ i \in M_\delta : \sum_{[j]} x^k_\delta = 1 \} \) and let \( M_\delta = M_\delta - M_\delta \). If \( M_\delta = \phi \), then \( x^k \) is an optimal solution for Problem \( P_k^t \). In addition, if \( x^k \) is integral, it is optimal to \( P_k^t \), and so, if \( LB_k < UB \), then let \( UB = LB_k \), and let \( x^k = x \), and go to Step 7. Otherwise, order the variables separately within each of the two sets \( M_\delta \) and \( M_\delta \) in ascending order of \( \frac{c_i}{|M_i|} \) 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 \( PS = \{ j : x^k_j = 1 \} \) for \( j \in N_\delta, i \in M_\delta \), where the jobs in \( PS \) are arranged as they are in \( M_\delta \). If HGP terminates before the time limit \( \tau_{\text{heur}} \) 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 \( x^k \) by this improved heuristic solution to GP, and set \( UB = cx^k \). 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 \( \frac{(UB - LB_k)}{UB} \leq \varepsilon \), then stop; we have found an \( \varepsilon \)-optimal solution \( x^k \) to problem GP. Otherwise, let \( P_k^t \) be the most recent subproblem recorded in \( \Lambda \). 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 $r_{ij}$ were generated uniformly on the interval $[0, 50]$, the processing times $\Delta_{ij}$ were generated uniformly over $[3, 6]$, and the deadlines $d_{ij}$ were computed as $d_{ij} = r_{ij} + \Delta_{ij} + \delta_{ij}$, where $\delta_{ij}$ 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 $[0, \max_{ij}(d_{ij})]$. Also, the weights $w \equiv (w_{ik}, \ k \in S_i, \ i \in \mathcal{N})$ were taken as either $w^1$ or $w^2$ specified below, again using NSWC guidelines.

$$w^1_{ik} = \frac{\max_{p,q}(d_{p,q} - r_{p,q})^2}{(d_{i,k} - r_{i,k})^2}$$ \quad \text{and} \quad w^2_{ik} = \frac{10^5}{(d_{i,k} - r_{i,k})^2} \quad \forall k \in S_i, \ i \in \mathcal{N} \quad (8.1)$$

Recall that the objective function of problem GP seeks to minimize $\sum_{i \in \mathcal{N}} \sum_{k \in S_i} w_{ik} x_{ik}$. Hence, using (8.1), a preference is given to assigning target $i$ to that slot $k$ on illuminator $\mu_k$ for which not only does that engagement occur early in the time horizon (i.e., $\tau_k$ is small), but also the corresponding engageability time-window duration $(d_{i,k} - r_{i,k})$ 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 $\tau_k$ using illuminator $\mu_k$. (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 $W_{i,k}$ 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 $W_{i,k}$ plays the inverse role of $w_{i,k}$ in the minimization objective of Problem GP.) Then, we can use the following objective function in problem GP.

$$\text{Maximize} \sum_{i \in N} \sum_{k \in S_i} W_{i,k} f_{i,k} x_{i,k}$$  \hspace{1cm} (8.2)

This objective function (8.2) seeks to maximize the total expected reward of successful strikes. If $W_{i,k} = 1$, $\forall (i,k)$, then (8.2) maximizes the aggregate probability of successful (damaging) hits. The cost function (8.2) requires the specification of an appropriate weight function $w_{i,k}$. One such function can be specified as follows.

$$W_{i,k} = (d_{i,\mu_k} - \tau_k)^2$$  \hspace{1cm} (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 \( \tau_k \) 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

\[
W_{ik} = \frac{(d_{i,\mu_k} - r_{i,\mu_k})^2}{\tau_k}
\]  

(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 solv-
ing 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 \( w' \), 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 \( w' \) 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 \( w' \). Note that the simplex run times are 5-10 times that for applying the entire heuristic HGPLD, while the conjugate sub-gradient 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 \( w^i \) 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 \( \tau_{gap} = 1 \) 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of Targets</th>
<th>Number of Illuminators</th>
<th>Number of Variables of Model GP</th>
<th>Number of Rows of Model GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>30</td>
<td>6</td>
<td>1015</td>
<td>150</td>
</tr>
<tr>
<td>Problem 2</td>
<td>40</td>
<td>8</td>
<td>1908</td>
<td>243</td>
</tr>
<tr>
<td>Problem 3</td>
<td>40</td>
<td>5</td>
<td>1591</td>
<td>347</td>
</tr>
<tr>
<td>Problem 4</td>
<td>45</td>
<td>5</td>
<td>1981</td>
<td>370</td>
</tr>
<tr>
<td>Problem 5</td>
<td>45</td>
<td>7</td>
<td>2634</td>
<td>460</td>
</tr>
<tr>
<td>Problem 6</td>
<td>50</td>
<td>5</td>
<td>1178</td>
<td>170</td>
</tr>
<tr>
<td>Problem 7</td>
<td>50</td>
<td>5</td>
<td>2108</td>
<td>387</td>
</tr>
<tr>
<td>Problem 8</td>
<td>50</td>
<td>5</td>
<td>2141</td>
<td>373</td>
</tr>
<tr>
<td>Problem 9</td>
<td>55</td>
<td>7</td>
<td>1278</td>
<td>502</td>
</tr>
<tr>
<td>Problem 10</td>
<td>50</td>
<td>8</td>
<td>4084</td>
<td>533</td>
</tr>
<tr>
<td>Problem 11</td>
<td>20</td>
<td>3</td>
<td>455</td>
<td>175</td>
</tr>
<tr>
<td>Problem 12</td>
<td>25</td>
<td>7</td>
<td>1581</td>
<td>456</td>
</tr>
<tr>
<td>Problem 13</td>
<td>30</td>
<td>5</td>
<td>1256</td>
<td>318</td>
</tr>
</tbody>
</table>
Table 8.1 (Continued)  Size Characteristics of the Test Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of Targets</th>
<th>Number of Illuminators</th>
<th>Number of Variables of Model</th>
<th>Number of Rows of Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 14</td>
<td>35</td>
<td>7</td>
<td>2041</td>
<td>422</td>
</tr>
<tr>
<td>Problem 15</td>
<td>40</td>
<td>5</td>
<td>1768</td>
<td>355</td>
</tr>
<tr>
<td>Problem 16</td>
<td>40</td>
<td>6</td>
<td>3014</td>
<td>410</td>
</tr>
<tr>
<td>Problem 17</td>
<td>45</td>
<td>7</td>
<td>2684</td>
<td>460</td>
</tr>
<tr>
<td>Problem 18</td>
<td>45</td>
<td>8</td>
<td>3123</td>
<td>553</td>
</tr>
<tr>
<td>Problem 19</td>
<td>50</td>
<td>6</td>
<td>2141</td>
<td>373</td>
</tr>
<tr>
<td>Problem 20</td>
<td>50</td>
<td>8</td>
<td>3289</td>
<td>585</td>
</tr>
<tr>
<td>Problem 21</td>
<td>59</td>
<td>7</td>
<td>3275</td>
<td>502</td>
</tr>
<tr>
<td>Problem 22</td>
<td>55</td>
<td>8</td>
<td>3588</td>
<td>550</td>
</tr>
<tr>
<td>Problem 23</td>
<td>60</td>
<td>8</td>
<td>4084</td>
<td>583</td>
</tr>
<tr>
<td>Problem 24</td>
<td>60</td>
<td>7</td>
<td>3578</td>
<td>460</td>
</tr>
<tr>
<td>Problem 25</td>
<td>65</td>
<td>7</td>
<td>3809</td>
<td>523</td>
</tr>
<tr>
<td>Problem 26</td>
<td>65</td>
<td>8</td>
<td>4430</td>
<td>573</td>
</tr>
</tbody>
</table>
Table 8.2  Computational Results for HGP and HGPLP using Weight Function $w^1$.

<table>
<thead>
<tr>
<th>Problem</th>
<th>HGP</th>
<th></th>
<th>XMP (Dual Simplex)</th>
<th></th>
<th>HGPLP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$v(HGP)$</td>
<td>Run time</td>
<td>$v(GP)$</td>
<td>Run time</td>
<td>$v(HGPLP)$</td>
<td>Run time</td>
</tr>
<tr>
<td>Problem 1</td>
<td>656</td>
<td>0.09</td>
<td>519.5</td>
<td>4.9</td>
<td>606</td>
<td>6.8</td>
</tr>
<tr>
<td>Problem 2</td>
<td>843</td>
<td>0.02</td>
<td>854.7</td>
<td>13.1</td>
<td>862</td>
<td>12.8</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2070</td>
<td>0.09</td>
<td>2014.5</td>
<td>6.1</td>
<td>2018</td>
<td>7.21</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2210</td>
<td>0.18</td>
<td>2102.6</td>
<td>10.1</td>
<td>2209</td>
<td>12.2</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2125</td>
<td>0.19</td>
<td>2094.5</td>
<td>14.5</td>
<td>2090</td>
<td>18.5</td>
</tr>
<tr>
<td>Problem 6</td>
<td>1111</td>
<td>0.09</td>
<td>198</td>
<td>14.5</td>
<td>1010</td>
<td>16.72</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2756</td>
<td>0.2</td>
<td>2567.5</td>
<td>12.44</td>
<td>2591</td>
<td>13.7</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2897</td>
<td>0.3</td>
<td>2525.8</td>
<td>12.1</td>
<td>2548</td>
<td>16.2</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2005</td>
<td>0.3</td>
<td>2150.8</td>
<td>33.15</td>
<td>2161</td>
<td>32.2</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2639</td>
<td>0.4</td>
<td>2592.0</td>
<td>32.2</td>
<td>2618</td>
<td>32.3</td>
</tr>
</tbody>
</table>
Table 8.3  Computational Results for Heuristic HGPLD using Weight Function w1.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>v(HGPLD)</th>
<th>v(LDGP)</th>
<th>Run time (unit = seconds)</th>
<th>% Optimality (ε = 0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSBH</td>
<td>HWCBH</td>
<td>ADSBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>606</td>
<td>537.9</td>
<td>569.7</td>
<td>1.7</td>
</tr>
<tr>
<td>Problem 2</td>
<td>890</td>
<td>843.5</td>
<td>940.1</td>
<td>2.4</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2070</td>
<td>1971.4</td>
<td>1971.4</td>
<td>0.2</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2152</td>
<td>2058.9</td>
<td>2080.9</td>
<td>1.8</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2124</td>
<td>2022.9</td>
<td>2133.2</td>
<td>0.6</td>
</tr>
<tr>
<td>Problem 6</td>
<td>998</td>
<td>950.5</td>
<td>961.7</td>
<td>2.4</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2640</td>
<td>2596</td>
<td>2528.4</td>
<td>1.7</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2590</td>
<td>2532</td>
<td>2515.3</td>
<td>1.3</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2187</td>
<td>2099.2</td>
<td>2089.2</td>
<td>0.8</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2638</td>
<td>2638</td>
<td>2515</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>v(HGPLD)</th>
<th>v(LDGP)</th>
<th>Run time (unit = seconds)</th>
<th>% Optimality (ε = 0.01)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSBH</td>
<td>HWCBH</td>
<td>ADSBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>606</td>
<td>598</td>
<td>578.1</td>
<td>2.5</td>
</tr>
<tr>
<td>Problem 2</td>
<td>875</td>
<td>879</td>
<td>852.8</td>
<td>4.4</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2028</td>
<td>2028</td>
<td>2010.4</td>
<td>0.8</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2142</td>
<td>2142</td>
<td>2102.1</td>
<td>6.5</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2092</td>
<td>2090</td>
<td>2070.0</td>
<td>2.4</td>
</tr>
<tr>
<td>Problem 6</td>
<td>998</td>
<td>1002</td>
<td>988.1</td>
<td>4.2</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2579</td>
<td>2596</td>
<td>2568.2</td>
<td>4.9</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2684</td>
<td>2584</td>
<td>2524.5</td>
<td>7.1</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2168</td>
<td>2170</td>
<td>2149.5</td>
<td>7.8</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2695</td>
<td>2610</td>
<td>2590.8</td>
<td>2.1</td>
</tr>
</tbody>
</table>
Table 8.4  Further Computational Results for Heuristic HGPLY with $\varepsilon = 0.05$.

<table>
<thead>
<tr>
<th>Problem</th>
<th>XMP (Weight function $w^1$)</th>
<th>ADSBH (Weight function $w^1$)</th>
<th>ADSBH (Weight function $w^2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$v(GP)$</td>
<td>Run time</td>
<td>Solution</td>
</tr>
<tr>
<td>Problem 11</td>
<td>1159</td>
<td>1.88</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 12</td>
<td>1017</td>
<td>4.01</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 13</td>
<td>1849</td>
<td>3.92</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 14</td>
<td>1702</td>
<td>7.35</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 15</td>
<td>2052.0</td>
<td>7.31</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 16</td>
<td>1849</td>
<td>7.31</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 17</td>
<td>2093.5</td>
<td>16.2</td>
<td>Fractional</td>
</tr>
<tr>
<td>Problem 18</td>
<td>2112</td>
<td>14.12</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 19</td>
<td>2524.8</td>
<td>14.9</td>
<td>Fractional</td>
</tr>
<tr>
<td>Problem 20</td>
<td>2112</td>
<td>20.1</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 21</td>
<td>2150.9</td>
<td>34.1</td>
<td>Fractional</td>
</tr>
<tr>
<td>Problem 22</td>
<td>2159.5</td>
<td>21.1</td>
<td>Fractional</td>
</tr>
<tr>
<td>Problem 23</td>
<td>2592.0</td>
<td>52.2</td>
<td>Fractional</td>
</tr>
<tr>
<td>Problem 24</td>
<td>3277</td>
<td>34.8</td>
<td>Integer</td>
</tr>
<tr>
<td>Problem 25</td>
<td>-</td>
<td>&gt; 120</td>
<td>-</td>
</tr>
<tr>
<td>Problem 26</td>
<td>2975</td>
<td>92.83</td>
<td>Fractional</td>
</tr>
</tbody>
</table>

# Obj. Ratio = $\frac{v(LDGP)}{v(GP)}$, 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 \( w' \) and \( \varepsilon = 0.01 \).

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>Best UB found</th>
<th>( v'(LDGP) ) at root node</th>
<th>Run time/# nodes</th>
<th>% Optimality ( ^1(\varepsilon = 0.01) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSBH</td>
<td>HWCBH</td>
<td>ADSBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>581</td>
<td>582</td>
<td>579.4</td>
<td>579.5</td>
</tr>
<tr>
<td>Problem 2</td>
<td>858</td>
<td>867</td>
<td>854.7</td>
<td>855.1</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2020</td>
<td>2029</td>
<td>2015.3</td>
<td>2015.3</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2142</td>
<td>2100</td>
<td>2102.4</td>
<td>2102.4</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2088</td>
<td>2105</td>
<td>2082.8</td>
<td>2083.1</td>
</tr>
<tr>
<td>Problem 6</td>
<td>998</td>
<td>1000</td>
<td>988.1</td>
<td>987.1</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2590</td>
<td>2570</td>
<td>2567.1</td>
<td>2567.2</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2557</td>
<td>2552</td>
<td>2525.4</td>
<td>2525.7</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2165</td>
<td>2171</td>
<td>2150.2</td>
<td>2150.3</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2813</td>
<td>2592</td>
<td>2810.9</td>
<td>2811.8</td>
</tr>
</tbody>
</table>

\(^1\) Optimality = 100 \times \frac{1-(Best \ UB - v'(LDGP))}{Best \ UB}
Table 8.5 (Continued)  Computational Results for Algorithm GPBB using the Fixed-Order Branching Rule along with Weight Function $w$ and $\epsilon = 0.005$.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>Best Ut found</th>
<th>v(LDGP) at root node</th>
<th>Run time/4 nodes</th>
<th>% Optimality ($\epsilon = 0.005$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSBH</td>
<td>HWCBH</td>
<td>ADSBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>591</td>
<td>582</td>
<td>579.3</td>
<td>579.3</td>
</tr>
<tr>
<td>Problem 2</td>
<td>859</td>
<td>889</td>
<td>854.7</td>
<td>955.1</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2020</td>
<td>2026</td>
<td>2015.4</td>
<td>2015.5</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2142</td>
<td>2160</td>
<td>2102.4</td>
<td>2102.3</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2085</td>
<td>2097</td>
<td>2082.8</td>
<td>2083.3</td>
</tr>
<tr>
<td>Problem 6</td>
<td>998</td>
<td>1000</td>
<td>999.2</td>
<td>997.1</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2574</td>
<td>2573</td>
<td>2587.6</td>
<td>2587.3</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2547</td>
<td>2544</td>
<td>2525.4</td>
<td>2525.7</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2160</td>
<td>2180</td>
<td>2150.8</td>
<td>2152.6</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2601</td>
<td>2621</td>
<td>2551.8</td>
<td>2591.8</td>
</tr>
</tbody>
</table>
Table 8.6  Computational Results for Algorithm GPBB using the Dynamic Branching Rule along with Weight Function  w' and ε = 0.01.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>Best UB found</th>
<th>v(LDGPI) at root node</th>
<th>Run time/# nodes</th>
<th>% Optimality (ε = 0.01)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADBH</td>
<td>HWCBH</td>
<td>ADBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>581</td>
<td>582</td>
<td>579.4</td>
<td>579.3</td>
</tr>
<tr>
<td>Problem 2</td>
<td>859</td>
<td>867</td>
<td>854.7</td>
<td>855.1</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2020</td>
<td>2029</td>
<td>2015.5</td>
<td>2011.5</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2142</td>
<td>2180</td>
<td>2102.4</td>
<td>2102.4</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2098</td>
<td>2105</td>
<td>2092.8</td>
<td>2093.3</td>
</tr>
<tr>
<td>Problem 6</td>
<td>996</td>
<td>1000</td>
<td>967.1</td>
<td>967.1</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2590</td>
<td>2579</td>
<td>2567.1</td>
<td>2567.3</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2557</td>
<td>2552</td>
<td>2525.4</td>
<td>2525.7</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2185</td>
<td>2171</td>
<td>2150.2</td>
<td>2150.3</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2913</td>
<td>2617</td>
<td>2590.9</td>
<td>2591.9</td>
</tr>
</tbody>
</table>
### Table 8.6 (Continued)  Computational Results for Algorithm GPBB using the Dynamic Branching Rule along with Weight Function $w^t$ and $\epsilon = 0.005$.  

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>Best UB found</th>
<th>$w(LODP)$ at root node</th>
<th>Run time/# nodes</th>
<th>% Optimality ($\epsilon = 0.005$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSBH</td>
<td>HWCBH</td>
<td>ADSBH</td>
<td>HWCBH</td>
</tr>
<tr>
<td>Problem 1</td>
<td>581</td>
<td>582</td>
<td>579.9</td>
<td>579.3</td>
</tr>
<tr>
<td>Problem 2</td>
<td>850</td>
<td>851</td>
<td>854.7</td>
<td>855.1</td>
</tr>
<tr>
<td>Problem 3</td>
<td>2020</td>
<td>2029</td>
<td>2015.4</td>
<td>2015.5</td>
</tr>
<tr>
<td>Problem 4</td>
<td>2142</td>
<td>2150</td>
<td>2197.4</td>
<td>2192.9</td>
</tr>
<tr>
<td>Problem 5</td>
<td>2068</td>
<td>2077</td>
<td>2092.8</td>
<td>2093.3</td>
</tr>
<tr>
<td>Problem 6</td>
<td>898</td>
<td>1000</td>
<td>889.1</td>
<td>887.1</td>
</tr>
<tr>
<td>Problem 7</td>
<td>2566</td>
<td>1573</td>
<td>2567.2</td>
<td>2567.9</td>
</tr>
<tr>
<td>Problem 8</td>
<td>2050</td>
<td>2544</td>
<td>2525.4</td>
<td>2525.7</td>
</tr>
<tr>
<td>Problem 9</td>
<td>2150</td>
<td>2160</td>
<td>2150.6</td>
<td>2150.6</td>
</tr>
<tr>
<td>Problem 10</td>
<td>2601</td>
<td>2617</td>
<td>2591.8</td>
<td>2591.9</td>
</tr>
</tbody>
</table>
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 \( i \in \mathbb{N} \equiv \{1, \ldots, n\} \), \( m \) illuminators, indexed \( j \in \mathbb{M} \equiv \{1, \ldots, m\} \), and that there are \( p \) launchers, indexed \( l \in \mathbb{L} \equiv \{1, \ldots, p\} \). Each target \( i \) has a specified availability duration \([r_{i,l}, d_{i,l}]\), 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 \( \text{problem EGP} \).

We assume that the processing times \( \Delta_{i,j} \), and the target availability interval (time-window) endpoints \( r_{i,l} \) and \( d_{i,l} \), as well as the end points of each unavailability interval duration on each of the illuminators \( j \in \mathbb{M} \), are all nonnegative integer valued data. Under this assumption, letting \( T = \max_{i,j} \{d_{i,j}\} \) 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 \( k \in \mathbb{K} \equiv \{1, \ldots, K\} \) index all the available or unassigned slots over all the illuminators.
To formulate problem EGP, let $\tau_k$ be the upper interval end-point of slot $k$, for $k \in K$. For each target $i \in \mathbb{N}$ and each slot $k \in K$, define $L_k = \{l \in L : [\tau_k - \Delta_{i,\mu_k}, \tau_k] \subseteq [l, l] \}$. We also define for each target $i \in \mathbb{N}$, a set

$$S_i = \{k \in K : [\tau_k - \Delta_{i,\mu_k}, \tau_k] \subseteq [l, l], \text{and} \quad (\tau_k - \Delta_{i,\mu_k}, \tau_k) \cap \text{[unavailability intervals]} \}$$

for illuminator $\mu_k = \phi$, for some launcher $l \in L$.

where $\mu_k$ 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 $\tau_k$. 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 $\mu_k$. Moreover, the cost in the objective function for this assignment is $c_{ik} = \min_{i \in i} \{w_{ik} \tau_k\}$ for $i \in \mathbb{N}$, $k \in S_i$, where $w_{ik}$, for $k \in S_i$, $i \in \mathbb{N}$, 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 $J_k = \{(i, \rho) : i \in \mathbb{N}, \rho \in S_i, \text{and} \quad [\tau_k - 1, \tau_k] \subseteq [\tau_k - \Delta_{i,\mu_k}, \tau_k]\}$. Note that for each $k \in K$, $J_k$ is the set of combinations $(i, \rho)$ such that slot $k$ will be occupied if $x_{ik} = 1$. Then, problem EGP can be formulated as follows.

\[
\text{EGP:} \quad \text{Minimize} \quad \sum_{i \in \mathbb{N}} \sum_{k \in S_i} c_{ik} x_{ik} \\
\text{subject to} \quad \sum_{k \in S_i} x_{ik} = 1 \quad \text{for} \quad i \in \mathbb{N}
\]
\[
\sum_{(i, p) \in J_k} x_{i p} \leq 1 \quad \text{for } k \in K
\]

\[
x_{i k} \in \{0, 1\} \quad \text{for } k \in S_i, \ i \in N
\]

**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 \( j \in \mathbf{M} \). 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


E. Balas, "Facets of the knapsack polytope,” Mathematical Programming 8 (1975), 146-164.


E. Balas and S.M. Ng, “On the set covering polytope: II. lifting the facets with coefficients in {0,1,2},” Mathematical Programming 45 (1999), 1-29.


E.S. Gottlieb and M.R. Rao, "(1,k)-configuration facets for the generalized assignment problem," *Mathematical Programming* 46 (1990), 53-60.

References


M. Padberg, "The boolean quadric polytope; some characteristics, facets and relatives," *Mathematical Programming* 45 (1989), 139-172.


L.A. Wolsey, "Valid inequalities for 0-1 knapsacks and MIPs with generalized upper bound constraints," *Discrete Applied Mathematics* 29 (1990), 251-261.


Vita

Youngho Lee, born on August 30, 1980 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 Blue Ridge Mountains while in Blacksburg, Virginia.

Youngho Lee