THE THEORY OF SIMULTANEOUS LIFTING: CONSTELLATIONS IN CONFLICT HYPERGRAPHS

by

SAMIR PAHWA

B.S., Kansas State University, 2009

A THESIS

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial and Manufacturing Systems Engineering

College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2009

Approved by:

Major Professor

Dr. Todd Easton

ABSTRACT

Integer programming (IP) is a powerful technique used by many companies and organizations to determine optimal strategies for making decisions and managing resources to achieve their goals. One class of IP problems is the multiple knapsack (MK) problem. However, MK and other IP problems, are extremely complicated since they are $\mathcal{NP}$-hard problems. Furthermore, there exist numerous instances that can not be solved.

One technique commonly used to reduce the solution time for IP problems is lifting. This method, introduced by Gomory, takes an existing valid inequality and strengthens it. Lifting has the potential to form facet defining inequalities, which are the strongest inequalities to solve an IP problem. As a result, lifting is frequently used in integer programming applications.

This research takes a broad approach to simultaneous lifting and provides its theoretical background for. The underlying hypergraphic structure for simultaneous lifting in an MK problem is identified and called a constellation. A constellation contains two hypercliques and multiple hyperstars from various conflict hypergraphs. Theoretical results demonstrate that a constellation induces valid inequalities that could be obtained by simultaneous lifting. Moreover, these constellation inequalities can be facet defining.

The primary advancements, constellations and the associated valid inequalities, of this thesis are theoretical in nature. By providing the theory behind simultaneous lifting, researchers should be able to apply this knowledge to develop new algorithms that enable simultaneous lifting to be performed faster and over more complex integer programs.

Dedicated to my Mom, Dad, Mrinal, Manika,

and the rest of my family and friends.

Contents

List of Tables **vi**

1 Introduction **1**

1.1 Integer Programming 3

1.2 Research Motivations 6

1.3 Research Contributions 6

1.4 Outline 7

2 Background Information **9**

2.1 Integer Programming 9

2.2 Polyhedral Theory 12

2.3 Knapsack Problem 15

2.4 Covers 17

2.5 Lifting 20

2.5.1 Sequential Lifting 22

2.5.2 Simultaneous Lifting 25

2.6 Graph Theory 29

2.7 Hypergraph Theory 32

2.8 Conflict Graphs and Integer Programming 35

2.9 Conflict Hypergraphs and Integer Programming 38

3 Simulataneous Uplifting with Constellations 41

3.1 Constellation Background 42

3.2 Constellation in a Multiple Knapsack 52

3.3 Constellation Summary 57

3.4 Constellation in a Knapsack 61

4 Conclusion 68

4.1 Future Work 69

List of Tables

2.1 Associated Weight and Benefit 16

Chapter 1

Introduction

Integer programming (IP) is a powerful technique used by many companies across many different industries to improve their operations and functionality. Integer programming can be used to accomplish corporate goals such as maximizing profit, minimizing cost, or managing investment portfolios. It can also be used by large organizations or institutions for scheduling purposes, logistics and routing, and to better manage resources.

As an example, AT&T saved $750 million annually by using integer programming to implement a new system that would better design call centers for various business customers [19]. Similarly IBM applied integer programming concepts to improve their network of spare parts and accessibility to them, and increased revenues by $20 million and saved another $250 million in inventory costs per year [28]. Integer programming techniques have also been applied in other areas of business operations such as manufacturing [1, 40, 54], transportation [20, 21, 82, 88], and energy [66, 72, 87].

Solving integer programs has resulted in solutions to a broader array of complex problems. However, these problems are extremely complicated since they can take an exponential amount of time to complete, and some have yet to be solved. Research in integer programming focuses on finding faster techniques to solve integer programs. While recent advancements in technology and solution methodologies have allowed more complex problems to be solved [15], generating novel integer programming methods can still improve the solution time required to solve integer programs and is an active research area [53].

This thesis focuses on a particular class of integer programs called the multiple knapsack problem (MK). The name knapsack comes from an analogy of a camper's problem of packing a knapsack. In this case, the camper wants to take certain objects for his or her trip, but at the same time he or she can only carry a maximum weight. Each of the objects has an associated benefit and weight. Therefore, the camper seeks to maximize his or her benefit while not exceeding the maximum weight limitation.

In some similar problems, there is more than one knapsack type constraints, and these are referred to as multiple knapsack problems. For instance, these additional constraints can represent volume, budget, etc. Each object still has an associated benefit, but now the camper seeks to maximize benefit without exceeding any of the constraints in the multiple knapsack problem.

A typical application of a knapsack problem is capital budgeting. The selection of investments is similar to the selection of objects in the aforementioned trip. Therefore, a capital budgeting problem can be modeled as a knapsack problem, and the solution provides the optimal investment strategy [39, 64, 69]. Other applications of knapsack problems include treating cancer with radiotherapy [23], improving networks for mobile phones [60], and studying ant colony optimization [91].

1.1 Integer Programming

Integer programs are similar to linear programs except that an integer condition is placed on all of the variables. Integer programs are optimization problems as they try to determine the best possible solution subject to certain constraints. There is an objective function that is to be maximized or minimized and there are constraints that restrict the feasible solutions. This being the case, the challenge is to determine a solution that is not only feasible, but optimal. Unfortunately IP problems are classified as $\mathcal{NP}$-hard [55] because they can not be solved in polynomial time unless $\mathcal{P} = \mathcal{NP}$. Thus, researchers consistently seek to develop faster methods to solve integer programs.

The simplest method to solve an integer program is to enumerate every possible solution and then choose the solution with the best objective value. Although the technique is straightforward, it is hardly practical as the number of different solutions exponentially escalates. Consider an integer program with 100 binary decision variables. The number of possible solutions is $2^{100} = 1.27 * 10^{30}$. If the number of decision variables increases by one, the number of solutions doubles. Therefore, the number of solutions grows exponentially, and evaluating every solution is virtually impossible. Several other techniques have been developed to solve integer programs. Of these techniques, two of the most widely used are branch and bound, and cutting planes.

Branch and bound is a general algorithm to solve integer programs. The method was first introduced by A. H. Land and A. G. Doig in 1960 [61]. Branch and bound aims to solve integer programs by taking a systematic approach to enumerating possible solutions. It starts by solving a linear relaxation, which is the integer program minus the integer restriction. If the optimal solution to the linear program does not contain all integer values, then one of the variables with a fractional value is selected as the branching variable. From this parent node, two separate child nodes are formed. The first node has the same linear relaxation of the parent node with the additional constraint that the branching variable must be greater than or equal to the rounded up value of the branching variable. The other node also has the existing linear relaxation of the parent node, but this time the constraint is less than or equal to the rounded down value of the branching variable. This process continues to branch until all nodes are fathomed. A node is fathomed for one of three reasons; if the solution of the node has all integer values, if the solution of the node is infeasible, or if the objective value of the solution is worse than the objective value of the best found integer solution.

The effectiveness of branch and bound is contingent upon the branching criteria. Breadth first search, which searches one level of the tree at a time, can be inefficient since it necessitates solving a large number of linear programs before any new integer solution can be gained. On the other hand, depth first search, which searches one particular branch of the tree, can be ineffective if it happens to be searching a part of the solution space that does not have a feasible solution. Each strategy has the potential to be good or bad given a certain problem. Thus, there is no clear notion of which search technique is superior and it depends entirely on the problem.

Gomory introduced the idea of cutting planes, which aim to generate valid inequalities [41, 42, 43]. These inequalities, or cuts as they are often referred to, cut off some part of the linear relaxation while ensuring that any feasible solution is not eliminated. For an inequality to be valid, every feasible integer point must satisfy the inequality. The easiest way to determine if an inequality is valid is to substitute it as the objective function and solve the integer program. Any cut generated from this method is inserted into the original integer program as a new constraint and could eliminate portions of the linear relaxation. The strongest cutting planes are those inequalities that are facet defining.

Lifting is a frequently used method to generate cutting planes. Because of the usefulness of lifting, much research has been done on it over the years. The primary objective of lifting is to strengthen valid inequalities, which is achieved by appending more variables to the existing inequality and/or modifying the coefficients that are already included in the inequality. On some occasions the new lifted inequality can be facet defining. As a result, the lifted inequality can assume the role of a cutting plane and potentially expedite the time required to solve an integer program.

1.2 Research Motivations

In 1978 Zemel [89] introduced simultaneous lifting as a method to lift sets of integer variables. The technique was limited to only solve problems with binary variables and the algorithm required solving exponentially many integer programs to be successful. At Kansas State University, Dr. Easton has continued research in topics that use simultaneous lifting. Recently one of his students, Talia Gutierrez, developed a new technique to perform simultaneous uplifting [49]. This method requires the solution to an integer program, which can be too computationally challenging to be an effective tool.

The motivation for this research is to provide the theoretical foundations of simultaneous lifting. Since hypergraphic structures have been used to identify various valid inequalities, it seems promising that a new hypergraphic substructure can be identified that allows for simultaneous lifting. With this structure researchers may be able to find faster techniques to perform simultaneous lifting, which would make it a computationally attractive tool.

1.3 Research Contributions

By combining number theory, polyhedral theory, graph theory, and integer programming principles, this thesis provides new theory regarding simultaneous lifting in a multiple knapsack problem. The major breakthrough is the discovery of a constellation structure that exists when there are two hypercliques. This is achieved by identifying a collection of hyperstars that exist between the hypercliques. The result is the constellation structure and more importantly, two constellation inequalities that are valid.

Constellation inequalities are beneficial since they are potentially facet defining. Therefore, determining simultaneously uplifted inequalities from a hypergraphic structure in a multiple knapsack problem can assist in solving integer programs. In a larger sense, a constellation structure provides a better conceptual understanding for simultaneous lifting. By analyzing the structure it is easier to find valid inequalities. All of this research adds to the general body of integer programming knowledge.

Ultimately, this research provides a theoretical understanding of the relationship between simultaneous lifting and hypergraphic structures in integer programming, and specifically in the multiple knapsack problem. By providing the theory behind simultaneous lifting, researchers should be able to apply this knowledge to develop new algorithms that enable simultaneous lifting to be performed faster and over more complex integer programs.

1.4 Outline

A detailed introduction to integer programming and the fundamental ideas it is predicated on are discussed in Chapter 2. Topics such as polyhedral theory, the knapsack problem, covers and cover inequalities, and general lifting techniques are presented. In addition, concepts and definitions from graph and hypergraph theory are explained.

Chapter 3 presents the exact simultaneous lifting hypergraphic method discussed in this thesis. An in depth explanation is given to show how to determine the new simultaneous lifting inequalities and the hypergraphic structures that correspond to them. An example of a constellation is also given to describe the process.

Chapter 4 contains any conclusions drawn from this research. The development of a constellation structure and inequality using simultaneous lifting, as achieved through this research, has many different possibilities for potential exploration and few of these ideas are briefly discussed.

Chapter 2

Background Information

To better understand the purpose of this research, it is necessary to first become familiar with the background knowledge and fundamentals of integer programming. Integer programming is a field within operations research, yet there are several other aspects of integer programming such as polyhedral theory, lifting techniques, and graph theory that have had a significant hand in IP research. This chapter focuses on all of these topics and supports the basic concepts of integer programming that are required to understand this thesis.

2.1 Integer Programming

An integer program (IP) is defined as maximize $c^T x$, subject to $Ax \leq b$, $x \in \mathbf{Z}^n_+$ where $A \in \mathbf{R}^{mxn}$, $C \in \mathbf{R}^n$ and $b \in \mathbf{R}^m$. The feasible region is the collection of integer solutions that satisfy the constraints of the IP, where $P = \{x \in \mathbf{Z}^n_+ | Ax \leq b\}$. Denote $N = \{1, 2, 3, ..., n\}$ as the indices of the variables.

IPs are usually solved by incorporating the solution of a linear program as a subroutine. Define the linear relaxation of an integer program to be the integer program without the integer constraint. Thus, define IP_{LR} to be maximize $c^T x$, subject to $Ax \leq b$, $x \in \mathbf{R}^n_+$. The feasible region of IP_{LR} is $P_{LR} = \{x \in \mathbf{R}^n_+ : Ax \leq b\}$.

A commonly used method to solve IPs is branch and bound. As mentioned earlier, the primary drawback of branch and bound is that it may require exponential time and memory to solve the IP. Additionally, the IP may spend considerable time searching through a portion of the solution space that may not contain any feasible points. Thus, an alternate strategy to solving integer programs uses valid inequalities. Valid inequalities were first used by Ralph Gomory [41, 42, 43]. A valid inequality takes the form of $\sum_{i=1}^{n} \alpha_i x_i \leq b$ and all $x \in P$ must satisfy this inequality. Several synonyms exist for valid inequalities and include cuts or cutting planes. There are two main criteria of any cut implemented for it to be valid and useful. Firstly, a cut must not eliminate a feasible integer solution, by definition. Secondly, the purpose of a cutting plane is to eliminate part of the feasible region of the linear relaxation. Each time a cut is generated, it is added as a constraint to the original IP. The following example depicts this idea.

Example 2.1.1 Consider the following IP.

Maximize

$8x_1 + 6x_2$

Subject to

$4x_1 + 4x_2 \leq 5$

$x_i \geq 0$ and integer $\forall\ i = 1, 2$

Before solving the IP, the first step is to solve the linear relaxation of this problem. This is done by solving the following linear program.

Maximize

$8x_1 + 6x_2$

Subject to

$4x_1 + 4x_2 \leq 5$

$x_i \geq 0\ \forall\ i = 1, 2$

The linear relaxation solution to the problem is $(1\frac{1}{4}, 0)$ with the objective value $z^* = 10$. However, if we introduce a valid inequality, $x_1 + x_2 \leq 1$, the optimal solution to the linear relaxation is eliminated. Now consider the same linear program with this added valid inequality.

Maximize

$8x_1 + 6x_2$

Subject to

$4x_1 + 4x_2 \leq 5$

$x_1 + x_2 \leq 1$

$x_i \geq 0 \ \forall \ i = 1, 2$

The solution to the linear relaxation is now $(1, 0)$ and $z = 8$. Notice the solution contains integer values so this is the optimal solution to the integer program.

2.2 Polyhedral Theory

Polyhedral theory describes the fundamental principles on which many problems are based. It provides the basic knowledge for IP problems and deals with polyhedra, which are the feasible sets for linear programming problems.

First a few definitions are necessary. A set, S, is a convex set if, and only if, for any two points p and q in S, $\lambda p + (1 - \lambda)q \in S$ for each $\lambda \in [0, 1]$. It can be observed that $\lambda p + (1 - \lambda)q$ signifies a point on the line segment spanning from p to q. In other words, a point of the form $\lambda p + (1 - \lambda)q$ where $0 \leq \lambda \leq 1$ is the weighted average of p and q. Therefore, S is called a convex set if for any two points p, q in S there is a straight line that connects them, and the line is within the set S in its entirety. Define the convex hull of a set S to be the intersection of all convex sets that contain S, which is denoted by S^{conv}. Convexity is a pillar of many optimization problems.

Halfspaces are the solution space for a single linear inequality. For example, all $x \in \mathbf{R^n}$ such that $\sum_{i=1}^{n} \alpha_i x_i \leq \beta$ is a halfspace. The intersection of finite half spaces forms a polyhedron. Clearly, the feasible region of a linear program is a polyhedron. A polyhedron is trivially convex.

When trying to solve an IP, the linear relaxation is typically solved first. The linear relaxation has the corresponding polyhedron, P_{LR}, which contains both integer and non-integer points. Observe that P is neither a polyhedron nor convex. However, it is the intersection of all convex sets of P that is of interest. Determining P^{conv} is a primary part of integer programming research. If a linear relaxation is solved over P^{conv}, then an optimal solution is integer and branch and bound can be avoided. Thus, polyhedral theory in integer programming seeks to transform the feasible region of the linear relaxation to be P^{conv} by adding additional constraints, valid inequalities or cuts.

A significant portion of polyhedral theory deals with the dimension of a polyhedron and the face of a valid inequality. The dimension of a polyhedron is the number of linearly independent vectors contained in the polyhedron. However, the feasible region of an IP is a collection of points so affine independence is used rather than linear independence.

The collection of points $x_1, x_2, x_3, ..., x_r \in \mathbf{R}^n_+$ are affinely independent if and only if $\sum_{i=1}^{r} \lambda_i x_i = 0$ and $\sum_{i=1}^{r} \lambda_i = 0$ is uniquely solved by $\lambda_i = 0$ for all $i = 1, ..., r$. The number of affinely independent points is one more than the number of linearly independent vectors for a particular convex set. Furthermore, the dimension of a convex set can be stated as the maximum number of affinely independent points minus one. This also suggests that an empty set has a dimension of -1.

A critical component of integer programming polyhedral theory is the idea of a valid inequality or cutting plane. A valid inequality is any inequality that does not eliminate a feasible solution. Thus, $\alpha^T x \leq \beta$ is valid for P^{conv} if, and only if, every $x' \in P$ satisfies $\alpha^T x' \leq \beta$.

Every valid inequality defines a face of P^{conv}. Let $\alpha^T x \leq \beta$ be a valid inequality of P^{conv}, then its corresponding face is the set of points in the polyhedron that meets this inequality at equality. Formally, the face defined by $\alpha^T x \leq \beta$ is $\{x \in P^{conv} : \alpha^T x = \beta\}$.

There are an infinite number of inequalities that could induce the faces of a polyhedron, but the most important faces are contingent upon the dimension. Those valid inequalities that induce a face with dimension of exactly one less than the dimension of P^{conv} are categorized as facet defining inequalities. Defining at least one valid inequality for each and every facet is sufficient to describe P^{conv}. This concept is a fundamental aspect of research in integer programming [4, 7, 9, 10, 26, 32, 34, 45, 50, 57, 65]. Furthermore, finding new classes of facet defining inequalities should remain an essential part of IP research for years to come.

2.3 Knapsack Problem

Within the realm of integer programming, a special type of IP is the Knapsack Problem (KP). The term knapsack is used because this kind of problem is analogous to packing a bag with n items, where each item has a certain benefit, c_i, and mass, a_i. Regardless of the benefit of the items, the total weight in the bag must be less than the specified limit the person can carry, which in this case is b. The formulation for a knapsack problem is maximize $\sum_{i=1}^{n} c_i x_i$ and subject to $\sum_{i=1}^{n} a_i x_i \leq b$ and $x_i = \{0,1\}$ $\forall i = 1, ..., n$.

Notice that a knapsack polyhedron is independent of the objective function. Rather a knapsack polyhedron only considers finding valid solutions and therefore only requires satisfying the constraints. Therefore, an objective function will only be given in examples in which it is specific to the problem. Just as IP problems are $\mathcal{NP}$-hard, KP problems follow suit. Thus, it is useful to find new methods to efficiently solve KP problems. Let the feasible region of a KP be denoted by $P_{KP} = \{x \in \mathbb{B}^n : \sum_{i=1}^{n} a_i x_i \leq b\}$ and $P_{KP}^{conv} = conv(P_{KP})$.

Without losing generality, assume that the variables a_i are sorted in descending order. Thus $a_i \geq a_j$ where $i < j$ and $i, j \in N$. Assume that $a_i \geq 0$ $\forall$ $i \in N$. However, if any $a_i < 0$, then x_i is replaced with $x_i' = 1 - x_i$ and it is equivalent to $a_i > 0$. Moreover, if $a_i \geq b$, then $x_i = 0$ for all feasible solutions and x_i can be eliminated from the problem. With these assumptions stated it can be seen that P_{KP}^{conv} is full dimensional with the affinely independent points of 0 and e_i $\forall$ $i = 1, 2, ..., n$ where e_i is the i^{th} identity point.

Objects	1	2	3	4	5	6	7	8
Benefit	100	45	60	49	36	9	14	3
Weight	20	18	15	14	12	9	7	6

Table 2.1: Associated Weight and Benefit

Example 2.3.1 Consider the following knapsack problem. Using the analogy of the camper, in this problem there are a total of 8 objects. The associated benefits and the weight for each object is given in the Table 1. Additionally, the camper is constrained by a maximum weight of 53. The main objective of the problem is to maximize the benefit while not violating the weight constraint. The corresponding IP is as follows.

Maximize

$$100x_1 + 45x_2 + 60x_3 + 49x_4 + 36x_5 + 9x_6 + 14x_7 + 3x_8$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 6x_8 \leq 53$$

$$x_i = \{0, 1\} \ \forall \ i = 1, 2, ..., 8.$$

The solution to the problem is $(1, 0, 1, 1, 0, 0, 0, 0)$ with the total benefit equal to 209. Thus, the camper should pack objects $1, 3$ and 4, which have an overall weight = 20 + 15 + 14 = 49 lbs. The benefit = 100 + 60 + 49 = 209.

A knapsack constraint has much importance since it is closely related to IP problems. This is because any binary IP constraint can be modified to fit the stipulations of a KP. If the IP constraint is of type '=', then it is transformed into two separate valid inequalities, one with '$\leq$' and the other with '$\geq$'. A greater than or equal to constraint can be multiplied by -1 to become less than or equal to constraints. As mentioned previously, if any $a_i < 0$, then x_i is replaced with $x_i' = 1 - x_i$. On the other hand, if $a_i > 0$, then that variable can be dropped from the problem. Therefore, a cutting plane of a KP is applicable to a binary IP constraint. Determining robust cutting planes for a KP plays a significant role in integer programming research [8, 75, 63].

There are other forms of a knapsack problem. The most relevant to this research is the multiple knapsack problem. A Multiple Knapsack (MK) problem has multiple knapsack constraints which may be volume, budget, or resources when considering the knapsack analogy. The formulation for a multiple knapsack problem is maximize $\sum_{i=1}^{n} c_i x_i$ and subject to $\sum_{i=1}^{n} a_{ji} x_i \leq b_j$ for $j = 1, ..., m$ and $x_i = \{0, 1\}$ for $i = 1, ..., n$. In this case, let the feasible region of a MK be denoted by $P_{MK} = \{x \in \mathbb{B}^n : \sum_{i=1}^{n} a_{ji} x_i \leq b_j\}$ for $j = 1, ..., m$ and $P_{MK}^{conv} = conv(P_{MK})$.

2.4 Covers

For a knapsack problem one of the most common cutting planes is called a cover inequality. A cover is a set of indices that when the variables are set to one violates the right hand side of the constraint. Formally, a cover, C, is defined as a set of indices where $\Sigma_{i \in C}\ a_i > b$.

When forming a cover, it is important to select the most beneficial covers. One type of cover is a minimal cover. Minimal covers are covers in which the removal of any one index from the set causes the set not be a cover. Formally, a cover is minimal if $\Sigma_{i \epsilon C - \{j\}}\ a_i \leq b$ for each $j \in C$. Minimal covers have a dimension of at least $|C| - 1$ in

P_{KP}^{conv}, which is why they are more useful than other covers.

Reconsider the constraint of the knapsack problem $20x_1+18x_2+15x_3+14x_4+12x_5+9x_6+7x_7+6x_8 \leq 53$. From this constraint, a cover C can be $\{1,2,3,5,6,8\}$ because the sum of the coefficients $20+18+15+12+9+6=80>53$. Another cover could be $\{3,4,5,6,7\}$ as the sum of these variables is $57>53$. Covers can include any number or combination of variables. As a result, there can be exponentially many covers for a given problem. A cover takes the form, $\sum_{i\in C} x_i \leq |C|-1$, which is a valid inequality and it is called a cover inequality.

Using the same constraint from above, the cover with indices $\{2,3,4,5\}$ exists because the sum of the coefficients $18+15+14+12=59$, which is greater than the right hand side value of 53. The valid cover inequality is $\mathrm{x}_2+x_3+x_4+x_5 \leq 3$. This is also a minimal cover since the removal of any element from the cover results in the sum of the coefficients being less than 53.

A way to strengthen a cover inequality is to find an extended cover. A set $E(C)=C\cup\{j\in N-C : a_j > a_i \; \forall \; i\in C\}$ is called an extended cover of C. $E(C)$ can be used to generate an extended cover inequality, which takes the form, $\sum_{i\in E(C)} x_i \leq |C|-1$. Extended covers are valid inequalities and the following theorem shows the necessary and sufficient conditions for an extended cover inequality $E(C)$ to be facet defining [50].

Theorem 2.4.1. *Let $C=\{i_1,i_2,...,i_r\}$ be a minimum cover with $i_1<i_2<...<i_r$. If any of the following conditions hold, then the extended cover inequality is a facet defining inequality of P_{KP}^{conv}.*

a. $C = N$.

b. $E(C) = N$ *and (i)* $(C \backslash \{i_1, i_2\}) \cup \{1\}$ *is not a cover.*

c. $C = E(C)$ *and (ii)* $(C \backslash \{i_1\}) \cup \{p\}$ *is not a cover, where* $p = min\{i : i \in N \backslash E(C)\}$.

d. $C \subset E(C) \subset N$ *and (i) and (ii).*

Thus we can extend the cover C to $E(C)$ by appending 1, which results in $E(C) = \{1, 2, 3, 4, 5\}$. The extended cover inequality is $x_1 + x_2 + x_3 + x_4 + x_5 \leq 3$. This inequality satisfies Theorem 2.4.1 and is facet defining. The eight affinely independent points that help prove this are shown in the following matrix.

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Extended covers can be very helpful, but sometimes these inequalities are not facet defining. Lifting improves the dimension of an inequality. There are several ways to do lifting and these techniques are discussed in the next section.

2.5 Lifting

Determining methods to generate cutting planes is a staple of integer programming research. One way to create cutting planes is a technique called lifting. The purpose of lifting is to strengthen an existing valid inequality by changing some of the coefficients in the inequality. Ralph Gomory [44] was the first to implement lifting and further advancements in the technique were made in [4, 10, 11, 12, 22, 26, 29, 32, 34, 37, 46, 47, 49, 57, 65, 68, 74, 90]. There are at least three broad categories of lifting; up vs down, exact vs approximate, and sequential vs simultaneous. Given the three categories and the two options for each lifting attribute, there are a total of 8 (2^3) different ways to conduct lifting.

An important component of lifting is the concept of a restricted space. Let the restricted space of P^{conv} on the set of $E \subseteq N$ be defined as $P^{conv}_{E,K} = conv\{x \in P : x_j = k_j \text{ for all } j \in E\}$ where $K = (k_1, k_2, ..., k_{|E|}) \in \mathbf{Z}^{|E|}$. Instead of observing the polyhedron in entirety, only a subset of variables is considered in a restricted space. This implies that $x_j = k_j$ for all $j \in E$. In other words all variables with indices in E are fixed to certain values.

The general lifting procedure starts with an inequality $\sum_{i \in N \setminus E} \alpha_i x_i + \sum_{i \in E} \alpha_i x_i \leq \beta$ that is valid over $P^{conv}_{E,K}$. The lifting procedure seeks a valid inequality of the form $\sum_{i \in N \setminus E} \alpha_i x_i + \sum_{i \in E} \alpha'_i x_i \leq \beta'$ that is valid over P^{conv}. These different versions of lifting are determined by the size of the set E, the values of K and the values of α' and β'.

Exact lifting stipulates that the coefficients be calculated with complete accuracy.

Thus, exact lifting should increase the dimension of the inequality as there must exist a point not in the restricted space that meets the exact lifted inequality it at equality [10, 36, 78]. Since exact lifting typically requires solving an integer program, researchers have sacrificed the accuracy of the lifting coefficient for a faster time to obtain the coefficient. This is called approximate lifting and has been used by [76, 85].

Sequential lifting changes the coefficients for one variable at a time so $|E| = 1$. Simultaneous lifting modifies the coefficient of a group of variables at the same time and $|E| \geq 2$. In recent times, a substantial amount of research has recently been done on efficient methods to perform simultaneous lifting[5, 48, 49, 79, 86].

Uplifting assumes that there is a valid inequality of $P^{conv}_{E,K}$ where $K = \{0, 0, ..., 0\}$. Uplifting does not change the right hand side of the valid inequality and seeks to increase the coefficients associated with variables in E. Since this thesis primarily focuses on uplifting, any $P^{conv}_{E,K}$ with $K = \{0, 0, ..., 0\}$ is denoted as P^{conv}_{E}.

Conversely, down lifting assumes a valid inequality of $P^{conv}_{E,K}$ where $K = \{u_{j_1}, u_{j_e}, ..., u_{j_{|E|}}\}$ where u_j is the upper bound for variable j. Down lifting typically decreases the values of the right hand side of the valid inequality and also the coefficients for the variables in E. There is also a middle lifting, which is roughly a combination of both up and down lifting [84].

This research develops theoretical results for exact simultaneous uplifting.

2.5.1 Sequential Lifting

The most widely used lifting method is sequential uplifting [7, 9, 50, 67, 83, 84]. Sequential uplifting a binary variable begins by formulating an IP in which the valid inequality is the objective function, while the constraints are kept the same as the original. Then the variable that is to be lifted is set to 1 so another constraint is inserted to represent this. Next the IP is solved and the objective value, Z^*, is computed. To determine the lifting coefficient, $\alpha = \beta - Z^*$. Each time a different variable is lifted, a constraint is substituted to set that variable to 1 and the objective function is updated. With these changes the IP is then resolved. The new objective value is obtained and then α is calculated. The process repeats for each variable that is to be lifted. The order of lifting is important as different orders result in different lifting coefficients. The following example explains this.

Example 2.5.1 The next example uses the previous knapsack problem and has $C = \{1, 2, 3, 4\}$. The cover inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$. The variables to be sequentially uplifted are x_5, x_6, x_7 and x_8. Sequentially lifting x_5 requires solving the following integer program:

Maximize

$$x_1 + x_2 + x_3 + x_4$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$$

$$x_5 = 1$$

$$x_i = \{0, 1\} \ \forall \ i = 1, 2, ..., 8.$$

The solution is $Z^* = 2$ with $(0, 0, 1, 1, 1, 0, 0, 0)$. The coefficient for lifting x_5 is $\alpha_5 = 3 - 2 = 1$. The new uplifted inequality is now $x_1 + x_2 + x_3 + x_4 + x_5 \leq 3$. Next x_6 is lifted by solving the following problem.

Maximize

$$x_1 + x_2 + x_3 + x_4 + x_5$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$$

$$x_6 = 1$$

$$x_i = \{0, 1\} \ \forall \ i = 1, 2, ..., 8.$$

The solution is $Z^* = 3$ with $(0, 0, 1, 1, 1, 1, 0, 0)$. The coefficient for lifting x_6 is $\alpha_6 = 3 - 3 = 0$. The inequality is now $x_1 + x_2 + x_3 + x_4 + x_5 + 0x_6 \leq 3$. Since x_6 was uplifted into the inequality with a zero coefficient, and both x_7 and x_8 have smaller coefficients than x_6, it can be concluded that Z^* will be 3 or higher. As a result, α_7 and α_8 are 0. Thus, the sequentially lifted inequality is finished and is $x_1 + x_2 + x_3 + x_4 + x_5 + 0x_6 + 0x_7 + 0x_8 \leq 3$.

This inequality is facet defining.

As stated earlier, sequential lifting involves changing one variable at a time, each with its own coefficient, in the overall inequality. This requires resolving an optimization problem each time a new variable is to be lifted. It is also important to note that the order the variables are lifted can vary. As a result, there are $(n - |C|)!$ different ways to perform sequential lifting in this problem. Similarly, as the lifting order varies, the coefficients of the lifted variables may possibly change.

In the previous example x_5 was the first variable to be lifted, but it is possible to start with any of the variables from x_5 to x_8. In this problem any variable from x_5 to x_7 chosen to be lifted first would have an $\alpha = 1$, while the the other two variables would have $\alpha = 0$. No matter the order x_8 is lifted with a coefficient of 0. Performing each of the lifting sequences leads to the group of valid and facet defining inequalities listed below:

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3$$

$$x_1 + x_2 + x_3 + x_4 + +x_6 \leq 3$$

$$x_1 + x_2 + x_3 + x_4 + +x_7 \leq 3$$

Averaging these inequalities together results in the following inequality: $x_1 + x_2 + x_3 + x_4 + \frac{1}{3}(x_5 + x_6 + x_7) \leq 3$. Since x_5 to x_7 are being uplifted with the same value this is an example of simultaneous lifting, which is discussed in the next section. Furthermore, this $\frac{1}{3}$ coefficient can be strengthened so this is an example of approximate simultaneous uplifting.

2.5.2 Simultaneous Lifting

The advent of simultaneous lifting is another approach to generate cutting planes. This method originated in 1978 by Zemel [89]. However, Zemel's method could only lift integer programs with binary variables and still required solving exponentially many IPs. Clearly, this method is technically accurate, but it cannot be applied in a practical instance so it is more of a theoretical result.

Since the late 1990s research has continued in this area. The next development in simultaneous lifting was sequence independent lifting [5, 48, 79]. Sequence independent lifting is a technique that ignores the order in which the variables are lifted and does not require solving any integer programs. The basic idea is to develop a superadditive function for a cover inequality and then there exists a lower bound for every coefficient. Thus, all variables are lifted simultaneously based off of a simple expression. While sequence independent lifting is a faster technique to create cutting planes, it is only an approximate lifting technique. Therefore, the valid inequalities formed are not necessarily facet defining and could be strengthened.

Exact simultaneous lifting aims to work efficiently but not sacrifice any precision. In simultaneous lifting a set of variables is added to the inequality. The advantage of this method is that by lifting multiple variables at the same time it reduces the number of optimization problems that have to be solved. Furthermore, these inequalities tend to be stronger cuts. Once the lifted inequality is formed it can be used as a cutting plane and reduce the solution space of a integer program.

A substantial amount of work on simultaneous lifting has continued at Kansas State University under the guidance of Todd Easton. Easton and Hooker worked on the background concepts regarding simultaneous lifting research [52]. Ultimately, they presented a linear time algorithm to simultaneously lift a set of variables into a cover inequality for a binary knapsack problem.

In this case, let C represent a cover and $E \subseteq N \backslash C$. The variables in E are simultaneously lifted into the cover inequality, which takes the form of $\sum_{i \in C} x_i + \alpha \sum_{j \in E} x_j \leq |C| - 1$ where α is the coefficient of lifting. This algorithm takes $O(|C| + |E|)$ time to generate a valid inequality assuming C and E are sorted in descending order.

Later Sharma extended this idea by performing additional theoretical research and computational studies [78]. The advantage of Sharma's technique is that it assists in selecting which sets of variables to lift. The algorithm generates numerous inequalities and needs quadratic time to run. Sharma also showed impressive computational results.

While Sharma was developing her method, Gutierrez [49] developed a lifting technique that can exactly lift sets of bounded integer variables simultaneously by solving a single integer program. Gutierrez's algorithm begins by setting α high, such as $\alpha = M$. An integer program is solved where the objective is the left hand side of the proposed simultaneously lifted inequality with the specific α value. The constraints are the same as of the original IP. If $Z \leq \beta$, then the algorithm terminates and reports α as the lifting coefficient. If $Z > \beta$, then the x^* from the IP is used to solve for a new α and the process repeats. The following problem demonstrates Gutierrez's technique with the previous example.

Example 2.5.2 Reconsider the constraint $20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$. This example has the cover $C = \{1, 2, 3, 4\}$. The valid inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$. Since C is minimal, this inequality is facet defining over $P_{KP_C}^{conv}$. The variables to be simultaneously uplifted are x_5, x_6, and x_7. This inequality takes the form of $x_1 + x_2 + x_3 + x_4 + \alpha(x_5 + x_6 + x_7) \leq 3$.

Applying Gutierrez's method to this problem results in solving the following IP.

Maximize

$$x_1 + x_2 + x_3 + x_4 + x_5 + M(x_5 + x_6 + x_7)$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$$

$$x_i = \{0, 1\} \ \forall \ i = 1, 2, ..., 8.$$

The objective function has the value $Z^* = 3M+1$ with the solution of $(0, 0, 0, 1, 1, 1, 1, 0)$. Inserting the solution into the simultaneously lifted constraint set at equality results in $(0+0+0+1)+\alpha(1+1+1) = 3$. Solving for α leads to $\alpha = \frac{2}{3}$. Since $Z^* > \beta = 3$, the process repeats again with the proposed inequality, $x_1+x_2+x_3+x_4+\frac{2}{3}(x_5+x_6+x_7) \leq 3$. So the following IP is solved.

Maximize

$$x_1 + x_2 + x_3 + x_4 + x_5 + \frac{2}{3}(x_5 + x_6 + x_7)$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$$

$$x_i = \{0, 1\} \; \forall \; i = 1, 2, ..., 8.$$

The objective function has the value $Z^* = 3\frac{1}{3}$ with the solution of $(0, 0, 1, 1, 1, 1, 0, 0)$. Inserting this solution into the simultaneously lifted constraint set at equality results in $(0 + 0 + 1 + 1) + \alpha(1 + 1 + 0) = 3$. This results in $\alpha = \frac{1}{2}$. Since $Z^* > \beta = 3$, the process repeats with the proposed inequality $x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$. Next, solve

Maximize

$$x_1 + x_2 + x_3 + x_4 + x_5 + \frac{1}{2}(x_5 + x_6 + x_7)$$

Subject to

$$20x_1 + 18x_2 + 15x_3 + 14x_4 + 12x_5 + 9x_6 + 7x_7 + 3x_8 \leq 53$$

$$x_i = \{0, 1\} \; \forall \; i = 1, 2, ..., 8.$$

The objective function has the value $Z^* = 3$ with the solution of $(0, 1, 1, 1, 0, 0, 0, 0)$. Since $Z^* = \beta = 3$, the lifting procedure is completed and α remains at $\frac{1}{2}$. Thus, Gutierrez's simultaneous lifting algorithm generates the inequality $x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) \leq 3$. The simultaneously lifted inequality clearly dominates the average of the sequentially lifted inequality since the coefficient for lifting has increased from $\frac{1}{3}$ to $\frac{1}{2}$. This inequality is facet defining as shown by the affinely independent points given in the following matrix.

$$\begin{vmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

2.6 Graph Theory

From the outset of integer programming, graph theory has played a critical role in improving the solution times to integer programs. This thesis builds upon the previous knowledge by focusing on the graphical and hypergraphical structures necessary to perform exact simultaneous uplifting. Before these results can be discussed some background on graph theory is necessary.

Graph theory has been in various problems from all sorts of sciences. Graphs are uniquely helpful as they can visually represent large and complex problems that are otherwise based on intangible ideas. They have applications in wireless communication problems and mobile network optimization [30, 59, 81]. In addition, graph theory has a played a vital role in electrical engineering and the development of computer chips [24, 33, 38]. Some other fields that use graph theory are theoretical chemistry and genetics. In chemistry, scientists use graph theory to better understand molecular structures and bonds between atoms [6, 70, 80]. Similarly, graph theory is used for gene sequencing and understanding biological systems [51, 62, 73]. Thus, the understanding of graph theory is beneficial as it has a wide use of applications and it can contribute in many different areas of research.

A graph is defined by $G = (V, E)$ where $V(G) = \{1, ..., n\}$ is the set of vertices and $E(G) = \{d_1, ..., d_m\}$ is the set of edges where $d_i = \{u, v\}$ with $u, v \in V$. Given graphs $G = (V, E)$ and $G' = (V', E')$, if $V' \subset V$ and $E' \subset E$, then G' is a subgraph of G.

Graphs can have many different structures, each with unique properties and significance. Some common types of graph structures include a clique, a hole, a star, a wheel, and a tree, which are defined below.

An $v_1 - v_n$ path begins from vertex v_1 and follows edges until it reaches v_n. It is formally defined by $V = \{v_1, ..., v_n\}$ and $E = \{v_i, v_{i+1}\}$ for $i = 1, ..., n - 1$.

A cycle, L_n, is a graph that is a path with an additional edge between the starting and ending vertices. It is formally defined by $V = \{v_1, ..., v_n\}$ and $E = \{\{V_i, V_{i+1}\} : i = 1, ..., n - 1\} \cup \{n, 1\}$.

A clique, K_n, is a graph where every vertex is adjacent to every other vertex. Formally, a graph K_n is a clique of size n if and only if $V(K_n) = \{v_1, ..., v_n\}$ and $\{u, v\} \in E(K_n)$ for all $u, v \in V(K_n)$. In other words, a clique contains all possible $\begin{pmatrix} n \\ 2 \end{pmatrix}$

edges.

A star or fan, S_n, is a graph where one central vertex is adjacent to all other peripheral vertices but none of these peripheral vertices are adjacent to each other. It is formally defined by $V = \{v_1, ..., v_n\}$ and $E = \{\{V_1, V_i\} : i = 2, ..., n\}$.

A hole, Y_n, is a graph that is a cycle with no additional edges in the induced subgraph. It is formally defined by $V = \{v_1, ..., v_n\}$ and $E = \{\{i, i+1\} : i = 1, ..., n-1\} \cup \{\{n, 1\}\}$.

An antihole, A_n, is the compliment of a hole. Therefore $A_n = K_n \setminus E(Y_n)$.

A wheel, W_n, is a graph that can be seen as a single central vertex adjacent to all other peripheral vertices and the peripheral vertices form a cycle. Equivalently, a wheel can also be thought of as the combination of a star and hole together. It is formally defined by $V = \{v_1, ..., v_n\}$ and $E = \{(V_1, V_i) : i = 2, ..., n\} \cup \{v_i, v_{i+1} : i = 2, ..., n-1\} \cup \{v_n, v_2\}$.

A tree is a graph without any cycles. Trees play an important role when implementing branching techniques to solve integer programs. For more information regarding graph theory, refer to Diestel [35].

The basics of graph theory can be extended into hypergraph theory. Hypergraphs build upon graphs by allowing for more flexibility regarding the characteristics of the graph. Some graphical structures that are not possible to create with graphs can be made with hypergraphs, and this allows for further research and applications in various fields.

2.7 Hypergraph Theory

In basic graph theory a graph is simply depicted by a list of vertices and edges with 2 vertices in each edge. In hypergraph theory a hypergraph has edges that contain any number of vertices. Although hypergraphs are less common than regular graphs, they still have a significant impact on numerous applications and research endeavors. Hypergraph theory has been used in the research of coding theory and many other algorithms used in computer programming [13, 27, 77]. To better understand hypergraph theory, some basic concepts are explained in [14].

The extension of graph theory to hypergraph theory has been emphasized in this thesis, however there is some ambiguity of graph to hypergraph transitions that must be accepted. The reason is that graphs can be represented by a simple listing of the vertices and edges. Yet in the case of hypergraphs because a particular edge can encompass multiple vertices, the definition of hypergraph structures is open to interpretation.

Take a cycle and a hypercycle for example. In regular graphs a cycle can be six vertices $(1,2,3,4,5,6)$ and the six edges $\{\{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{5,6\}, \{6,1\}\}$ that form a closed loop. For the hypercycle, even an edge that consists of three vertices can form different hypergraphs. One hypercycle can have the edges shift by one vertex at a time to form a total of six edges, $\{\{1,2,3\}, \{2,3,4\}, \{3,4,5\}, \{4,5,6\}, \{5,6,1\}, \{6,1,2\}\}$. On the other hand, a stipulation can be that only the last vertex in the edge can overlap between edges. This will result in a total of three edges $\{\{1,2,3\}, \{3,4,5\}, \{5,6,1\}\}$. The following definitions play an integral part in setting a foundation for the new research on hypergraphs and constellations.

A hypergraph, $H = (V, E)$, is defined by a set of vertices, $V(H) = \{1, ..., n\}$ and a set of edges $E(H) = \{d_1, ..., d_m\}$ where $d_i \subseteq V(H)$ for all $i = 1, ..., m$. Unlike a graph where edges are forced to be two vertices, hypergraphs can have an arbitrary number of vertices in each edge. Thus, edges in a hypergraph can be any subset of vertices in the vertex set. Given two hypergraphs $H = (V, E)$ and $H' = (V', E')$, if $V' \subseteq V$ and $E' \subseteq E$, then H' is a subhypergraph of H.

Critical to this work is the definition of a uniform hypergraph. A k-uniform hypergraph, H_k, is a hypergraph where all edges have k vertices. A k-uniform hypergraph is denoted as H_k and defined as $H_k = (V, E)$ where all $d \in E$ satisfy $|d| = k$.

Similar to graphs, hypergraphs have several different structures that play a key role in integer programming research. These include a hyperclique, a hyperhole, a hyperfan, a hyperstar, a hyperwheel, and a hypertree which are defined below.

A hyperclique, $K_{n,k}$, where n is the number vertices and k is the edge size, is a hypergraph that contains all possible edges of size k. Formally, a k-uniform hypergraph H_k with n vertices is a hyperclique if, and only if, $E(K_{n,k}) = \{\{u_1, ..., u_k\} : u_1, ..., u_k \in V(H_k)\}$. This means that selecting any k vertices from the entire set of vertices is an edge. It is important to note that in a hyperclique all possible $\begin{pmatrix} n \\ k \end{pmatrix}$ edges exist.

A hyperfan, $F_{n,m,k}$, [52] is a k uniform hypergraph with n nodes that contains $m < n$ vertices, called C^M, in the middle or hub. The remaining $n - m$ vertices are called perimeter vertices and denoted by C^P. Each edge contains C^M and an additional $k - m$ nodes from C^P. Additionally, $e_i \cap e_j = C^M$ for each e_i, $e_j \in E(F_{n,m,k})$. Notice that each vertex in the perimeter is in exactly one edge.

A hyperstar, $S_{n,m,l,k}$, is a k uniform hypergraph with n nodes that contains $m < n$ vertices, called C^M, in the middle or hub. The remaining $n - m$ vertices are called perimeter vertices and denoted by C^P. The edge set consists of all possible combinations that include exactly l vertices from the middle and $k - l$ vertices from the perimeter. Thus $E(S_{n,m,l,k}) = \{d \subseteq V : |d| = k,\ |d \cap C^M| = l \text{ and } |d \cap C^P| = k - l\}$. There are clearly $\begin{pmatrix} m \\ l \end{pmatrix} * \begin{pmatrix} n-m \\ k-l \end{pmatrix}$ edges in a hyperstar.

Observe that this definition is a slight modification from the hyperstars used by Hooker and Easton [52]. Their hyperstars only had three parameters and assumed that each edge must contain every vertex in the middle. This definition is more generic and necessary for this research.

A hyperhole, $Y_{n,k}$ is a hypergraph with n vertices, $\{v_1, ..., v_n\}$ if there is a relabeling of the vertices such that $E(Y_{n,k}) = \cup_{i=1}^{n} \{v_i, v_{(i \ mod \ n)+1}, v_{((i+1) \ mod \ n)+1}, ..., v_{((i+k-2) \ mod \ n)+1}\}$. Thus, a hyperhole creates a cyclic type structure of the edges.

A hyperantihole, $A_{n,k}$, is the compliment of a hyperhole. Therefore $A_{n,k} = K_{n,k} \setminus E(Y_{n,k})$.

A hyperwheel is the union of a hyperfan and a hyperhole. Observe that the union of a hyperstar and a hyperhole with the same sized edges is a hyperclique.

A hypertree is an acyclic hypergraph; however the definition of a hypercycle is ambiguous. So a hypergraph is a hypertree if the following reduction graph is acyclic. Let an edge graph of a hypergraph be defined as follows. Each vertex of the graph corresponds to an edge of the hypergraph. If two edges of the hypergraph share a vertex, then the corresponding vertices in the graph have an edge between them.

2.8 Conflict Graphs and Integer Programming

Graphs are very helpful to integer programming research because they can be used to show relationships between variables. Once the graph is determined, different graphic structures create valid inequalities. In some cases, the structures can lead to facet defining inequalities, which is a primary goal of much of the research in integer programming. The definitions discussed above are important because they act as a liaison between integer programming and graph theory.

A graph of particular interest in this thesis is the conflict graph $G_c = (V_c, E_c)$ [3, 31, 56]. It is a graph that depicts the constraints of a binary integer program. In a conflict graph each vertex represents a variable, while an edge $\{u, v\}$ exists between two vertices if setting both variables equal to one is not a feasible point. Determining all of the edges in a conflict graph and combining them together forms a conflict graph. Thus, a conflict graph is used to visually show which sets of variables are infeasible. The following example demonstrates how a knapsack problem can be converted into a conflict graph.

Example 2.8.1 Consider the following example.

Maximize

$4x_1 + 1x_2 + 7x_3 + 5x_4 + 6x_5$

Subject to

$6x_1 + 6x_2 + 5x_3 + 3x_4 + 2x_5 \leq 7$

$x_i \in \{0, 1\}$

By looking at the constraint it is clear that setting $x_1 = 1$ and $x_2 = 1$ is not a feasible point as $6 + 6 = 12$, which is greater than 7. Thus $\{1, 2\} \in E_c$. Similarly, $x_2 = 1$ and $x_3 = 1$ is not feasible so $\{2, 3\} \in E_c$. The same process can be repeated for all combinations of variables $1, 2, 3, 4$ and 5. This results in $V(G_c) = \{1, ..., 5\}$ and $E(G_c) = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}\}$.

Observe that the vertices of each edge cannot both be set to one and be feasible. Thus, edge $\{i, j\}$ implies a valid inequality $x_i + x_j \leq 1$. For edge $\{1, 5\}$ the inequality is $x_1 + x_5 \leq 1$. In some instances, these inequalities can be combined to form a stronger inequality. A common example of this is the clique inequality [4].

In this example $\{1, 2, 3, 4\}$ is a clique. It is clear that a conflict graph is a set of indices for which at most only one variable is set to one. If more than one variable is set to one, then it is infeasible. The conflict graph has a clique $\{1, 2, 3, 4\}$ since $E(G_c)$ contains $\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$. These associated inequalities are listed below.

$x_1 + x_2 \leq 1$

$x_1 + x_3 \leq 1$

$x_1 + x_4 \leq 1$

$x_2 + x_3 \leq 1$

$x_2 + x_4 \leq 1$

$x_3 + x_4 \leq 1$

Notice that all six edges of a $\begin{pmatrix} 4 \\ 2 \end{pmatrix}$ clique are present. By combining the 6 inequalities, the valid inequality $x_1 + x_2 + x_3 + x_4 \leq 1$ can be formed. This is commonly known as a clique inequality. A clique inequality is important because of its facet defining characteristics.

In general, a maximal clique, K_n, is facet defining and has the following inequality of $\sum_{i \in K_n} x_i \leq 1$. Since every edge exists, no two variables can be set to one. Thus the sum of all variables has to be one or less. Therefore the clique inequality is valid.

A star is also a useful structure. In general, a star, S_n, has the inequality $(n-2)x_1 + \sum_{i=2}^{n-1} x_{i+1} \leq n - 2$. Every node has an edge with x_1, this suggests that at most x_1 is set to one, or all other variables can be set to one. Because the coefficient for x_1 is the same is the right hand side, the star inequality accommodates this and is valid. Stars are important because they set the foundation for hyperstars, which are used in this thesis. While stars alone are helpful in showing basic relationships among the indices, hyperstars are much more useful since they can be facet defining. As a result, the advancement from a star to a hyperstar is significant for the purpose of this research.

Many researchers have used conflict graphs to depict integer programming polytopes and to derive facet-defining inequalities. In some restrictive classes of integer programs, induced subgraphs such as odd holes and odd anti-holes have also been shown to induce valid inequalities, which are facet defining in some instances [2, 3, 16, 18, 25, 71]. An odd hole, Y_n, has the inequality $\sum_{i \in Y_n} x_i \leq (\lfloor \frac{n}{2} \rfloor)$. On the other hand, an odd anti-hole has the inequality $\sum_{i \in A_n} x_i \leq 3$. These structures are not necessarily used in this thesis, however the goal in research of conflict graphs is to assist in determining valid inequalities that are potentially facet defining.

2.9 Conflict Hypergraphs and Integer Programming

A conflict hypergraph, $H = (V, E)$, is similar to a conflict graph except that the edges include more than 2 vertices. Once again these edges correspond to infeasible sets of vertices based off the constraints of the given integer programming problem. The next example modifies the previous example to show how a conflict hypergraph can be generated. Notice that b is now 13 instead of 7.

Example 2.9.1 Consider the following example.

Maximize

$4x_1 + 1x_2 + 7x_3 + 5x_4 + 2x_5$

Subject to

$6x_1 + 6x_2 + 5x_3 + 3x_4 + x_5 \leq 13$

$x_i >= 0 \ \forall \ i = 1, 2, ..., 5.$

By looking at the constraint it is clear that setting $x_1 = 1$, $x_2 = 1$, and $x_3 = 1$ is not a feasible point as $6 + 6 + 5 = 17$. Thus $\{1, 2, 3\} \in E$. Similarly, $x_1 = 1$, $x_3 = 1$, and $x_4 = 1$ is not feasible so $\{1, 3, 4\} \in E$. The same process can be repeated for all combinations of variables $1, 2, 3$, and 4. Every edge implies a specific inequality. These are listed below.

$x_1 + x_2 + x_3 \leq 2$

$x_1 + x_2 + x_4 \leq 2$

$x_1 + x_3 + x_4 \leq 2$

$x_2 + x_3 + x_4 \leq 2$

The conflict hypergraph has $V(H) = \{1, ..., 5\}$ and $E(H) = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{3, 4, 5\}\}$. Notice that the induced hypergraph of $\{1, 2, 3, 4\}$ is a $K_{4,3}$ hyperclique. Therefore, a $K_{4,3}$ hyperclique exists with $V(H) = \{1, ..., 4\}$ and $E(H_c) = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$. The associated hyperclique inequality $x_1 + x_2 + x_3 + x_4 \leq 2$ is clearly valid.

Similar to a clique, a hyperclique inequality is significant because it can also have facet defining characteristics. In general a hyperclique $K_{n,k}$ defines a valid inequality of the form $\sum_{i \in K_{n,k}} x_i \leq k - 1$. Since a hyperclique implies that every edge is present in the conflict hypergraph it suggests that only $k - 1$ variables can be selected. Selecting k variables would form an edge irrespective of which variables are taken. Therefore the hyperclique inequality is valid.

Another structure that creates a valid inequality is a hyperstar. A hyperstar $S_{n,m,l,k}$ induces a valid inequality. Recent work on hyperstars by [52] required $m = l$. In this case a hyperstar generates a valid inequality of the form $\sum_{i \in C^M} \frac{n-k+1}{l} x_i + \sum_{i \in C^P} x_i \leq n - m$. These researchers provide additional results on when these inequalities are facet defining.

Chapter 3

Simulataneous Uplifting with Constellations

Chapter 3 discusses the basics of the constellation inequalities as well as the criteria used to construct a constellation. Multiple theorems are given to assist in the proof and understanding of constellations. Furthermore, an example with the step by step process to determine the new constellation inequalities are shown. The inequalities are also shown to be facet defining, which is an important aspect of research in integer programming.

3.1 Constellation Background

The fundamental advancement of this thesis is the creation of the conflict hypergraph substructure that enables simultaneous lifting. This structure is referred to as a constellation, which is a collection of hyperstars and hypercliques across several conflict hypergraphs. Several preliminary comments must be given prior to providing a definition.

The basic idea of a constellation is to have a small hyperclique, K_{m,k_0}, surrounded by a larger hyperclique, K_{m+p,k_q}, with hyperstar substructures, denoted by S_{m+p,m,i,k_i} where i ranges from 1 to $k_0 - 1$, being contained in the hypergraphs consisting of edges ranging in size from $k_0 + 1$ to $k_q - 1$. The vertices of the smaller hyperclique become the middle of the constellation and the vertices of the larger hyperclique minus the vertices of the smaller hyperclique are the perimeter.

With these two hypercliques, several hyperstars must be found. The middle and perimeter of all of the hyperstars is the same as the middle and perimeter of the constellation. The vertices selected from the middle of the hyperstar range from 1 to the size of the edges in the smaller hyperclique minus one. Once the number of elements from the middle are determined, the hypergraph with the minimum sized edge must be determined where the induced subhypergraph of the middle and perimeter vertices contains a hyperstar with the appropriate properties.

Formally, a collection of uniform hypergraphs $\mathcal{H} = \{H_{k_0}, H_{k_1}, ..., H_{k_q}\}$ is a constellation with $m + p$ nodes, denoted by $C_{m,k_0,\mathcal{S},m+p,k_q}$ where $\mathcal{S}$ is a set of four tuples, if, and only if, the following properties are satisfied:

(i) Each H_{k_i} for $i = 1, ..., q$, has $m + p$ vertices and each set of vertices can be partitioned into the middle C^M and the perimeter C^P with $|C^M| = m$ and $|C^P| = p$.

(ii) In H_{k_0} the induced subhypergraph of C^M is a hyperclique.

(iii) In H_{k_q} the induced subhypergraph of $C^M \cup C^P$ is a hyperclique.

(iv) For each $(m + p, m, l, k_l) \in \mathcal{S}$, the hypergraph with edges of size $k \in l$ has a subhyperstar with middle C^M, perimeter C^P and l vertices taken from the middle.

Constellations are complicated subhypergraphic structures. In general determining the hyperstars that are a part of the constellations is a challenging problem. There is no way to know beforehand exactly what the edge size of the conflict hypergraphs will be nor how many edges will need to be argued to provide evidence of this hypergraph. However, there is a bound on the number of edges required given by $\begin{pmatrix} m \\ k_0 \end{pmatrix} + \begin{pmatrix} m+p \\ k_q \end{pmatrix}$ $+ \begin{pmatrix} k_q \\ (\lceil \frac{k_q}{2} \rceil) \end{pmatrix} * (\sum_{i=1}^{k_q - 1} \begin{pmatrix} m \\ i \end{pmatrix})$.

The term $\begin{pmatrix} m \\ k_0 \end{pmatrix}$ represents the edges needed for the middle hyperclique. Additionally, $\begin{pmatrix} m+p \\ k_q \end{pmatrix}$ signifies the edges needed for the perimeter hyperclique. Next, $\begin{pmatrix} k_q \\ (\lceil \frac{k_q}{2} \rceil) \end{pmatrix}$ is the maximum number of edges that could be found for any combination derived from a set of indices, and ($\sum_{i=1}^{k_q-1} \begin{pmatrix} m \\ i \end{pmatrix}$) represents each possibility of edges selected from the middle hyperclique. Together these factors contribute to the bound on edges needed to find a constellation.

With such a complicated structure, do constellations even exist? Provided that there are at least two hypercliques present in a multiple knapsack problem, a constellation can always be found in a conflict hypergraph. The following theorem formalizes this idea.

Theorem 3.1.1. *Given a P_{MK} with corresponding collection of conflict hypergraphs $H_1,\ldots,H_n$, if there exist K_{m,k_0} and K_{m+p,k_q} hypercliques where $V(K_{m,k_0}) \cap V(K_{m+p,k_q}) = V(K_{m,k_0}) \neq \emptyset$ and $p \geq 1$, then $H_{k_0},\ldots,H_{k_q}$ contains a constellation.*

Proof: Let $C^M = V(K_{m,k_0})$ and $C^P = V(K_{m+p,k_q}) \setminus V(K_{m,k_0})$ and (i) is satisfied. From the assumption, conditions (ii) and (iii) are satisfied by the two hypercliques. Since K_{m+p,k_q} is a hyperclique, there exists a hyperstar substructure in H_{k_q} of the form S_{m+p,m,m',k_q} for $m' = 1, \ldots, k_0 - 1$ and condition (iv) is satisfied.

□

Now that constellations are a common structure used in conflict hypergraphs, the next step is to find the appropriate valid inequalities. Since there will be multiple hyperstars, it is necessary to find a lifting coefficient that is valid for each of the hyperstars. To ensure that this occurs, the minimum lifting coefficient determined from the hyperstars has to be used. The lifting coefficient puts a higher weight on the variables in C^M, and together with the variables in C^P, a new valid inequality is created.

The set of hyperstars $\mathcal{S}$ is the set of hyperstars S_{m+p,m,j,k_j} where $m+p$ is the total number of vertices, j is the number of vertices selected from C^M, and k_j is the edge size for that particular hyperstar. The procedure to determine these hyperstars is simple to describe, but $\mathcal{NP}$-hard in general. Start by selecting one variable in the middle hyperclique, and then select variables from the perimeter until infeasibility is reached. Then select two variables from the middle, and select as many variables as needed to reach infeasibility. Each time the process is repeated take one more variable from the middle and follow the same steps. Continue to do this until $k_0 - 1$ variables have been taken from the middle. Once all of the hyperstars are determined, they form $\mathcal{S}$ where $\mathcal{S} = \{(m+p, m, 1, k_1),\ (m+p, m, 2, k_2), ..., (m+p, m, k_0-1, k_{k_0-1})\}$ and represents hyperstars in the appropriate hypergraph. All of this forms a constellation. The following theorem describes how the lifting coefficients for both valid inequalities are calculated from the set of hyperstars constructed from the two hypercliques.

Theorem 3.1.2. *Given a multiple knapsack with corresponding collection of conflict hypergraphs $\mathcal{H}$, if there exists a constellation with middle C^M and perimeter C^P of the form $C_{m,k_0,\mathcal{S},m+p,k_q}$ where $\mathcal{S}$ represents the hyperstars in the constellation and is $\{(m+p, m, 1, k_1),\ (m+p, m, 2, k_2), ..., (m+p, m, k_0-1, k_{k_0-1})\}$, then the following inequalities are valid for P_{MK}^{conv}:*

$\alpha \sum_{i \in C^M} x_i + \sum_{i \in C^P} x_i \leq k_q - 1$ *where* $\alpha \leq \alpha'^* = min\{min_{\{j=1,...,k_0-1\}}\{\frac{k_q-1-(k_j-j-1)}{j}\}, \frac{k_q-1}{k_0-1}\}$ *(1).*

$\sum_{i \in C^M} x_i + \alpha \sum_{i \in C^P} x_i \leq k_0 - 1$ *where* $\alpha \leq \alpha''^* = min\{min_{\{j=1,...,k_q-1\}}\{\frac{k_0-1-j}{k_j-j-1}\}, \frac{k_0-1}{k_q-1}\}$

(2).

Proof: Given a P_{MK} such that the conflict hypergraphs contain a constellation with middle C^M and perimeter C^P of the form $C_{m,k_0,\mathcal{S},m+p,k_q}$. It suffices to prove the extreme case when $\alpha = \alpha^*$ for any of the inequalities. If the extreme inequality is valid, then the inequality is valid for any $\alpha < \alpha^*$. The proof will treat each inequality separately, but the arguments are similar.

For contradiction, assume that constellation inequality (1) is invalid. Thus, there exists an $x' \in P_{MK}$ such that $\alpha \sum_{i \in C^M} x'_i + \sum_{i \in C^P} x'_i > k_q - 1$ (*). Assume $|\{x'_i = 1 : i \in C^M\}| = m'$ and $|\{x'_i = 1 : i \in C^P\}| = p'$. If $m' = 0$, then the inequality (*) reduces to either hyperclique inequality or there are not enough coefficients in C^P to violate this inequality, a contradiction.

If $p' = 0$, then $m' \leq k_0 - 1$ due to the K_{m,k_0} hyperclique. Since $\alpha \leq \frac{k_q-1}{k_0-1}$, $*$ reduces to $\alpha \sum_{i \in C^M} x'_i \leq \frac{k_q-1}{k_0-1}(k_0 - 1) = k_q - 1$, a contradiction. Thus, $m' \geq 1$ and $p' \geq 1$.

Equation (*) now reduces to $\alpha m' + p' > k_q - 1$ and $\alpha > \frac{k_q-1-p'}{m'}$. However, $H_{k_{m'}}$ does not contain a hyperstar with C^M, C^P and m' vertices selected from C^M due to the existence of x' where $k_{m'} = m' + p'$. Consequently, the hyperstar in the constellation that contains m' vertices from C^M must have edges of size $k_{m'}$ where $k_{m'} \geq m' + p' + 1$. Thus, $(m + p, m, m', k_{m'}) \in \mathcal{S}$ where $k_{m'} \geq m' + p' + 1$. Since α is less than or equal to the minimum, it must be less than the particular case when m' vertices are taken from the hub. Thus, $\alpha \leq \frac{k_q-1-(m'+p'+1-m'-1)}{m'} = \frac{k_q-1-p'}{m'} < \frac{k_q-1-p'}{m'}$, which is a contradiction to $*$ and the first inequality is valid.

For contradiction, assume that constellation inequality (2) is invalid. Thus, there exists an $x'' \in P_{MK}$ such that $\sum_{i \in C^M} x_i^{''} + \alpha \sum_{i \in C^P} x_i^{''} > k_0 - 1$ (**). Assume $|\{x_i^{''} = 1 : i \in C^M\}| = m''$ and $|\{x_i^{''} = 1 : i \in C^P\}| = p''$. If $p'' = 0$, then the inequality (**) reduces to hyperclique inequality, which is clearly valid, a contradiction.

If $m'' = 0$, then $p'' \leq k_q - 1$ due to the K_{m+p,k_q} hyperclique. Since $\alpha \leq \frac{k_0-1}{k_q-1}$, ** reduces to $\alpha \sum_{i \in C^P} x_i'' \leq \frac{k_0-1}{k_q-1}(k_q - 1) = k_0 - 1$, a contradiction. Thus, $m' \geq 1$ and $p' \geq 1$.

Equation (**) now reduces to $m'' + \alpha p'' > k_0 - 1$ and $\alpha > \frac{k_0-1-m''}{p''}$. However, $H_{k_{m''}}$ does not contain a hyperstar with C^M, C^P and m'' vertices selected from C^M due to the existence of x'' where $k_{m''} = m'' + p''$. Consequently, the hyperstar in the constellation that contains m'' vertices from C^M must have edges of size $k_{m''}$ where $k_{m''} \geq m''+p''+1$. Thus, $(m + p, m, m'', k_{m''}) \in \mathcal{S}$ where $k_{m''} \geq m'' + p'' + 1$. Since α is the minimum, it must be less than the particular case when m'' vertices are taken from the hub. Thus, $\alpha \leq \frac{k_0-1-m''}{m''+p''+1-m''-1} = \frac{k_0-1-m''}{p''} < \frac{k_0-1-m''}{p''}$, which is a contradiction to $**$ and the result follows.

□

As mentioned in Chapter 2, defining at least one valid inequality for each and every facet sufficiently describes P^{conv}. Thus, the aim is that the new constellation inequalities are not only valid, but more importantly facet defining. Before this important result can be obtained, one more definition is necessary.

For each constellation, there exists an $\alpha'^* = min_{\{j=0,\ldots,k_0-1\}}\{\frac{k_q-1-(k_j-j-1)}{j}\}$. Define

j' to be the j that achieves this minimum. Similarly, let $\alpha''^{*} = min_{\{j=1,\ldots,k_q-1\}}\{\frac{k_0-1-j}{k_j-j-1}\}$ and j'' be the j that achieve this minimum. The next theorem describes when the constellation inequality defines a large dimensional face.

Theorem 3.1.3. *Given a multiple knapsack problem with corresponding collection of conflict hypergraphs $H_{k_0},\ldots,H_{k_q}$ and a $C_{m,k_0,\mathcal{S},m+p,k_q}$, then*

The constellation inequality (1) defines a face of dimension at least $m+p-1$ if $\alpha = \alpha'^{*}$, every vertex in C^P is in at least one minimal edge in K_{m+p,k_q} with $K_0+2 \leq K_q$ and every vertex in C^M is in at least one minimal edge in $S_{m+p,m,j',k_{j'}}$ where j' is the arg $min_{\{j=1,\ldots,k_q-1\}}\{\frac{k_0-1-j}{k_j-j-1}\}$.

Proof: To show that constellation inequality $\alpha'^{*}\sum_{i\in C^M} x_i + \sum_{i\in C^P} x_i \leq k_q - 1$ (1) with $\alpha'^{*} = min_{\{j=1,\ldots,k_0-1\}}\{\frac{k_q-1-(k_j-j-1)}{j}\}$ defines a face of dimension $m+p-1$, assume j' is the *arg* $min_{\{j=1,\ldots,k_q-1\}}\{\frac{k_0-1-j}{k_j-j-1}\}$, every vertex in C^P is in at least one minimal edge in K_{m+p,k_q}, every vertex in C^M is in at least one minimal edge in $S_{m+p,m,j',k_{j'}}$. Assume similar assumptions for (ii).

Clearly, the dimension of a multiple knapsack problem is full dimensional ($dim(P^{conv}_{MK} = n)$) as long as no $a_{i,j} > b_i$, which is a standard assumption for all multiple knapsack problems. From Theorem 3.1.2 (i) and (ii) are valid. From the assumption regarding $a_{i,j} > b_i$, H_1 contains no edges and so k_0 and $k_q \geq 2$. Thus, the origin never satisfies (i) or (ii) at equality since the right hand side is at least one. Consequently, both (i) and (ii) induce faces of dimension at most $n-1$. So it remains to find n affinely independent points that meet each of these inequalities at equality.

Hooker [52] has shown that if there exists a hyperclique $K_{m,k}$ in a conflict hypergraph such that each vertex belongs to a minimal edge, then there there exist m points that meet the hyperclique inequality at equality. Consequently, P contains $|C^M|$ affinely independent points that meet $\sum_{i \in C^M} x_i = k_0 - 1$. Clearly each of these points have all variables associated with indices in C^P set equal to 0 and meet this constellation inequality at equality.

Finding the remaining $n - |C^M|$ affinely independent points is fairly straightforward and follows a similar line of reasoning as used by Hooker. From the assumption every vertex in C^P is in at least one minimal edge in the hyperstar in $H_{k_{j''}}$. Now create a graph $G = (C^P, E_G)$ by having $\{u, v\} \in E_G$ if u and $v \in C^P$ and u and v are in the same minimal edge in the hyperstar in $H_{k_{j''}}$. The graph G naturally divides in to components and only component 1 will be considered here. Iteratively repeating this process provides $|C^P|$ more affinely independent points.

Due to the structure of the hyperstar every minimal edge in the hyperstar must contain at least 2 vertices in C^P or C^M does not induce a maximal hyperclique a contradiction to a constellation. Select a minimimal edge in the hyperstar and let this edge be denoted by $d = \{i_1, ..., i_{j''}, i_{j''} + 1, ..., i_{k_{j''}}\}$ where vertices $i_1, ..., i_{j''} \in C^M$ and $i_{j''} + 1, ..., i_{k_{j''}} \in C^P$. Include the points $\sum_{j=d} e_j - e_l$ for each $l \in \{i_{j''} + 1, ..., i_{k_{j''}}\}$. Let $D = \{i_{j''} + 1, ..., i_{k_{j''}}\}$.

In G find a vertex v_i adjacent to D. Thus there exists a minimial edge d' that contains v_i and at least one vertex from D. If $|d' \setminus D| = 1$, then include the point $\sum_{j \in d'} e_j - e_l$

for some $l \in (d' \cap D) \setminus C^M$. If $|d' \setminus D| \geq 2$, then include the points $\sum_{j \in d'} e_j - e_l$ for each $l \in d' \setminus D$. Set D to be $D \cup d'$ and iteratively repeat this process until D is equal to component 1. Repeating for each component generates the remaining points.

These points are clearly affinely independent as the bulk of the points comprise a cyclical permutation of say r ones over a $r+1$ rows and columns. Furthermore, each point is feasible due to the definition of a minimal edge and finally, each of these points meet the inequality at equality. Thus, this inequality has at least $m+p$ points in P that meet this inequality at equality and the result follows.

□

With this complicated proof, it is now trivial to follow a similar argument to show that the other constellation inequality can define a large dimensional face. Formally,

Corollary 3.1.4. *Given a multiple knapsack problem with corresponding collection of conflict hypergraphs* $H_{k_0},\ldots,H_{k_q}$ *and a* $C_{m,k_0,\mathcal{S},m+p,k_q}$*, then*

The constellation inequality (2) defines a face of dimension at least $m+p-1$ *facet defining if* $\alpha = \alpha''^*$*, every vertex in* C^M *is in at least one minimal edge in* K_{m,k_0} *and every vertex in* C^P *is in at least one minimal edge in* $S_{m+p,m,j'',k_{j''}}$ *where* j'' *is the arg* $min_{\{j=1,\ldots,k_q-1\}}\{\frac{k_0-1-j}{k_j-j-1}\}$.

Proof: An extremely similar proof exists for the other inequality. The difference is the points that begin are taken from the K_{m+p,k_q} and the points cycle through C^P. Then the points are found similarly by fixing the number of points taken from C^P and removing one from C^M in the appropriate minimal edge in the hyperstar. Thus, the result follows.

$\square$

With these two results, it is now straightforward to provide conditions when these inequalities are facet defining. While many conditions exist one of the simplest involves the maximality of a hyperclique.

Theorem 3.1.5. *Given a multiple knapsack problem with corresponding collection of conflict hypergraphs* $H_{k_0},...,H_{k_q}$ *that contain a* $C_{m,k_0,\mathcal{S},m+p,k_q}$*, which satisfies the conditions in Theorem 3.1.3, and* C^P *is a maximal hyperclique in* $H_{k_q} \setminus C^M$*, then the constellation inequality (1) is facet defining.*

Proof: Since C^P is a maximal hyperclique in $H_{k_q} \setminus C^M$, there exists does not exist at least one edge in H_{k_q} of the form $\{v_k\} \cup \{v_1, ...v_{k_q-1}\}$ for each $v_k \in H_{k_q} \setminus (C^P \cup C^M)$. Since this edge doesn't exist, the point $e_k + e_1 + ... + e_{k_q-1}$ is feasible and clearly meets the constellation inequality (1) at equality. Add these points to the previous $m + p$ points generated from Theorem 3.1.3. These points are clearly affinely independent and the result follows.

$\square$

A similar result is readily available for the other constellation inequality.

Theorem 3.1.6. *Given a multiple knapsack problem with corresponding collection of conflict hypergraphs* $H_{k_0},...,H_{k_q}$ *that contain a* $C_{m,k_0,\mathcal{S},m+p,k_q}$*, which satisfies the conditions in Theorem 3.1.3, and* C^M *is a maximal hyperclique in* $H_{k_0} \setminus C^P$*, then the constellation inequality (2) is facet defining.*

Proof: Since C^M is a maximal hyperclique in $H_{k_0} \setminus C^P$, there exists does not exist at least one edge in H_{k_0} of the form $\{v_k\} \cup \{v_1, ...v_{k_0-1}\}$ for each $v_k \in H_{k_0} \setminus (C^P \cup C^M)$. Since this edge doesn't exist, add the point $e_k + e_1 + ... + e_{k_0-1}$ to the previous $m + p$ points generated from Corollary 3.1.4. These points are clearly affinely independent and meet the inequality at equality. So the result follows.

□

The next section provides a detailed example for constructing a constellation given a multiple knapsack problem. This assists in understanding how the theory described above can be applied in a practical sense.

3.2 Constellation in a Multiple Knapsack

A constellation can be a difficult concept to visualize, let alone comprehend. Thus the next example combines all of the criteria required to form a constellation and uses the theorems to determine the new constellation inequalities. A detailed explanation of the process is given, followed by the valid inequalities and the affinely independent points to show that these inequalities are indeed facet defining for the given multiple knapsack problem.

Example 3.2.1 The following example depicts a constellation.

$$12x_1 + 17x_2 + 15_3 + 4x_4 + 18x_5 + 8x_6 + 9x_7 + 13x_8 + 9x_9 + 8x_{10} + 6x_{11} + 6x_{12} \leq 53$$

$$18x_1 + 15x_2 + 16_3 + 18x_4 + 6x_5 + 10x_6 + 11x_7 + 8x_8 + 10x_9 + 7x_{10} + 7x_{11} + 8x_{12} \leq 56$$

$$10x_1 + 12x_2 + 11_3 + 16x_4 + 17x_5 + 10x_6 + 9x_7 + 11x_8 + 8x_9 + 6x_{10} + 8x_{11} + 7x_{12} \leq 54$$

$$x_i = \{0, 1\} \ \forall \ i = 1, 2, ..., 12.$$

Observe that $\{1, 2, 3, 4, 5\}$ induces a $K_{5,4}$ hyperclique. This is because constraint 1 has edges $\{1, 2, 3, 5\}$ and $\{2, 3, 4, 5\}$. In addition constraint 2 has edges $\{1, 2, 3, 4\}$, $\{1, 2, 4, 5\}$, and $\{1, 3, 4, 5\}$. Notice all $\begin{pmatrix} 5 \\ \\ 4 \end{pmatrix} = 5$ edges are derived. Therefore, the valid hyperclique inequality is $x_1 + x_2 + x_3 + x_4 + x_5 \leq 4$.

Now the size of the large hyperclique must be determined and $\{1, ..., 12\}$ induces a $K_{12,7}$ hyperclique. This is because the point $(0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1)$ is feasible so there does not exist a $K_{12,6}$. However, the point $(0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1)$, which has the seven variables with the smallest coefficients from constraint 1, violates constraint 2 as $71 > 56$. Replacing any variable with another in this point would only increase the value of the point and therefore violate constraint 1. Thus, the $K_{12,7}$ is a hyperclique with the corresponding valid inequality $x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12} \leq 6$.

The basis of a constellation are two hypercliques with one vertex set a subset of the other vertex set. Clearly, the $K_{12,7}$ and $K_{5,4}$ hypercliques are such structures. The constellation is on vertices $\{1, 2, ..., 12\}$, which will be partitioned into the middle $C^M = \{1, 2, 3, 4, 5\}$ and the perimeter $C^P = \{6, 7, ..., 12\}$. Therefore, the simultaneously lifted inequality will take the form $\alpha'(x_1+x_2+x_3+x_4+x_5)+x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12} \leq 6$. A second inequality $x_1+x_2+x_3+x_4+x_5+\alpha''(x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12}) \leq 3$ is also formed.

To calculate α' and α'' it requires determining the edge size that would induce a hyperstar of the appropriate size on the conflict hypergraph generated from $\{1, 2, ..., 12\}$. Since there is a $K_{5,4}$, no four vertices from the middle can induce a feasible solution. Therefore, the only hyperstars of interest have 1, 2 or 3 vertices selected from the middle. Thus, we must find the minimum k_{q_1}, k_{q_2}, and k_{q_3} that make $S_{12,5,1,k_{q_1}}$, $S_{12,5,2,k_{q_2}}$, and $S_{12,5,3,k_{q_3}}$ hyperstars.

In this example, a total of 1462 conflict hypergraph edges are argued. This number is based on the two hypercliques found above, and the three hyperstars each with the appropriate k_{q_i} values that together constitute of a constellation. Although the number of edges has already been given, it is actually calculated after all of the hypercliques and hyperstars are determined. For this problem the max number of edges would be $\begin{pmatrix} 5 \\ 4 \end{pmatrix}$ $+ \begin{pmatrix} 12 \\ 7 \end{pmatrix} + \begin{pmatrix} 7 \\ 4 \end{pmatrix} * (\begin{pmatrix} 5 \\ 1 \end{pmatrix} + \begin{pmatrix} 5 \\ 2 \end{pmatrix} + \begin{pmatrix} 5 \\ 3 \end{pmatrix})$ and this equals 1672, which is greater than 1462.

There is a $S_{12,5,1,6}$ hyperstar. Observe that the point $(0,0,0,1,0,0,0,1,1,0,1,1)$ is a feasible point. Thus, there is not a $S_{12,5,1,5}$ hyperstar. In constraint 1, taking variable 2 from C^M and the five smallest coefficients corresponding to indices from C^P form the point $(0,1,0,0,0,1,0,0,1,1,1,1)$. Evaluating this point in constraint 1 leads to a sum of $54 > 53$ and so this point is not feasible. Similarly, taking variable 5 from C^M and the corresponding five smallest coefficients from C^P form the point $(0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1)$, which leads to a sum of $55 > 53$ and it violates constraint 1. Therefore any point with either variable 2 or 5 from C^M and five variables from C^P is not feasible and the corresponding edges exist in H_6.

Applying the same method to constraint 2 results in showing that either variable 1 or 4 from C^M and the five smallest coefficients from C_P violate this constraint. Therefore, these edges are in H_6.

In constraint 2, taking variable 3 from C^M and the five smallest coefficients of the indices from C^P form either the point $(0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1)$ or $(0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1)$. Choosing any other combination of five variables from C^P with variable 3 violates the second constraint. Now both $(0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1)$ and $(0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1)$ violate constraint 1 as $57 > 53$ and $56 > 53$, respectively. Therefore any point with variable 3 from C^M and five variables from C^P is not feasible.

Thus, there is no feasible point with one variable from C^M and five variables from C^P. Consequently, H_6 contains a $S_{12,5,1,6}$ hyperstar.

There is also a $S_{12,5,2,6}$ hyperstar. Observe that the point $(0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1)$ is a feasible point. Thus, there is not a $S_{12,5,2,5}$ hyperstar because it is missing at least one edge. In constraint 1, excluding variable 4, selecting any two of variables 1, 2, 3, or 5 from C^M and the four smallest coefficients of the indices from C^P violates this constraint.

Next in constraint 2, taking variable 4 along with any one of variables 1, 2, or 3 from C^M and the four smallest coefficients of the indices from C^P violates this constraint. Similarly, in constraint 3 taking variables 4 and 5 from C^M and the four smallest coefficients of the indices from C^P violates the third constraint.

Thus, there is no feasible point with any two of the five variables from C^M and four variables from C^P. Consequently, H_6 contains a $S_{12,5,2,6}$ hyperstar.

For the last case, there is a $S_{12,5,3,5}$ hyperstar. Observe that the point $(1,0,0,1,1,0,1,0,0,0,0,0)$ is a feasible point. Thus, there is not a $S_{12,5,3,4}$ hyperstar because it is missing at least one edge. In constraint 1, excluding variable 4, selecting any three of variables 1, 2, 3, or 5 from C^M and the two smallest coefficients of the indices from C^P violates this constraint.

In constraint 2, taking variable 4 along with any two of variables 1, 2, or 3 from C^M and the two smallest coefficients of the indices from C^P violates the constraint. Lastly, in constraint 3 taking variables 4 and 5, and one of variables 1, 2, or 3 from C^M and the two smallest coefficients of the indices from C^P violates the third constraint.

Thus, there is no feasible point with any three of the five variables from C^M and two variables from C^P and H_5 contains a $S_{12,5,3,5}$ hyperstar.

3.3 Constellation Summary

In summary H_4 contains a $K_{5,4}$ hyperclique, H_5 contains a $S_{12,5,3,5}$ hyperstar, H_6 contains $S_{12,5,1,6}$ and $S_{12,5,2,6}$ hyperstars, and H_7 contains a $K_{12,7}$ hyperclique. The conflict hypergraphs H_4,...,H_7 contain a constellation of the form $C_{5,4,\mathcal{S},12,7}$ where $\mathcal{S} = \{(12,5,1,6),(12,5,2,6),(12,5,3,5)\}$. Using Theorem 3.1.3 the simultaneous lifting coefficient is calculated by finding the minimum $\{\frac{k_q-1-p'}{m'}\}$. Thus, $S_{12,5,1,6}$ has $\alpha' = 2$, the $S_{12,5,2,6}$ has $\alpha' = 1.5$, the $S_{12,5,3,5}$ has $\alpha' = 1.6667$, and the $K_{5,4}$ has $\alpha' = 2$. The resulting constellation inequality is valid and is

$$1.5(x_1 + x_2 + x_3 + x_4 + x_5) + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} \leq 6.$$

It is shown that the valid constellation inequality is facet defining by the following twelve affinely independent points: $e_1 + e_5 + \sum_{i=10}^{12} e_i$, $e_2 + e_4 + \sum_{i=10}^{12} e_i$, $\sum_{i=3}^{5} e_i + \sum_{i=10}^{12} e_i - e_j$ for $j = 3, 4$ and 5, and $\sum_{i=6}^{12} e_i - e_j$ for $j = 6, ..., 12$. This signifies that the dimension of the face is eleven, which is one less than the dimension of P^{conv} and is therefore facet defining. The following matrix shows these points.

$$\left|\begin{array}{ccccc|ccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}\right|$$

The easiest points to identify are the last seven as these are the points from the C^P. Because C^P is a $K_{7,6}$ hyperclique, there are $\begin{pmatrix} 7 \\ 6 \end{pmatrix} = 7$ points with six 1's permuted through them. Clearly the rest of the indices in these points are set to zero. These seven points are clearly affinely independent and generated from the minimal edges.

The five remaining points relate to C^M. Each of these points must take two indices from C^M and three from C^P to satisfy the valid inequality at equality. These are the j' from the theorem. Using x_3, x_4, and x_5, three points can be found by selecting each combination of two out of three variables. This matrix consists of two 1's cyclically permuted through the three points, while x_1 and x_2 are set to zero. This is again due to the minimal edge $\{3, 4, 5, 10, 11, 12\}$. Because the only common divisor of three and two is one, this guarantees that these three points are affinely independent.

The other two points must include x_1 and x_2. In both cases feasibility is the main criteria so x_5 was used with x_1, and x_4 was used with x_2. These two points are also affinely independent since x_1 and x_2 were used mutually exclusively. For all five of these points x_{10}, x_{11}, and x_{12} were set to one, while the rest of the indices were set to zero to ensure feasibility. Since all twelve points are affinely independent it shows that the inequality is facet defining.

The nice thing about this method is that both inequalities can be found from the same hypergraphs and hyperstars. As mentioned earlier, H_4 contains a $K_{5,4}$ hyperclique, H_5 contains a $S_{12,5,3,5}$ hyperstar, H_6 contains $S_{12,5,1,6}$ and $S_{12,5,2,6}$ hyperstars, and H_7 contains a $K_{12,7}$ hyperclique. The conflict hypergraphs H_4,...,H_7 contain a constellation of the form $C_{5,4,\mathcal{S},12,7}$ where $\mathcal{S} = \{(12, 5, 1, 6), (12, 5, 2, 6), (12, 5, 3, 5)\}$. Using Theorem 3.1.3 the simultaneous lifting coefficient is calculated by finding the minimum $\{\frac{k_0-1-m''}{p''}\}$. Thus, $S_{12,5,1,6}$ has $\alpha'' = .5$, the $S_{12,5,2,6}$ has $\alpha'' = .333$, the $S_{12,5,3,5}$ has $\alpha'' = 0$, and the $K_{12,7}$ has $\alpha'' = .5$. The resulting constellation inequality is valid and is

$$x_1 + x_2 + x_3 + x_4 + x_5 + 0(x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12}) \leq 3.$$

It is shown that the valid constellation inequality is facet defining by the following twelve affinely independent: $e_i + e_{(i \ mod \ 5)+1} + e_{(i+1 \ mod \ 5)+1}$ for $i = 1, ..., 5$, $e_1 + e_4 + e_5 + e_i$ for $i = 6, ..., 12$. This signifies that the dimension of the face is eleven, which is one less than the dimension of P^{conv}, and is facet defining. The following matrix shows these points.

$$\left|\begin{array}{ccccc|ccccccc}
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right|$$

For the second inequality inequality, finding the twelve affinely independent points is much easier. The first five points are found by selecting three of the five variables in C^M and all variables in C^P are set to zero. The next seven points simply include x_1, x_4 and x_5 from C^M and then one variable from C^P with each variable in C^P used once. Since all twelve points are affinely independent it shows that this inequality is facet defining.

As seen above, identifying a constellation structure and determining a constellation inequality is a tedious process. Finding the appropriate hypercliques and hyperstars is an $\mathcal{NP}$-hard problem in general. However, if we restrict the IP, then some polynomial time methods may be obtainable. The next section discusses a constellation in a binary knapsack instance, which is the simplest IP.

3.4 Constellation in a Knapsack

A binary knapsack problem is more conducive for constellations not only because there is just one constraint, but because it allows for all of the variables to be sorted in descending order. The benefit is that as soon as an edge is found, numerous other edges exist of the same size because one could exchange a index in the edge with any index that has a higher coefficient and this new edge exists. An example of this is moving from a cover to an extended cover. Thus, determining the conflict hypergraphs and ultimately the resulting constellation inequality can be done much faster.

This basic sorting principle led to the linear [52], quadratic [78] and psuedopolynomial [58] time algorithms to simultaneous uplift a cover inequality in a knapsack constraint. Thus, it is natural to revisit the knapsack polyhedron to examine the constellations that these researchers unknowingly implemented. The following example presents these issues.

Example 3.4.1 Consider the feasible region of the following knapsack problem.

$$29x_1 + 29x_2 + 26_3 + 26x_4 + 16x_5 + 16x_6 + 16x_7 + 15x_8 + 15x_9 + 15x_{10} + 13x_{11} + 12x_{12} + 12x_{13} \leq 63$$

$$x_i = \{0, 1\} \;\forall\; i = 1, 2, ..., 13.$$

Observe that $\{1, 2, 3, 4\}$ induces a $K_{4,4}$ hyperclique. This is because the point $(1, 1, 1, 1, 0, 0, 0, 0,\ 0, 0, 0, 0, 0)$ is infeasible as $110 > 92$. Thus, the edge $\{1, 2, 3, 4\}$ exists and the valid hyperclique inequality is $x_1 + x_2 + x_3 + x_4 \leq 3$.

Now the size of the large hyperclique must be determined and $\{1, ..., 13\}$ induces a $K_{13,7}$ hyperclique. This is because the point $(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)$ is feasible so there does not exist a $K_{13,6}$. However, the point $(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)$, which has the seven variables with the smallest coefficients from the constraint violates the right hand side as $98 > 92$. Replacing any variable with another in this point would only increase the value of the point and therefore violate the constraint. Thus, the $K_{13,7}$ is a hyperclique with the corresponding valid inequality $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} \leq 6$.

The basis of a constellation are two hypercliques with one vertex set a subset of the other vertex set. Clearly, the $K_{13,7}$ and $K_{4,4}$ hypercliques are such structures. The constellation is on vertices $\{1, 2, ..., 13\}$, which will be partitioned into the middle $C^M = \{1, 2, 3, 4\}$ and the perimeter $C^P = \{5, 6, ..., 13\}$. Therefore, the simultaneously lifted inequality will take the form $\alpha'(x_1 + x_2 + x_3 + x_4) + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} \leq 6$. A second inequality $x_1 + x_2 + x_3 + x_4 + \alpha''(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13}) \leq 3$ is also formed.

To calculate α' and α'' it requires determining the edge size that would induce a hyperstar of the appropriate size on the conflict hypergraph generated from $\{1, 2, ..., 13\}$. Since there is a $K_{4,4}$, no four vertices from the middle can induce a feasible solution.

Therefore, the only hyperstars of interest have 1, 2 or 3 vertices selected from the middle. Thus, we must find the minimum k_{q_1}, k_{q_2}, and k_{q_3} that make $S_{13,4,1,k_{q_1}}$, $S_{13,4,2,k_{q_2}}$, and $S_{13,4,3,k_{q_3}}$ hyperstars.

There is a $S_{13,4,1,6}$ hyperstar. Observe that the point $(0,0,0,1,0,0,0,0,0,1,1,1,1)$ is a feasible point. Thus, there is not a $S_{13,4,1,5}$ hyperstar. In the constraint, taking variable 4 from C^M and the five smallest coefficients corresponding to indices from C^P form the point $(0,0,0,1,0,0,0,0,1,1,1,1,1)$. Evaluating this point leads to a sum of $93 > 92$ and so this point is not feasible. Replacing x_4 with any other variable in C^M would only increase the value of the point or keep it the same. Therefore any point with one variable from C^M and five variables from C^P is not feasible and the corresponding edges exist in H_6.

There is also a $S_{13,4,2,6}$ hyperstar. Observe that the point $(0,0,1,1,0,0,0,0,0,0,1,1,1)$ is a feasible point. Thus, there is not a $S_{13,4,2,5}$ hyperstar because it is missing at least one edge. In the constraint, selecting variables 3 and 4 from C^M and the four smallest coefficients of the indices from C^P forms the point $(0,0,1,1,0,0,0,0,0,1,1,1,1)$. This point violates the constraint as $104 > 92$. Replacing x_3 or x_4 with any other variable in C^M would only increase the value of the point. Therefore any point with two variables from C^M and four variables from C^P is not feasible and the corresponding edges exist in H_6.

For the last case, there is a $S_{13,4,3,4}$ hyperstar. Observe that the point $(0,1,1,1,0,0,0,0,0,0,0,0,0)$ is a feasible point. Thus, there is not a $S_{13,4,3,3}$ hyperstar because it is missing at least one edge. In the constraint, selecting variables 2, 3 and 4 from C^M and the smallest coefficient of the indices from C^P forms the point $(0,1,1,1,0,0,0,0,0,0,0,0,0,1)$. This point violates the constraint as $93 > 92$. Replacing either x_2, x_3 or x_4 with x_1 in C^M would only increase the value of the point. Therefore any point with three variables from C^M and one variable from C^P is not feasible and the corresponding edges exist in H_4.

In summary H_4 contains a $K_{4,4}$ hyperclique, H_6 contains $S_{13,4,1,6}$ and $S_{13,4,2,6}$ hyperstars, and H_7 contains a $K_{13,7}$ hyperclique. The conflict hypergraphs H_4, H_6, H_7 contain a constellation of the form $C_{4,4,\mathcal{S},13,7}$ where $\mathcal{S} = \{(13,4,1,6),(13,4,2,6),(13,4,3,4)\}$. Using Theorem 3.1.3 the simultaneous lifting coefficient is calculated by finding the minimum $\{\frac{k_q-1-p'}{m'}\}$. Thus, $S_{13,4,1,6}$ has $\alpha' = 2$, the $S_{13,4,2,6}$ has $\alpha' = 1.5$, the $S_{13,4,3,4}$ has $\alpha' = 2$, and the $K_{4,4}$ has $\alpha' = 2$. The resulting constellation inequality is valid and is

$$1.5(x_1+x_2+x_3+x_4)+x_5+x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12}+x_{13} \leq 6.$$

It is shown that the valid constellation inequality is facet defining by the following thirteen affinely independent points: $e_1+e_4+\sum_{i=11}^{13} e_i$, $e_3+e_4+\sum_{i=11}^{13} e_i$, $e_2+e_4+\sum_{i=11}^{13} e_i$, $e_2+e_3+\sum_{i=11}^{13} e_i$, $e_5+\sum_{i=9}^{13} e_i$, $e_6+\sum_{i=9}^{13} e_i$, and $\sum_{i=7}^{13} e_i - e_j$ for $j = 7, ..., 13$. This signifies that the dimension of the face is twelve, which is one less than the dimension of P^{conv} and is therefore facet defining. The following matrix shows these points.

$$
\left|\begin{array}{cccc|ccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}\right|
$$

Both inequalities can be found from the same hypergraphs and hyperstars. The conflict hypergraphs H_4, H_6, H_7 contain a constellation of the form $C_{4,4,\mathcal{S},13,7}$ where $\mathcal{S} = \{(13,4,1,6),(13,4,2,6),(13,4,3,4)\}$. Using Theorem 3.1.3 the simultaneous lifting coefficient is calculated by finding the minimum $\{\frac{k_0-1-m''}{p''}\}$. Thus, $S_{13,4,1,6}$ has $\alpha'' = .5$, the $S_{13,4,2,6}$ has $\alpha'' = .333$, the $S_{13,4,3,4}$ has $\alpha'' = \infty$, and the $K_{13,7}$ has $\alpha'' = .5$. The resulting constellation inequality is valid and is

$$x_1 + x_2 + x_3 + x_4 + .333(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13}) \leq 3.$$

It is shown that the valid constellation inequality is facet defining by the following thirteen affinely independent points: $e_1+e_4+\sum_{i=11}^{13} e_i$, $e_3+e_4+\sum_{i=11}^{13} e_i$, $e_2+e_4+\sum_{i=11}^{13} e_i$, $e_2+e_3+\sum_{i=11}^{13} e_i$, $e_3+e_4+e_5+e_{12}+e_{13}$, $e_3+e_4+e_6+e_{12}+e_{13}$, $e_3+e_4+e_7+e_{12}+e_{13}$, $e_3+e_4+e_8+e_{12}+e_{13}$, $e_3+e_4+e_9+e_{12}+e_{13}$, and $e_3+e_4+\sum_{i=10}^{13} e_i-e_j$ for $j = 10, ..., 13$. This signifies that the dimension of the face is twelve, which is one less than the dimension of P^{conv} and is therefore facet defining. The following matrix shows these points.

$$
\left|\begin{array}{cccc|ccccccccc}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}\right|
$$

Notice that in this example there was not a hyperstar that corresponded to H_5. This simply suggests that no combination of exactly five variables from C^M or C^P creates a minimal edge. However, for the purposes of a constellation it is not required that conflict hypergraph of every edge size be present. Instead, the presence of the hyperstars is of importance and is what ultimately leads to the determination of the lifting coefficient.

Chapter 4

Conclusion

The primary achievement of this thesis is to add to the general body of knowledge in integer programming. This research takes a broad approach to simultaneous lifting and provides theoretical background for it. Both knapsack and multiple knapsack instances are considered for implementing this method.

The major break through is the discovery of a constellation structure. This is beneficial because a constellation examines the underlying hypergraphic structures of multiple conflict graphs that exist in a knapsack or multiple knapsack problem at the same time. A constellation inequality builds on these principles by combining two hypercliques and multiple hyperstars together to determine the constellation structure. This is the most significant result because constellations assist in deriving the fundamental structure for exact simultaneous lifting.

Another advancement is the formulation of constellation inequalities. Most important is that one constellation can be used to create two distinct inequalities. Furthermore, these constellation inequalities are potentially facet defining, and this research provides some conditions for this to occur.

Finally, this thesis takes these abstract ideas into real examples. Two examples from the knapsack and multiple knapsack problems demonstrate the power and existence of the constellation structures and their inequalities. It is noteworthy that these constellation inequalities are derived without solving an integer program.

Fundamentally, this research provides the theoretical basis to help researchers generate more efficient techniques to perform simultaneous lifting for various classes of integer programs. Thus, there exists a substantial amount of research that should be pursued as a result of this work and the next section discusses some of these topics.

4.1 Future Work

One promising area of research would be to develop a polynomial time algorithm to generate valid constellation inequalities. Currently, using this method is extremely complicated and does not a have a significant benefit other than the knowledge and theory behind simultaneous lifting. However, an efficient algorithm to determine and implement constellation inequalities in multiple knapsack problem, or even a general integer program, would allow simultaneous lifting to be applied in more situations.

A constellation is proven to define two valid inequalities, but after concluding this research, it became evident that there is another valid inequality in Example 3.2.1. Notice that the points that generated the facet defining inequalities used 2 in C^M and 3 in C^P, and 0 in C^M and 6 in C^P, while the other inequality used 3 in C^M and 0 in C^P, and 3 in C^M and 1 in C^P. It appears that there is a third inequality that would use 2 in C^M and 3 in C^P, and 3 in C^M and 1 in C^P. The inequality generated from this would be $2(x_1 + ... + x_5) + x_6 + ... + x_{12} \leq 7$. Observe that this is not a lifted cover inequality, which makes this an interesting research area. Jennifer Bolton [17] is using this idea and the knapsack polytope as her thesis topic.

In this thesis, the constellation inequalities focus on hyperstars with two sets of variables, the middle and the perimeter. However, it is possible that more than two sets of variables could be simultaneously lifted. The challenge would be to identify structures that have three embedded hypercliques and understanding how a hyperstar has to change to enable an iterative form of simultaneous lifting.

This research provides a theoretical understanding of the relationship between simultaneous lifting and hypergraphic structures in integer programming, and specifically in the multiple knapsack problem. By providing the theory behind simultaneous lifting, researchers should be able to apply this knowledge to develop new algorithms that enable simultaneous lifting to be performed faster and over more complex integer programs. This should enable a reduction in the solution time of the integer programs.

Bibliography

[1] Ahmed, S. and N. V. Sahinidis (2003). "An approximation scheme for stochastic integer programs arising in capacity expansion ," *Operations Research*, **51** (3), 461-471.

[2] Atamtürk, A., G.L. Nemhauser, and M. W. P. Savelsbergh, (2000). "The mixed vertex packing problem," *Mathematical Programming*, **89**, 35-53.

[3] Atamtürk, A., G.L. Nemhauser, and M. W. P. Savelsbergh, (2000). "Conflict graphs in solving integer programming problems," *European Journal of Operational Research*, **121** (1), 40-55.

[4] Atamtürk, A. (2003). "On the facets of the mixed-integer knapsack polyhedron," *Mathematical Programming*, **98** (1-3), 145-175.

[5] Atamtürk, A. (2004). "Sequence independent lifting for mixed-integer programming," *Operations Research*, **52** (3), 487-491.

[6] Balaban, A. T. (1985). "Graph theory and theoretical chemistry," *THEOCHEM*, **21**, 117-142.

[7] Balas, E. (1975). "Facets of the knapsack polytope," *Mathematical Programming*, **8**, 146-164.

[8] Balas, E. (1971). "Intersection cuts-a new type of cutting planes for integer programming,"*Operations Research*, **19**, 19-39.

[9] Balas, E. and E. Zemel (1978). "Facets of the knapsack polytope from minimal covers," *SIAM Journal of Applied Mathematics*, **34**, 119-148.

[10] Balas, E. and E. Zemel (1984). "Lifting and complementing yields all facets of positive zero-one programming polytopes," *Proceedings of the International Conference on Mathematical Programming* (R.W. Cottle et. al., ed.), 13-24.

[11] Balas, E. and S. M. Ng (1989). "On the set covering polytope. II, Lifting the facets with coefficients in 0,1,2," *Mathematical Programming*,**45** (1), 1-20.

[12] Balas, E. and M. Fishetti (1993). "Lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets," *Mathematical Programming*,**58** (3), 325-352.

[13] Barg, A., and Zemor, G. (2008). "Codes on hypergraphs," *2008 IEEE International Symposium on Information Theory Conference*, 156-160.

[14] Berge, C., (1973). "*Graphs and Hypergraphs*," North-Holland Publishing Company, translated by Minieka, E.

[15] Bik, A. and Kreitzer, D. and Tian, X. (2008). "A case study on compiler optimizations for the Intel Core 2 Duo processor ," *International Journal of Parallel Programming*, **36**, 571-591.

[16] Bixby, R. E. and E. K. Lee, (1994). "Solving a Truck Dispatching Scheduling Problem Using Branch and Cut," *Operations Research*, **46**, No. 3, 355-367.

[17] Bolton, Jennifer (2009). "Synchronized simultaneous lifting in binary knapsack polyhedra," *MS Dissertation*, Department of Industrial and Manufacturing Systems Engineering, Kansas State University.

[18] Borndörfer, R., (1997). *Aspects of Set Packing, Partitioning, and Covering*, Ph.D. thesis,Technischen Universtät Berlin, Berlin Germany.

[19] Brigandi, A. J. and Dargon, D.R. and Sheehan, M.J. and Spencer, T. (1994). "AT&T's Call Processing Simulator (CAPS) Operational Design for Inbound Call Centers," *INTERFACES*, **24**, 6-28.

[20] Campbell. J. F. (1994). "Integer programming formulations of discrete hub location problems," *European Journal of Operational Research*, **72** (2), 387-405.

[21] Caprara. A., M. Fischetti, P. Toth, D. Vigo, P. L. Guida (1997). "Algorithms for railway crew management", *Mathematical Programming*, **79** (1-3), 125-141.

[22] Carr, R. (1996). "Separating over classes of TSP inequalities defined by 0 node-lifting in polynomial time," *Integer Programming and Combinatorial Optimization. 5th International IPCO Conference Proceedings*, 460-474.

[23] Cetin, E. (2007). "Handling organ motion in radiotherapy of cancer via Markov chains ," *Applied Mathematics and Computation*, **184**, 149-155.

[24] Chen, Y, and Wang, T, and Wong, D.F. (1993). "A graph partitioning problem for multiple-chip design," *IEEE International Symposium on Circuits and Systems*, 1778-1781.

[25] Chvátal, V., (1973). "Edmonds Polytopes and a Hierarchy of Combinatorial Problems", *Discrete Mathematics*, **4**, 305-337.

[26] Cho C., D., M. Padberg, and M. Rao (1983). "On the uncapacitated plant location problem. II. Facets and lifting theorems," *Mathematics of Operations Research*, **8** (4), 590-612.

[27] Cohen, G.D., and Litsyn, S., and Zemor, C. (1996). "On greedy algorithms in coding theory," *IEEE Transactions on Information Theory*, 6, 2053-2057.

[28] Cohen, M., and Kamesam, P.V., and Kleindorfer, P., and Lee, H., and Tekerian, A. (1990). "Optimizer: IBM's Multi-Echelon Inventory System for Managing Service Logistics," *INTERFACES*, 20, 65-82.

[29] Dahan, X., M. Maza, E. Schost. W. Wu, and Y. Xie (2005). "Lifting techniques for triangular decompositions," *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation ISSA '05.*

[30] Dai, Qin-Yun, and Hu, Xiu-Lin, and Yu, Hong-Yi, and Zhao, Jun (2005). "On the capacity of wireless ad-hoc network basing on graph theory," *4th International Conference on Networking.*.

[31] Das, N., Bhattacharya, B.B., and Dattagupta, J. (1990). "Analysis of conflict graphs in multistage interconnection networks ," *IEEE TENCON'90: 1990 IEEE Region 10 Conference on Computer and Communication Systems*, 175-179.

[32] De Farias Jr, I., E. Johnson, and G. Nemhouser (2002). "Facets of the complimentarity knapsack polytope," *Mathematics of Operations Research*, **27** (1), 210-227.

[33] Deo, S. (1974). "Graph theory with applications to engineering and computer science," *Prentice-Hall, xvii+478pp* .

[34] De Simone, C., (1990). "Lifting facets of the cut polytope," *Operations Research Letters* **9** (5), 341-344.

[35] Diestel, R. (2000). "Graduate Texts in Mathematics," *Springer; 3rd edition*.

[36] Easton, T. and K. Hooker, "Simultaneously Lifting Sets of Binary Variables into Cover Inequalities for Knapsack Polytopes," *Discrete Optimization, Special Issue: In Memory of George B. Dantzig*, **5**(2) May 2008, 254-261.

[37] Felici, G., and C. Gentile (2003). "Zero-lifting for integer blocks structured problems," *Journal of Combinatorial Optimization*,**7** (2), 161-167.

[38] Fernandez, F, and Gutierrez, J (1988). "Application of algebraic graph theory to the analysis and design of bit-serial architectures," *VLSI Signal Processing, III*, 352-361.

[39] Fidanova S. (2004). "Monte Carlo method for multiple knapsack problem," *Large-Scale Scientific Computing. 4th International Conference, LSSC 2002. Revised Papers*, 136-143 .

[40] Freidenfelds. J (1980). "Capacity expansion when demand is a birth-death random process ," *Operations Research*, **28** (3), 712-721.

[41] Gomory, R. E., (1958). "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, **64**, 275-278.

[42] Gomory, R. (1960). "Solving linear programming problems in integers," *Combinatorial Analysis*, R.E. Bellman and M. Hall, Jr. eds., American Mathematical Society 211-216.

[43] Gomory, R. (1963). "An algorithm for integer solutions to linear programs," *Recent Advances in Mathematical Programming*, R. Graves and P. Wolfe, eds. McGraw-Hill 269-302.

[44] Gomory, R. (1969). "Some polyhedra related to combinatorial problems," *Linear Algebra and its Applications*, **2**, 451-558.

[45] Gottlieb, E. S. and M. R. Rao, (1990). "The generalized assignment problem: Valid inequalities and facets," *Mathematical Programming*,**46** (1-3).

[46] Gu, Z., G. Nemhauser, and M. Savelsbergh (1998). "Lifted cover inequalities for 0-1 integer programs: computation," *Journal of Computing*, **10** (4), 427-437.

[47] Gu, Z., G. Nemhauser, and M. Savelsbergh (1999). "Lifted cover inequalities for 0-1 integer programs: computation," *Mathematical Programming*, **85** (3), 439-467. 109-121.

[48] Gu, Z., G. Nemhauser, and M. Savelsbergh (2000). "Sequence independent lifting in mixed integer programming," *Journal of Combinatorial Optimization*, **4**, 109-129.

[49] Gutierrez, Talia (2007) "Lifting general integer variables," *MS Dissertation*, Department of Industrial and Manufacturing Systems Engineering, Kansas State University.

[50] Hammer, P., E. Johnson, and U. Peled (1975). "Facets of regular 0-1 polytopes," *Mathematical Programming*, **8**, 179-206.

[51] Harju, T. and Chang, Li and Petre, I. (2008). "Graph theoretic approach to parallel gene assembly,"*Discrete Applied Mathematics*, **156**, 3416-3429.

[52] Hooker, Kevin, (2005) "Hypergraphs and integer programming polytopes, *Ph.D. Dissertation*, Department of Industrial and Manufacturing Systems Engineering, Kansas State University.

[53] Jiong-cheng, L. and Jiang-hong, B. (2009). "New method based on number theory for integer programming of combinatorial optimization ," *Computer Engineering and Design*, **30**, 1276-1278.

[54] Karabuk, S. and S.D. Wu (2003). "Coordinating strategic capacity planning in the semiconductor industry," *Operations Research*, **51** (6), 839-849.

[55] Karp, R. (1972). "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York 85-103.

[56] Kaski, P. and Penttinen, A. and Suomela, J. (2007). "Coordinating concurrent transmissions: a constant-factor approximation of maximum-weight independent set in local conflict graphs,"*Ad-Hoc, Mobile, and Wireless Networks. Proceeedings 6th International Conference*, 74-86.

[57] Koster, A., S. Hoesel, and A. Kolen (1998). "The partial constraint satisfaction problem: facets and lifting," *Operations Research Letters*, **23** (3), 89-98.

[58] Kubik, Lauren (2009). "Simultaneously lifting multiple sets in binary knapsack integer programs," *MS Thesis*, Department of Industrial and Manufacturing Systems Engineering, Kansas State University.

[59] Kuonen, P. and Ubeda, S. and Zerovnik, J. (1996). "Graph theory applied to mobile network optimisation," *Elektrotehniski Vestnik*, **63** (2), 69-72.

[60] Laakso, J. and Koljonen, T. and Rinne, M. (1998). "A radio resource knapsack packing for WTDMA air interface," *ICUPC '98. IEEE 1998 International Conference on Universal Personal Communications. Conference Proceedings*, **1**, 255-260.

[61] Land, A.H. and Doig, A.G. (1960). "An automatic method for solving discrete programming problems," *Econometrica*, **28**, 497-520.

[62] Lesne, A. (2006). "Complex networks: From graph theory to biology,"*Letters in Mathematical Physics*, **78**, 235-262.

[63] Marchand, H. and Martin, A. and Weismantel, R. and Wolsey, L., (2002). "Cutting planes in integer and mixed integer programming,"*Discrete Applied Mathematics*, **123**, 397-446.

[64] Nauss, R. M. (1975). "The 0-1 knapsack problem with multiple choice constraints," *Bulletin of the Operations Research Society of America*, **23**.

[65] Nemhauser, G. L. and P. H. Vance, (1994). "Lifted cover facets of the 0-1 knapsack polytope with GUB constraints," *Operations Research Letters*,**16** (5), 255-263.

[66] Nilsson, O. and D. Sjelvgren,(1995) "Mixed-integer programming applied to short-term planning of a hydro-thermal system," *IEEE Transactions on Power Systems*, **11** (1), 281-286.

[67] Padberg, M. (1973). "On the facial structure of set packing polyhedra," *Mathematical Programming*, **5**, 199-215.

[68] Park, K. (1997). "Lifting cover inequalities for the precedence-constrained knapsack problem," *Discrete Applied Mathematics*, **72** (3), 219-241.

[69] Pendharkar, P. C. and Rodger, J. A. (2006). "Information technology capital budgeting using a knapsack problem" *International Transactions in Operational Research*, **13** (4), 333-351.

[70] Pogliani, L., (2007). "Old and new concepts in chemical graph theory,"*AIP Conference Proceedings*, **963**, 540-543.

[71] Pulleyblank, W. R. and J. Edmonds, (1975). "Facets of 1-Matching Polyhedra," in *Hypergraph Seminar*, C. Berge and D. Ray-Chaudhuri, eds., Springer, 214-242.

[72] Pursimo J. M., H. K. Antila, M. K. Vilkko and P. A. Lautala (1998). "A short-term scheduling for a hydropower plant chain," *International Journal of Electrical Power and Energy Systems*, **20** (8), 525-532.

[73] Re, A. and Molineris, I. and Caselle, M., (2008). "Graph theory analysis of genomics problems: Community analysis of fragile sites correlations and of pseudogenes alignments,"*Computers and Mathematics with Applications*, **55**, 1034-1043.

[74] Richard J., I. De Farias, and G. Nemhauser (2003). "Lifted inequalities for 0-1 mixed integer programming: Superlinear lifting," *Integer Programming*, **98** (1-3), 115-143.

[75] Rosenberg, I.G., (1975). "On Chvatal's cutting planes in integer linear programming ,"*Mathematische Operationsforschung und Statistik*, **6**, 511-522.

[76] Santanu, S. D. and P. R. Jean-Philippe, (2006). "Linear programming based lifting and its application to primal cutting plane algorithms," School of Industrial Engineering, Purdue University.

[77] Schmidt, G., and Zyablov, V.V., and Bossert, M. (2003). "On expander codes based on hypergraphs," *Proceedings 2003 IEEE International Symposium on Information Theory*.

[78] Sharma, K. (2007). "Simultaneously Lifting Sets of Variables in Binary Knapsack Problems," *MS Dissertation*, Department of Industrial and Manufacturing Systems Engineering, Kansas State University.

[79] Shebalov, S. and D. Klabjan, (2006) "Sequence independent lifting for mixed integer programs with variable upper bounds," *Mathematical Programming*, **105** (2-3), 523-561.

[80] Shyi-Long, Lee and Chiuping, Li, (1994). "Chemical signed graph theory,"*International Journal of Quantum Chemistry*, **49**, 639-648.

[81] Wattenhofer, R (2004). "Wireless networking: graph theory unplugged," *Graph-Theoretic Concepts in Computer Science. 30th International Workshop, WG 2004*, 20-32.

[82] Wedlin, D. (1995) "An algorithm for large scale 0-1 integer programming with application to airline crew scheduling," *Annals of Operations Research*,**57**(1), 238-301.

[83] Wolsey, L.A. (1975). “Faces for a linear inequality in 0-1 variables,” *Mathematical Programming*, **8**, 165-178.

[84] Wolsey, L.A. (1975b). “Facets and strong valid inequalities for integer programs,” *Operations Research*, **24**, 367-372.

[85] Wolsey, L.A. (1976). “Facets and strong valid inequalities for integer programs,” *Operations Research*, **24**, 367-372.

[86] Wolsey, L.A. (1977). “Valid inequalities and superadditivity of 0/1 integer programs,” *Mathematics of Operations Research*, **2**, 66-77.

[87] Xiaohong Guan, Ni Ernan, Li Renhou and P. B. Luh, (1997).“ An optimization-based algorithm for scheduling hydrothermal powersystems with cascaded reservoirs and discrete hydro constraints,” *IEEE Transactions on Power Systems*, **12** (4), 1775-1780.

[88] Yen, J. W., J. R. Birge (2006). “A stochastic programming approach to the airline crew scheduling problem,” *Transportation Sciences*, **40** (1), 3-14.

[89] Zemel, E. (1978). “Lifting the facets of 0-1 polytopes” *Mathematical Programming*, **15**, 268-277.

[90] Zemel, E. (1989). “Easily computable facets of the knapsack polytope,” *Mathematics of Operations Research*.

[91] Zhao, P. and Zhao, P. and Zhang, X. (2006). "A new ant colony optimization for the knapsack problem," *2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design.*