Valid Inequalities for the Quadratic Knapsack Problem

Gloria BURENGENGWA (gloria@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by:Dr Trivikram Dokka Venkata Satyanaraya
Dr Franklin Djeumou Fomeni
Lancaster University management School, UK

18 May 2017

Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

The quadratic knapsack problem is an optimisation problem which consists of maximising a quadratic (profit) function with binary variables subject to a single knapsack constraint. This is a well-studied combinatorial optimisation problem, with a variety of important applications, for example in the location of satellites, airports, railway stations and freight terminals. It is known to be NP-hard in the strong sense. Exact solution techniques for this problem use the branch-and-cut algorithm, and rely on the availability of strong valid inequalities or cutting planes. In this project, by using the software package PORTA, different polyhedral structures for the quadratic knapsack problem represented in the form of a graph are analysed and some interesting inequalities as extended cover inequalities, tree inequalities, are found.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Gloria BURENGENGWA, 18 May 2017

Contents

Abstract **i**

1 Introduction **1**
- 1.1 Mathematical Formulation 1
- 1.2 Quadratic Knapsack Problem 2
- 1.3 The Cutting Planes 2
- 1.4 Structure of the Essay 4

2 The Linear Knapsack Problem **5**
- 2.1 Definition of the Linear Knapsack Problem 5
- 2.2 Valid Inequalities for the Linear Knapsack Problem 6

3 The Quadratic Knapsack Problem **9**
- 3.1 Definition of the Quadratic Knapsack Problem 9
- 3.2 Valid Inequalities for the Quadratic Knapsack Problem 10

4 Experimentation with PORTA **15**
- 4.1 Definitions 15
- 4.2 PORTA Software 16
- 4.3 Experimentation 17

5 Conclusion **24**
- 5.1 Future work 24

References **27**

1. Introduction

Optimization is a crucial tool in decision science as it is in the analysis of physical systems. In order to use optimization, it is necessary or even essential to identify an objective which represents a quantitative measure of the performance of the system under consideration. This can be any quantity that can be represented by a number such as the cost, the profit, the processing time, the capacity, the energy consumption, the efficiency. The objective is determined by certain characteristics of the system called variables. The goal of optimization is to find the values of the variables that minimize or maximize the objective. The feasible set of the problem is determined by certain constraints which limit the values of the variables optimizing the objective.

1.1 Mathematical Formulation

An optimization problem is written in the form of

$$\min_{x \in F} f(x) \tag{1.1.1}$$

or

$$\max_{x \in F} f(x) \tag{1.1.2}$$

where x is a variable and F a *feasible set* defined by $F = \{x \in \mathbb{R}^n : h(x) \geq 0, s(x) = 0\}$, where $h(x)$ and $s(x)$ are real-valued functions. The function f is the objective defined as $F \to S$, where $F \subseteq \mathbb{R}^n$ and $S \subseteq \mathbb{R}$.

There exist three types of optimization, namely continuous, discrete and mixed optimization. In continuous optimization, the values of the variable can be real. The *continuous optimization* can be formulated as a linear and non linear programming also as a convex. The variables of a discrete optimization are characterized by discrete values. The *discrete optimization* is composed by two significant domains such as the combinatorial optimization and integer programming since the variables are restricted to be discrete. Similar to continuous optimization, discrete optimization problems can be studied as linear, non linear, convex and non convex optimization problems. The third type which is the *mixed integer optimization*, combines the continuous and the discrete optimizations.

For (1.1.1) and (1.1.2), we have a continuous optimization when F is continuous and a discrete optimization otherwise.

A point $x \in \mathbb{R}^n$ is a *feasible point* for (1.1.1) and (1.1.2) if $x \in F$. When the interior of F is non empty, x is a *strictly feasible point* for (1.1.1) and (1.1.2) if x is in the interior of F. And since an optimization problem seeks to find the maxima or minima of a function, we define a *local maximum* (*local minimum*), a point $x^* \in F$ if $f(x^*) \geq f(x)$ $(f(x^*) \leq f(x))$ for all neighbourhood of x^*. We define a *global maximum* (*global minimum*) if $f(x^*) \geq f(x)$ $(f(x^*) \leq f(x))$ for all $x \in F$. So a point $x^* \in F$ is called an *optimizer* of f if x^* is a maximizer or a minimizer of f.

Optimization has many real life applications. Airline companies schedule aircraft and crews to minimize the cost. In nature, physical systems tend to state of minimum energy. Rays of light follow path that minimize their travel time. Investors seek to create portfolios that achieve high return while avoiding excessive risks. In this essay, we will focus on a combinatorial optimization problem called the quadratic knapsack problem.

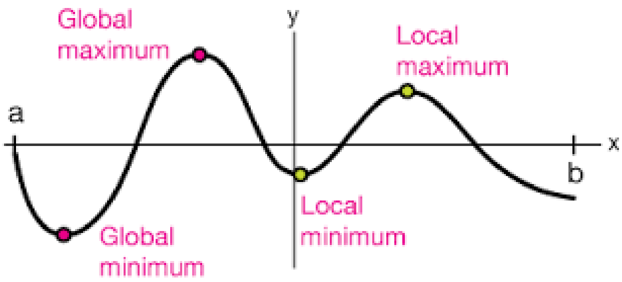

Figure 1.1: Illustration of global maximum and global minimum.

1.2 Quadratic Knapsack Problem

The quadratic knapsack problem (QKP) naturally appears in many types of problems in combinatorics, statistics and operations research. It has a wide spectrum of applications in finance in portfolio investment: For instance, given a number of assets $|N|$ owned by a company, where N represents a set of investments and c the return expected. This can be written in the form of a QKP (3.1.1) as it is shown in Chapter 3 where x_i is the rate of money invested in an asset i and p_{ij} the covariance between return of asset i and return of asset j (Mansini and Speranza, 1999). The technology of communication satellites uses a new technique in which the electronical transmission of messages is better than a physical transmission. Physical messages are converted into electronic ones and vice versa by electronic stations handling messages and that communicate with each other through satellites (Witzgall, 1975). Known n potentials sites for the stations, the investment cost c_i for building a station in site i, and the average daily mail p_{ij} observed between i and j, the problem of selecting a set $Q \subseteq \{1, \dots, N\}$ of locations such that the global traffic $\sum i \in Q \sum_{j \in Q} p_{ij}$ is maximised and a budget constraint $\sum_{j \in Q} c_i \leq b$ is met is obviously of the form (3.1.1) as expressed in Chapter 3 (Gallo et al., 1980). The QKP is also applied in location of freight handling terminals and airports. A problem is NP-hard when it cannot be solved in a polynomial time so the big challenge is to solve the quadratic knapsack problem that is known to be strongly NP-hard (Parker and Rardin, 2014). Several methods have been proposed for it, among them we count the Lagrangian based algorithm method, the cutting planes methods and the cut and branch that combine the cutting planes and the branch and bound methods (Caprara et al., 1999). In this essay, we use the software PORTA (Christof and Löbel, 2012) to generate valid inequalities that can be found in a given graph and our aim is to identify and to group them in general formulas of valid inequalities that are already known.

1.3 The Cutting Planes

In this section, we describe the methods for solving general convex and optimization problems whose the QKP. These methods are based on the use of cutting planes that are a clever way of improving bounds. The *cutting planes* are hyperplane that separate the point that we are considering currently from the optimal points. We called the methods cutting planes or *localisation methods*. Several advantages make them attractive (Boyd and Vandenberghe, 2007):

- Cutting planes are performed in decomposition of huge problems into smaller ones that can be solved in parallel and sequentially.

- While the interior point methods require for each iteration an evaluation of objective and constraints, the cutting planes do not require any evaluation.
- Quasi convex problems as well as convex can be solved by cutting planes methods and the differentiability of the objective and the constraint are not required.
- Cutting planes methods solve problems in large and complex number and for a same type of problem, they are faster than an interior point method.

To give better explanation of cutting planes methods, let us define an *integer linear programming* as:

$$\text{maximize } c^T x$$

$$\text{subject to } Ax \leq b$$

where x is a positive integer.

A smallest convex set X that contains all the feasible solutions of an integer programming is called a *convex hull* of the feasible solutions of an integer programming and is defined as follows:

$$conv\{x \in \mathbb{Z}_+^n : Ax \leq b\} \tag{1.3.1}$$

The aim of cutting planes is to find an optimal point for the problem in the convex set X if the latter is not empty.

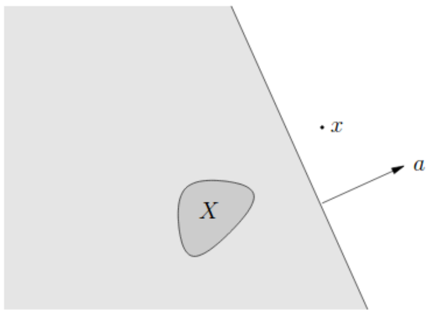

Figure 1.2: The figure illustrates the cutting plane for the point x and the target set X (Boyd and Vandenberghe, 2007).

The separating hyperplane between the point x and the target set X is $a^T z \leq b \quad$ for $z \in X$ that means $a \neq 0$ and b when $a^T x \geq b$. We called that hyperplane a *cut* or a *cutting plane* since it removes the half-space $\{z | a^T z > b\}$ from the search. No point z such that $a^T z > b$ exists in our target, the convex set X. We called a *neutral cutting plane*, the cutting plane in the form of $a^T z = b$ that contains x. The cutting plane in the form of $a^T x > b$ that eliminates the halfspace which x lies in the interior, is called a *deep cut*. A deep cut is better than a neutral cut since it excludes a larger set of points from consideration.

Since the aim of cutting planes is to reduce the feasible set as much as we reach the convex hull that contains the optimal solution, this require to add new cuts or constraints. So some valid inequalities are generated. The process is repeated until a tight upper bound or an optimal solution has been found.

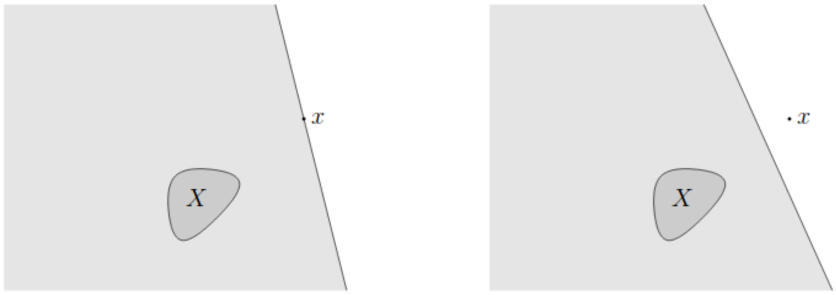

Figure 1.3: The figure illustrates the neutral cut for the point x and the target set X at the left and the deep cut for the point x and the target set X at the right(Boyd and Vandenberghe, 2007).

1.4 Structure of the Essay

- In Chapter 2, we present the linear knapsack problem.
- In Chapter 3, we develop the quadratic knapsack problem, its formulation and we provide proofs of some valid inequalities for the quadratic knapsack problem.
- In Chapter 4, we present the experimentations with the PORTA software package.
- In Chapter 5, we conclude the essay and present future work.

2. The Linear Knapsack Problem

2.1 Definition of the Linear Knapsack Problem

The knapsack problem (KP) was shown to be the 18th most popular out of 75 algorithmic problems and the 4th most needed after kd-trees, suffix trees and the bin packing problem in the 1998 study of the Stony Brook University Algorithm Repository (WikiKP).

The knapsack problem is one of the most important combinatorial optimization problems with many important applications. For instance, a thief who is robbing is exposed to the challenge of choosing items that he has to take among all the possibilities. Of course, he will think of the items that have the most value and considerable weight that he can carry since he cannot carry everything due to the capacity of the bag in which he will put all of the items stolen. This problem is called a *knapsack problem*. Another one that can explain the KP is the cargo air-plane that carries a huge number of items considering his maximal capacity. So a modelization of the problem is required.

Formally, KP can be defined as follows: suppose we have n items x_i, $i \in \{1, \ldots, n\}$, with different weights w_i, where each item has a profit p_i and suppose the knapsack has a limited capacity weight c. The problem is to determine which one of the items we have to choose in such a way that the total profit is maximum and the sum of the products of each item by their respective weights does not exceed the capacity of the knapsack. The KP can be formulated as:

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i \tag{2.1.1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq c \tag{2.1.2}$$

$$x_i \in \{0, 1\}^n \tag{2.1.3}$$

Where (2.1.1) is the maximisation of the objective function. Since we are interested in binary case 0 or 1, this problem is a linear integer programming. When all the decisions variables are integers, we call it *pure integer programming*. When some of the variables are restricted to be integers, it is called *mixed integer programming*. We denote by $KP(w, c)$ the knapsack problem with associated weights $\{w_1, \ldots, w_n\}$, where $w = (w_1, \ldots, w_n)$ and capacity c. We are looking for maximal profit that we can get from the choice of the items.

For example, the problem of capital budgeting is one of integer programming (Beasley, 1996). An enterprise faces a decision problem that consists of deciding which choice of potential investments to make among several numbers of investments. Since the choice of partial investments does not make sense, the decision variables represented by x_i are the binary numbers 0 or 1. When an investment i is accepted, $x_i = 1$, otherwise $x_i = 0$. Assuming that p_i is the profit from an investment i and c_{ij} is the amount j of necessary resource for investment i .

The problem can be formally written as:

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i \tag{2.1.4}$$

$$\text{subject to:} \quad \sum_{i=1}^{n} c_{ij} x_i \leq a_j \; (j = 1, \ldots, m) \tag{2.1.5}$$

$$x_i = 0 \text{ or } x_i = 1, \quad (i = 1, \ldots, n) \tag{2.1.6}$$

Where a_j is the amount of limit resource of each investment. The objective is to maximize the profit from all investments with a limited amount of resource as a constraint.

The Figure 2.1 illustrates a knapsack problem, where we are facing the problem of choosing boxes in such a way that we get the maximum amount of money without exceeding the total weight that is 15kg.

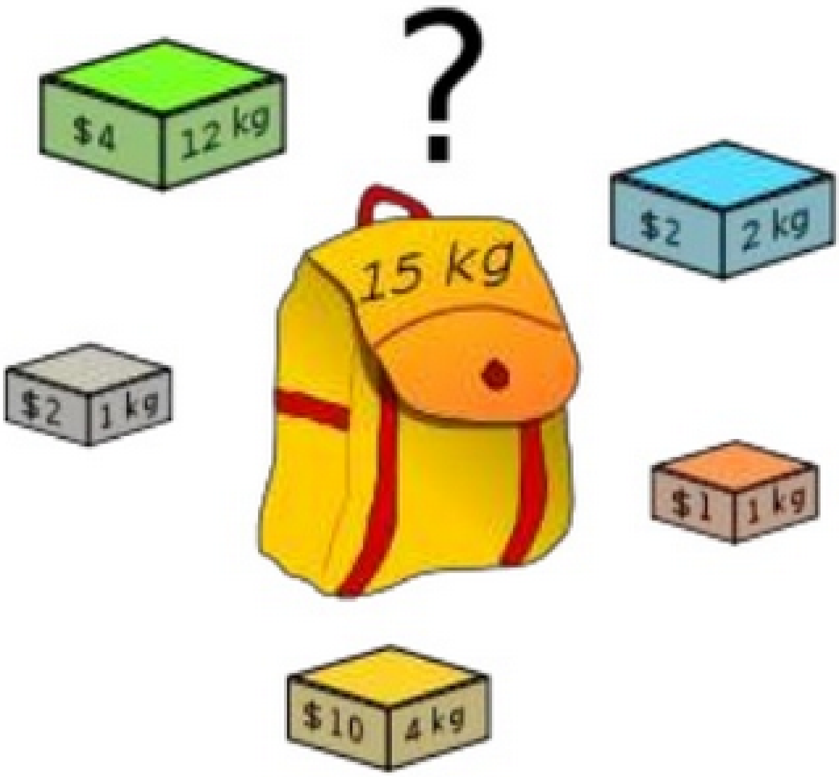

Figure 2.1: knapsack problem (WikiKP).

2.2 Valid Inequalities for the Linear Knapsack Problem

In the next subsections, we are going to look at some particular types of valid inequalities that are useful for the linear KP. Let $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^n$ and $b \in \mathbb{N}$. An inequality of the form $a^T x \leq b$ is said to be a *valid inequality* for a set $X \subseteq \mathbb{R}^n$, if $a^T x \leq b$ for all $x \in X$.

2.2.1 Cover Inequalities.

(Balas, 1975) Consider the inequality

$$\sum_{j \in N} w_j x_j \leq c \tag{2.2.1}$$

where $w_j \in \mathbb{Z}_+^n$ is the vector of weights, $c \in \mathbb{Z}_+$ the knapsack capacity and x_j equal to 0 or 1, for $j \in N = \{1, \ldots, n\}$.

A subset C of N $(C \subseteq N)$ is called a *cover*, if the total weight of its associated items exceeds the weight of the knapsack (Kaparis and Letchford, 2010). That means if the following condition with (2.2.1) is satisfied:

$$\sum_{j \in C} w_j > c \tag{2.2.2}$$

A cover or set covering is known as a special case of the general integer linear programming problem.

So for a cover C, we define a *cover inequality* as:

$$\sum_{j \in C} x_j \leq |C| - 1. \tag{2.2.3}$$

A cover inequality is said to be valid for KP(w, c).

A cover for the inequality (2.2.1) is called *minimal* if

$$\sum_{j \in S} w_j \leq c \quad \text{for all proper subsets } S \text{ of } C. \tag{2.2.4}$$

Since a minimal cover set does not contain any subset, from a minimal cover, we obtain a strong cover inequality.

2.2.2 Extended Cover Inequalities.

(Balas, 1975) Let C be a cover, we define the *extension of the cover* and the *extended cover inequality* respectively by the following:

$$E(C) := C \cup \{j \in N \setminus C : w_j \geq \max_{j \in C} w_j\}; \tag{2.2.5}$$

$$\sum_{j \in E(C)} x_j \leq |C| - 1. \tag{2.2.6}$$

The extended cover inequality is said to be valid for KP(w, c).

2.2.3 Example.

Given a set N of 5 items x_1, x_2, x_3, x_4 and x_5 with respective profits $20, 50, 30, 25$ and 50. Any item has its weight w. Let $w_1 = 10, w_2 = 4, w_3 = 8, w_4 = 12$ and $w_5 = 4$ be their respective weights. The capacity c of the knapsack is equal to 15.

The knapsack problem maximises the objective function (2.1.1) to get the total profit. For our example, we have to maximize $\sum_{i=1}^{5} p_i x_i$.

That means we have to maximize the function

$$20x_1 + 50x_2 + 30x_3 + 25x_4 + 50x_5$$

$$\text{subject to: } \sum_{i=1}^{5} w_i x_i \leq c$$
$$x_i = 0 \text{ or } x_i = 1, \quad (i = 1, \cdots, 5)$$

So our constraint is given by

$$10x_1 + 4x_2 + 8x_3 + 12x_4 + 4x_5 \leq 15.$$

We can choose a cover set as long as the condition (2.2.2) is satisfied. We are maximizing to get the best solution. Therefore, to get that solution, we need to find the best inequality i.e. the strong inequality, which derived from a minimal cover set. So, we have to find a minimal cover set in order to attain that solution. Let us choose the set $\{1, 3, 5\}$, this is a cover because the condition (2.2.2) is verified since $10+8+4 > 15$. However, it is not a minimal cover because, from it, we can get the subset $\{1, 3\}$ that still satisfies the condition of a cover. We are no longer interested in the cover $\{1, 3, 5\}$ because it is not minimal. Let us choose a minimal one, we can consider the minimal cover $\{1, 3\}$ but let us take another one, let say $\{1, 2, 5\}$. The condition $\sum_{i \in C} w_i > c$ is satisfied since $10 + 4 + 4 > 15$. The corresponding cover inequality is

$$\sum_{i \in C} x_i \leq |C| - 1$$
$$x_1 + x_2 + x_5 \leq 3 - 1$$
$$x_1 + x_2 + x_5 \leq 2.$$

The cardinality of the cover $|C|$, is 3 and the formula of the cover inequality is (2.2.3). Then to find the extended cover $E(C)$, we add to the set $C = \{1, 2, 5\}$ an item in $N \setminus C$ such that it has the weight which exceeds or is equal to the maximum weight of an item in the cover C. Therefore, we have to choose between the items 3 and 4 the one that the weight exceeds or equals to 10, the maximal weight in C. That item is 4 since the weight of 4 is 12. $E(C) = \{1, 2, 4, 5\}$ and the extended cover inequality corresponding to it, is $x_1 + x_2 + x_4 + x_5 \leq 2$.

3. The Quadratic Knapsack Problem

The knapsack problem is a decision problem that consists of deciding which choice of items to make in the case where there are no items that depend on one another. There exists another combinatorial optimisation problem that consider the relations between pair of items. It is called, the quadratic knapsack problem QKP.

3.1 Definition of the Quadratic Knapsack Problem

The quadratic knapsack problem is a generalisation of the knapsack problem with a quadratic objective function. In real life, there are many applications of the quadratic knapsack problem such as the selection of a portfolio optimisation in the finance area, the location of airports, of railways stations in engineering and many other.

More formally, the QKP is defined as follows: Given a number of items n, positive integer weights w_i for each item x_i, a knapsack capacity c (that is the maximal weight that can be contained by the knapsack) and a non-negative integer $n \times n$ profit matrix $P = (p_{ij})$, where p_{ii} is the profit hailed from the selection of the same item x_i and $p_{ij} + p_{ji}$, the profit when both x_i and x_j items have been selected. The QKP can be formulated as follows:

$$\text{maximize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} x_i x_j \tag{3.1.1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq c \tag{3.1.2}$$

$$x_i \in \{0,1\} \quad \text{for } i \in N = \{1, \dots, n\}. \tag{3.1.3}$$

The linearisation of the quadratic knapsack problem is obtained by using some constraints and new variables y_{ij} that represent the product of the items x_i and x_j, where the variable $x_i = x_i^2$ represents the product of the item x_i by itself, with $x_i \in \{0,1\}$, then it can be reformulated as follows:

$$\text{maximize} \quad \sum_{i=1}^{n} p_{ii} x_i + \sum_{i,j=1}^{n} (p_{ij} + p_{ji}) y_{ij} \tag{3.1.4}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq c \tag{3.1.5}$$

$$y_{ij} \leq x_i \tag{3.1.6}$$

$$y_{ij} \leq x_j \tag{3.1.7}$$

$$x_i + x_j \leq y_{ij} + 1 \tag{3.1.8}$$

$$x_i, y_{ij} \in \{0,1\}. \tag{3.1.9}$$

The constraint (3.1.8) forces y_{ij} to be 1 in the case where both items x_i and x_j are 1. Without loss of generality, we may assume:

$$\max_{i \in N} \, w_i \leq c < \sum_{i \in N} w_i \, . \tag{3.1.10}$$

If the weight of an item x_i is greater than the knapsack capacity, that item is not selected. We have $x_i = 0$. If the weight of an item x_i is less than the knapsack capacity, that item can be selected. We have $x_i = 1$. Another possibility is when the weight of the item is equal to the capacity of the knapsack, in that case we may decompose the problem. Then in order to satisfy the constraint $\sum_{i \in N} w_i x_i \leq c$, we must satisfy the condition $0 \leq w_i < c$. The profit gotten from the selection is non negative and the matrix $P = \{p_{ij}\}$ is therefore a non negative integer matrix for all $i < j$ (Pisinger, 2007).

Given a subset Q of V. It is easy to notice that QKP is a generalisation of the clique problem. This problem is based on selecting a subset Q of vertices. To model it as a QKP, we need to define its cardinality, let us denote it by h since the cardinality of the set of vertices is $|V| = n$. The knapsack capacity is given by h and the weight $w_j = 1$ for $j \in N$. The profit of selecting item i and item j is symmetric and equal to 1 $(p_{ij} = p_{ji} = 1)$, if $(i,j) \in E$, otherwise, there is no profit $(p_{ij} = 0)$.

The QKP is also a generalisation of the 0-1knapsack problem as previously mentioned.

The QKP is not only a generalisation of the clique problem and the 0-1knapsack problem but it is also known as a constrained version of the quadratic 0-1programming problem (QP). The latter problem is defined as a QKP but without the capacity constraint. Therefore, all the solutions of the QKP are included in the solutions of QP. All the valid inequalities for the QP are also valid for QKP. The valid inequalities can be used to tighten the bounds for the QKP (Pisinger, 2007).

An *upper plane* is defined as a linear function f satisfying for any feasible solution x of (3.1.1) the condition

$$f(x) \geq \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j.$$

A family of upper bounds that are based on upper was invented by Gallo (Kellerer et al., 2004).

In branch and bound algorithm that upper bound can be used to speed up the search (Pisinger et al., 2007). Several valid inequalities for the QKP are discussed.

3.2 Valid Inequalities for the Quadratic Knapsack Problem

There exist several methods of generating valid inequalities such as logical cuts method, Chvatal-Gomory cuts method and cover inequalities method. In this essay, we will use the method of cover inequalities to generate the valid inequalities for the QKP. Let us recall the definition of a connected graph.

A *graph* $G = (V, E)$ is said to be *connected*, if there exists a path between any two vertices i and j of the set of vertices V.

For each vertex $i \in V$, let $H(i) = \{j \in V : (i,j) \in E\}$ define the neighbourhood of the vertex i and let $d(i) = |H(i)|$ represents the degree of the vertex i.

3.2.1 Tree Inequalities.

(Rader, 1997) A connected subgraph of a graph G that does not contain cycles is called a *tree*. We denote it by T. And a *spanning tree* of an undirected graph G is a tree T that contains all the vertices of G but has a minimum possible number of edges.

3.2.2 Theorem.

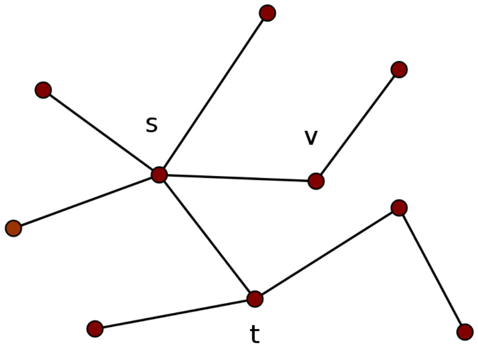

Figure 3.1: The figure illustrates a tree. The degree of the vertex s is 5, the degree of t is 3 and the degree of v is 2.

Let a subset C of a set of vertices V ($C \subseteq V$) be a cover and suppose the subgraph $G' = (C, E(C))$ is connected. Let T be a spanning tree of G'. The inequality (3.2.1) is valid for the quadratic knapsack problem:

$$\sum_{(i,j)\in T} y_{ij} \leq \sum_{i\in C}(d_T(i) - 1)x_i. \tag{3.2.1}$$

Proof. (Rader, 1997) Let Q be a subset of V ($Q \subseteq V$). A feasible solution of QKP can be defined as follows:

$$x_i = \begin{cases} 1 & \text{if } i \in Q \\ 0, & \text{otherwise} \end{cases} \tag{3.2.2}$$

$$y_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E(Q) \\ 0, & \text{otherwise.} \end{cases} \tag{3.2.3}$$

Let $F = E(Q) \cap T$. Then F is a collection of edges in $E(Q)$ and T simultaneously. Let T' be a subset of F ($T' \subseteq F$) in one component. That means that pair of vertices of T' are connected to each other by paths. If Q_0 is the set of vertices of T', $Q_0 = V(T')$,

$$|T'| = \sum_{(i,j)\in T'} y_{ij} \tag{3.2.4}$$

$$= |Q_0| - 1 \tag{3.2.5}$$

$$= (|Q_0| - 2) + 1. \tag{3.2.6}$$

We get (3.2.5) since the number of edges in a tree is less than the number of its vertices by one unit.

Moreover, for each vertex i in Q_0, the degree of i is defined by: $d'_T(i) = |\{j : (i,j) \in T'\}|$ we have $\sum_{i\in Q_0} d'_T(i) - 1 = |Q_0| - 2$. So,

$$|T'| = \left(\sum_{i\in Q_0} d'_T(i) - 1\right) + 1. \tag{3.2.7}$$

Since C is a cover and $Q_0 = V(T')$, we have $|Q_0| < |C|$. Thus, for all $j \in Q_0$, $d_T(j) > d'_T(j)$.

Then $|T'| = (\sum_{i \in Q_0} d'_T(i) - 1) + 1$, which implies that

$$|T'| < (\sum_{i \in Q_0} d_T(i) - 1) + 1 \quad (3.2.8)$$

$$\leq (\sum_{i \in Q_0} d_T(i) - 1). \quad (3.2.9)$$

But

$(\sum_{i \in Q_0} d_T(i) - 1) = (\sum_{i \in Q_0} d_T(i) - 1)x_i$ because $x_i = 1$ for $i \in Q_0$.

So

$$|T'| \leq \sum_{i \in Q_0} (d_T(i) - 1)x_i. \quad (3.2.10)$$

Hence the validity of the tree inequality is proved. The corresponding figure is the Figure 3.1. □

3.2.3 Star Inequalities.

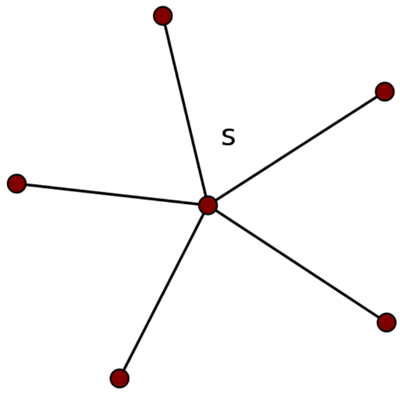

Figure 3.2: The figure illustrates a star.

(Rader, 1997) A tree T is called a *star*, if each edge (i, j) of the tree is adjacent to a given vertex i of the set of vertices. Let $Q = \{i_1, \cdots, i_q\} \subseteq V$. If Q is not a cover, $C = Q \cup \{t\}$ is a (1,h)-configuration, where $q \geq h$. For all subset $Q' \subseteq Q$ we define $Q' \cup \{t\}$ a cover such that $|Q'| \geq h$.

3.2.4 Theorem. *Let $C = Q \cup \{t\}$ be a $(1, h)$-configuration and suppose that a subgraph G' of G $G' = (C, E(C))$ is connected. Let a tree T be a spanning star centered at t on C. The inequality (3.2.11) is valid for QKP:*

$$\sum_{j \in Q} y_{jt} \leq (h - 1)x_t. \quad (3.2.11)$$

Proof. (Rader, 1997) Let us show that (3.2.11) is satisfied. Let consider a feasible point (X, Y) of QKP. If $x_t = 0$, it is satisfied obviously.

If $x_t = 1$, the inequality (3.2.11) which is the same as

$$x_t \sum_{j \in Q} x_j \leq (h - 1)x_t$$

becomes

$$\sum_{j \in Q} x_j \leq (h-1).$$

It is verified because at most $h-1$ elements of Q have value 1. A star inequality corresponds to the Figure (3.2).

□

3.2.5 Double Star Inequalities.

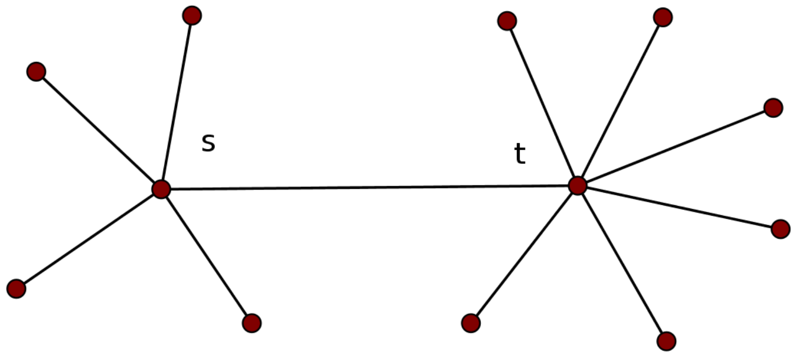

Figure 3.3: Illustration of a double-star.

The general formula of a double star inequality is given by the following inequality:

$$\sum_{i \in S} y_{is} + \sum_{i \in T} y_{it} \leq (|S|)x_s + (|T|)x_t + (b - 2 - |S| - |T|)y_{st} \tag{3.2.12}$$

Proof. (Djeumou Fomeni et al., 2015) Let $\alpha^T x \leq \beta$ with α and β both positive. Let s and t be any two distinct nodes in V the set of nodes. Let S and T be any two disjoints sets. They are in V and do not contain the two nodes s and t. And let $\alpha(S)$ respectively $\alpha(T)$, denote $\sum_{i \in S} \alpha_i$ and $\sum_{i \in T} \alpha_i$ respectively. Let denote by μ, the quantity

$$\sum_{i \in S} \alpha_i y_{is} + \sum_{i \in T} \alpha_i y_{it} \tag{3.2.13}$$

If $x_s = x_t = 0, \mu = 0$.

If $x_s = 1$ but $x_t = 0$, μ cannot exceed the quantity

$$\min \{\alpha(S), \beta - \alpha_s\}.$$

We call this quantity λ_s.

If $x_s = 0$ but $x_t = 1$, μ cannot exceed the quantity

$$\min \{\alpha(T), \beta - \alpha_t\}.$$

We call this quantity λ_t.

Finally, if $x_s = x_t = 1$, the quantity μ cannot exceed

$$\beta - \alpha_s - \alpha_t.$$

We call this quantity $\lambda_s t$.

Then from that we have the inequality

$$\mu \leq \lambda_s x_s + \lambda_t x_t + (\lambda_s t - \lambda_s - \lambda_t) y_{st}.$$

That inequality is called a double star inequality and the Figure 3.3 is its corresponding figure. □

4. Experimentation with PORTA

4.1 Definitions

Recall the definition of a convex hull (1.3.1). In n-dimensional Euclidean space $\mathbb{R}^n$, the convex hull of m points $x_i \in \mathbb{R}^n$, $i = \{1, \dots, m\}$ that is the intersection of all convex sets containing these points is defined to be a *polytope* or a *convex polytope* (Schäfer, January 10, 2012) (Readdy, 2013). A polytope is a geometric object that has flat sides and exist in any general number of dimensions n. Given a polytope P in $\mathbb{R}^n$ with supporting hyperplane H, and given H_+ and H_-, the half open regions determined by the hyperplan H, we say that $P \cap H$ is a *face* if $P \cap H \neq \emptyset$, $P \cap H_+ \neq \emptyset$ and $P \cup H_- = \emptyset$. Then:

- A 0-dimensional face is a vertex.
- A 1-dimensional face is an edge.
- A 2-dimensional polytope is a polygon.
- A 3-dimensional polytope is a polyhedron.

A face of a polytope is a polytope in its own right (Readdy, 2013). An n-dimensional polytope P is bounded by a number of $(n-1)$-dimensional facets. These facets are themselves polytopes, whose facets are $(n-2)$-dimensional ridges of the polytope P. A *polygon* is a figure bounded by a finite straight line segments called edges. It has vertices that are the intersections of the segments.

A *polyhedron* $S \in \mathbb{R}^n$ is defined as a set of intersections of a finite set of closed halfspaces. It is formulated as follows:

$$S = \{x \in \mathbb{R}^n | Ax \leq b\} \tag{4.1.1}$$

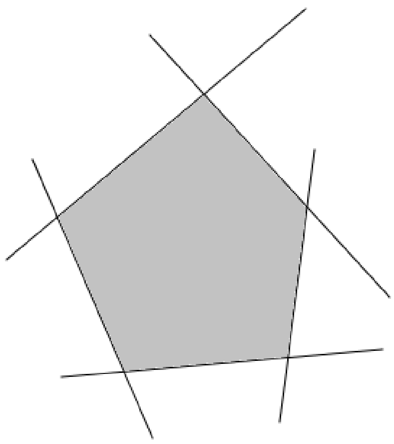

Figure 4.1: The figure illustrates a polyhedron.

A *0-1 knapsack polytope* is a polytope of the form

$$KP(w_i, c) = conv\{x \in \{0,1\}^n : w_i x \leq c\}$$

where $w_i \in \mathbb{Z}_+^n$ is the vector of weights and $c \in \mathbb{Z}_+$ is the knapsack capacity. The 0-1 knapsack polytope is the convex hull of feasible solutions to a 0-1 knapsack problem.

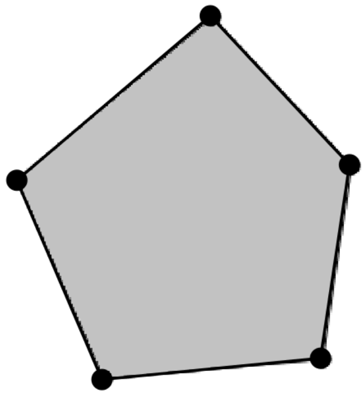

Figure 4.2: The figure illustrates the faces of a polytope. The faces are in bold.

A convex hull of vertices of a polytope can be represented by the following figure

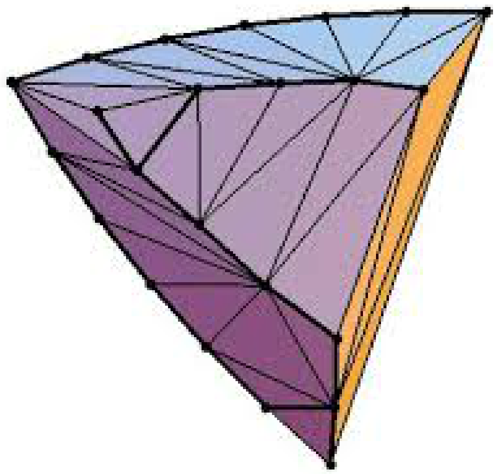

Figure 4.3: The figure illustrates a convex hull in 3 dimension.

4.2 PORTA Software

The main work of this chapter is to use the PORTA software to simulate the valid inequalities that we have studied in the previous chapters.

Given a graph $G = (V, E(V))$ with a set of vertices V and edges $E(V)$. If x_i and x_j are vertices of V, they can be linked by an edge y_{ij} of $E(V)$ such that $y_{ij} = x_i x_j$, G is a spanning tree defined as an undirected graph in which all the vertices are connected to each other. To get the inequalities from PORTA, we define the dimension DIM of the graph that is the number n of vertices add the number of the edges corresponding to the graph. The input of PORTA is:

- The dimension of the graph, DIM.

- The matrix defines all the possible cases to choose or not an item. The total number of the possibilities is 2^n, where n is the number of vertices.
- The matrix given to PORTA has DIM as the number of columns and 2^n the number of rows.

PORTA computes the input and generates a system of inequalities. From that, we have to identify the inequalities and group them in a way that each group satisfies the same conditions. The final step is to find their general form.

4.3 Experimentation

a) Consider the following graph G :

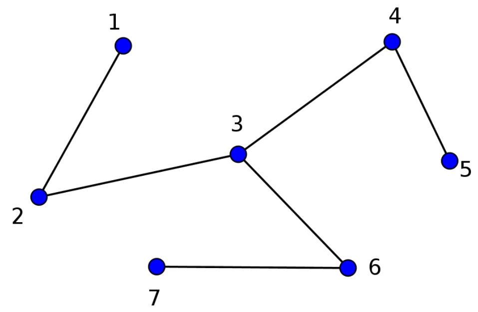

Figure 4.4: The figure illustrates a graph of 7 vertices and 6 edges.

For the given graph $G = (V, E(V))$, the set of vertices V contains 7 vertices and the set of edges $E(V)$ contains 6 edges. That represents 7 items x_i and each of the items can take the value 0 when the item is rejected or 1 when it is accepted. The edge or link between two nodes i and j is denoted y_{ij} which represents the product of the items x_i and x_j. If both items are rejected, y_{ij} has the value 0, this is the same as if one of them is rejected. If all of them are accepted the value of y_{ij} is 1.

Input

To get inequalities of the graph from PORTA, we give as input the matrix associated to the graph and its dimension. The dimension of the matrix is

$$m = DIM = |V| + |E(V)|$$
$$= 7 + 6$$
$$= 13$$

The matrix is $m \times 2^{|V|}$, for this case it is 13×2^7.

Output

As output, we get a system of inequalities, for this case we get especially 24 inequalities that we have to identify and group within the condition they satisfy. That give the general formula of each group of inequalities.

The first group is formed by valid inequalities for QKP that satisfy the condition $y_{ij} \geq 0$.

(1)	-y12						<=	0
(2)		-y23					<=	0
(3)			-y34				<=	0
(4)				-y45			<=	0
(5)					-y36		<=	0
(6)						-y67	<=	0

This condition states that the product y_{ij} stays positive whenever the items x_i and x_j are rejected or accepted.

The next group of inequalities satisfy the condition $x_i \geq y_{ij}$ for QKP.

(7)	-x1					+y12						<=	0
(8)		-x2					+y23					<=	0
(10)			-x3							+y36		<=	0
(11)			-x3					+y34				<=	0
(13)				-x4					+y45			<=	0
(16)					-x6						+y67	<=	0

The condition is that they satisfy the state regardless of the value of the item x_j. The product y_{ij} will always be less or equal to x_i since both items are binary variables.

The following block of inequalities is almost the same as the previous one, the only difference is that the condition for QKP that its inequalities satisfy is $x_j \geq y_{ij}$.

(9)	-x2						+y12						<=	0
(12)		-x3						+y23					<=	0
(14)			-x4						+y34				<=	0
(15)				-x5						+y45			<=	0
(17)					-x6						+y36		<=	0
(18)						-x7						+y67	<=	0

Regardless of the value of the item x_i, the product y_{ij} will always be less than or equal to the item x_j.

In the last block, we get valid inequalities for QKP that satisfy the condition $x_i + x_j \leq y_{ij} + 1$

(19)						+x6	+x7						-y67	<=	1
(20)				+x4	+x5						-y45			<=	1
(21)			+x3			+x6						-y36		<=	1
(22)			+x3	+x4						-y34				<=	1
(23)		+x2	+x3						-y23					<=	1
(24)	+x1	+x2						-y12						<=	1

That condition means that whenever the items x_i and x_j are selected or not, their sum will always be less or equal to their product add to 1.

- $x_i = 0$ and $x_j = 0$

  $x_i + x_j = 0$

  $x_i + x_j \leq y_{ij} + 1$ is satisfied.

- $x_i = 1$ and $x_j = 0$, or

  $x_i = 0$ and $x_j = 1$

$x_i + x_j = 1$

$x_i + x_j \leq y_{ij} + 1$ is satisfied.

- $x_i = 1$ and $x_j = 1$

  $x_i + x_j = 2$

  $x_i + x_j \leq y_{ij} + 1$ is also satisfied.

b) Now let us study another graph $G = (V, E(V))$ in the form of:

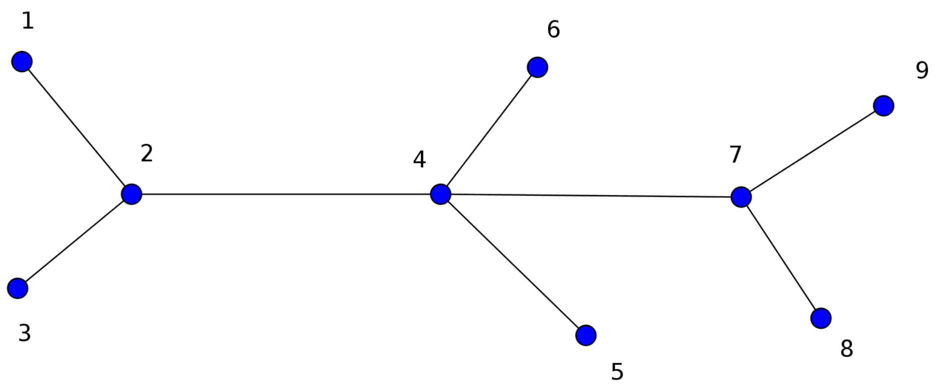

Figure 4.5: The figure illustrates a graph of 9 vertices and 8 edges.

For that graph, the set of vertices V contains 9 vertices and the set of edges $E(V)$ contains 8 edges. So we have 9 items x_i and 8 edges.

Input

Our input is the matrix associated to the graph and its dimension. The dimension of the matrix is

$$m = DIM = 9 + 8$$

$$= 17$$

The matrix is $m \times 2^{|V|}$, for this case it is 17×2^9.

Output

The output gives us a system of 32 inequalities. We study each of the inequalities and we group together those who satisfy the same condition. We group them in four blocks that are constituted by all the types of constraints we found in the previous case: $y_{ij} \geq 0$, $x_i \geq y_{ij}$, $x_j \geq y_{ij}$ and $x_i + x_j \leq y_{ij} + 1$. But we get more inequalities for each group than previously due to the number of edges. Previously, we had 6 inequalities per group because the graph had 6 edges, now we have 8 inequalities per group since the number of edges of the graph is 8 and from that, we get the 32 inequalities.

Now, let us consider the extended cover inequality by varying the value of α and observe what kind of inequalities we can have. Recall the extension of a cover:

$$E(C) := C \cup \{j \in N \setminus C : w_j \geq \max_{j \in C} w_j\}. \quad (4.3.1)$$

The corresponding extended cover inequality is

$$\sum_{j \in E(C)} x_j \leq |C| - 1. \tag{4.3.2}$$

- Consider $\alpha \leq 8$, besides the 32 inequalities, we get the following inequality:

```
-2x2   -3x4       -2x7       +y12+y23+y24+y45+y46+y47+y78+y79 <= 0.
```

  That inequality satisfies the condition of a tree inequality and the general formula of it, is

$$\sum_{(i,j) \in T} y_{ij} \leq \sum_{i \in C} (d_T(i) - 1) x_i.$$

- For $\alpha \leq 7$, we get more than 32 inequalities, since we get 59 inequalities. The new group of inequalities added to the 32 inequalities, is the following:

```
( 25)     -2x2    -2x4       -2x7          + y12+ y23+ y24     +y46+ y47+ y78+ y79 <= 0
( 26)     -2x2    -2x4       -2x7          + y12+ y23+ y24+y45     + y47+ y78+ y79 <= 0
( 27)     -2x2    -3x4       - x7          + y12+ y23+ y24+y45+y46+ y47      + y79 <= 0
( 28)     -2x2    -3x4       - x7          + y12+ y23+ y24+y45+y46+ y47+ y78       <= 0
( 29)     - x2    -3x4       -2x7               + y23+ y24+y45+y46+ y47+ y78+ y79 <= 0
( 30)     - x2    -3x4       -2x7          + y12      + y24+y45+y46+ y47+ y78+ y79 <= 0
```

  The general formula for those inequalities is a general form of a tree inequality written as:

$$\sum_{(i,j) \in T} y_{ij} \leq \sum_{i \in C} (d_T(i) - 1) x_i.$$

  We get another inequality that satisfies the condition of an extended cover inequality,

```
( 57) + x1+ x2+ x3+ x4+x5+x6+ x7+ x8+ x9                                      <= 7.
```

  The general formula is

$$\sum_{j \in E(C)} x_j \leq |C| - 1.$$

```
( 40)     -2x2    -2x4       + x7+ x8+ x9+ y12+ y23+ y24+y45+y46     - y78       <= 2
( 45) + x1+2x2+ x3-2x4       -2x7          - y12- y23- y24+y45+y46+ y47+ y78+ y79 <= 2

( 48)     - x2    + x4+x5+x6- x7          + y12+ y23     -y45          + y78+ y79 <= 3
( 49)     - x2    + x4+x5+x6- x7          + y12+ y23          -y46     + y78+ y79 <= 3
( 55)     - x2    + x4+x5+x6+ x7+ x8+ x9+ y12+ y23               - y47           <= 5
( 56) + x1+ x2+ x3+ x4+x5+x6- x7                     - y24               + y78+ y79 <= 5
( 58)     -3x2    - x4+x5+x6+3x7+2x8+2x9+2y12+2y23+ y24+y45+y46- y47- y78- y79 <= 7
( 59) +2x1+3x2+2x3- x4+x5+x6-3x7          - y12- y23- y24+y45+y46+ y47+2y78+2y79 <= 7

( 52) + x1    + x3-2x4            + x8+ x9          + y24+y45+y46+ y47            <= 4
( 53)     - x2          +x5+x6+ x7+ x8+ x9+ y12+ y23+ y24                         <= 5
( 54) + x1+ x2+ x3    +x5+x6- x7                                  + y47+ y78+ y79 <= 5
```

New inequalities

With PORTA, we get new inequalities that are unkown, where the coefficients of y_{ij} are 2:

```
( 31)      -2x2    -4x4      -2x7        + y12+ y23+2y24+y45+y46+2y47+ y78+ y79 <= 0
( 46)      -2x2    -3x4          + x8+ x9+ y12+ y23+2y24+y45+y46+ y47          <= 2
( 47) + x1     + x3-3x4      -2x7                    + y24+y45+y46+2y47+ y78+ y79 <= 2
( 58)      -3x2    - x4+x5+x6+3x7+2x8+2x9+2y12+2y23+ y24+y45+y46- y47- y78- y79 <= 7
( 59) +2x1+3x2+2x3- x4+x5+x6-3x7        - y12- y23- y24+y45+y46+ y47+2y78+2y79 <= 7.
```

c) Let us consider a graph of 10 vertices:

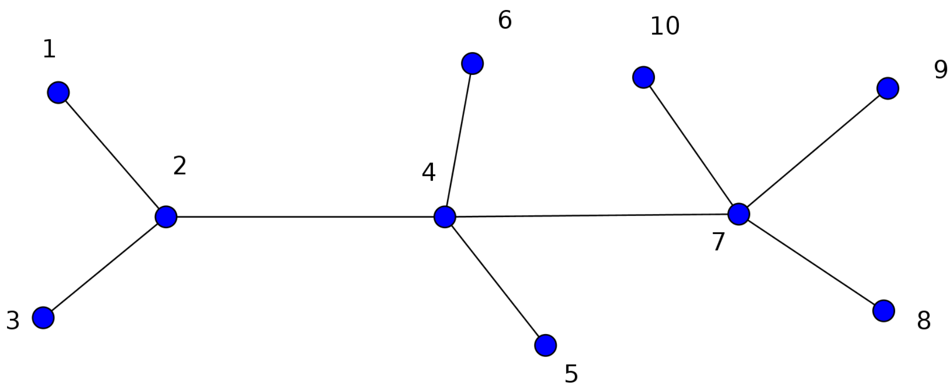

Figure 4.6: The figure illustrates a graph of 10 vertices and 9 edges.

For the given graph $G = (V, E(V))$, the set of vertices V contains 10 vertices and the set of edges $E(V)$ contains 9 edges. So we have 10 items x_i. If x_i and x_j items are accepted, the value of y_{ij} is 1, otherwise it is 0.

Input

We give as input to PORTA, the matrix associated to the graph and its dimension. The dimension of the matrix is

$$\begin{aligned} m = DIM &= |V| + |E(V)| \\ &= 10 + 9 \\ &= 19. \end{aligned}$$

The matrix is $m \times 2^{|V|}$, for this case it is 19×2^{10}.

Output

The output gives us a system of 36 inequalities. We study those inequalities as we did for the previous case, we group them in four blocks that are constituted by all the types of constraints. We get more inequalities that explained why we got the 36 inequalities.

- This following type of inequalities, is when $\alpha \leq 9$, for this block, one inequality is added to the 36 we got.

  ```
  -2x2    -3x4        -3x7            +y12+y23+y24+y45+y46+y47+y78+y79+y710 <= 0.
  ```

  That inequality is in the form of a tree inequality that satisfy

$$\sum_{(i,j) \in T} y_{ij} \leq \sum_{i \in C} (d_T(i) - 1) x_i.$$

- For $\alpha \leq 8$, we get more inequalities.

  This block is constituted by tree inequalities as the ones we get for $\alpha \leq 9$

$$\sum_{(i,j)\in T} y_{ij} \leq \sum_{i\in C}(d_T(i)-1)x_i.$$

```
( 28)    -2x2    -3x4     -2x7          +y12+y23+y24+y45+y46+y47     +y79+y710 <= 0
( 29)    -2x2    -3x4     -2x7          +y12+y23+y24+y45+y46+y47+y78     +y710 <= 0
( 30)    -2x2    -3x4     -2x7          +y12+y23+y24+y45+y46+y47+y78+y79       <= 0
( 33)    - x2    -3x4     -3x7              +y23+y24+y45+y46+y47+y78+y79+y710 <= 0
( 34)    - x2    -3x4     -3x7          +y12    +y24+y45+y46+y47+y78+y79+y710 <= 0
```

  We have also this form of inequalities

```
( 45) +x1+ x2+ x3-2x4     -3x7           -y12        +y45+y46+y47+y78+y79+y710 <= 2
( 46) +x1+ x2+ x3-2x4     -3x7               -y23    +y45+y46+y47+y78+y79+y710 <= 2
( 47) +x1+2x2+ x3-2x4     -3x7           -y12-y23-y24+y45+y46+y47+y78+y79+y710 <= 2
```

  We also get the following type of inequalities:

```
( 49)    - x2    + x4+x5+x6-2x7            +y12+y23   -y45          +y78+y79+y710 <= 3
( 50)    - x2    + x4+x5+x6-2x7            +y12+y23       -y46      +y78+y79+y710 <= 3
( 56)    -2x2    -2x4      +3x7+x8+x9+x10 +y12+y23+y24+y45+y46-y47-y78-y79-y710 <= 3
```

  Another type of inequality, given by the following block is also found:

```
( 58) +x1    +x3 -2x4          +x8+x9+x10         +y24+y45+y46+y47                   <= 5
( 59) +x1 +x2+x3     +x5+x6-2x7                                +y47+y78+y79+y710 <= 5
( 60) +x1 +x2+x3 + x4+x5+x6-2x7                   -y24                +y78+y79+y710 <= 5

( 61)     -x2        +x5+x6+ x7+x8+x9+x10 +y12+y23+y24                              <= 6
( 62)     -x2    + x4+x5+x6+ x7+x8+x9+x10 +y12+y23            -y47                  <= 6
```

- For this case $\alpha \leq 7$

  This following block is formed by tree inequalities, we mention that they satisfy the condition:

$$\sum_{(i,j)\in T} y_{ij} \leq \sum_{i\in C}(d_T(i)-1)x_i$$

```
( 28)    -2x2    -2x4     -2x7              +y12+y23+y24    +y46+y47     +y79+y710 <= 0
( 29)    -2x2    -2x4     -2x7              +y12+y23+y24    +y46+y47+y78     +y710 <= 0
( 38)    - x2    -3x4     -2x7                  +y23+y24+y45+y46+y47     +y79+y710 <= 0
( 39)    - x2    -3x4     -2x7                  +y23+y24+y45+y46+y47+y78     +y710 <= 0
( 48)            -3x4     -3x7                      +y24+y45+y46+y47+y78+y79+y710 <= 0
```

  We also get an extended cover inequality which general formula is

$$\sum_{j\in E(C)} x_j \leq |C|-1.$$

```
( 65) + x1+ x2+ x3+ x4+x5+x6+ x7+ x8+ x9+x10      <= 8.
```

New inequalities

a) The first type of new inequalities is a type of inequalities that have 2 as a coefficient of the y_{ij}, we have:

```
( 35)    -2x2    -4x4     -3x7            + y12+ y23+2y24+ y45+y46+2y47+y78+ y79+ y710 <= 0
( 48) +x1    + x3-3x4     -3x7                      + y24+ y45+y46+2y47+y78+ y79+ y710 <= 2
( 49)    -2x2    -3x4     -3x7            + y12+ y23+2y24     +y46+2y47+y78+ y79+ y710 <= 0
( 50)    -2x2    -3x4     -3x7            + y12+ y23+2y24+ y45    +2y47+y78+ y79+ y710 <= 0
( 51)    -2x2    -4x4     -2x7            + y12+ y23+2y24+ y45+y46+2y47    + y79+ y710 <= 0
( 63)    -4x2    -4x4     +3x7+2x8+2x9+2x10+2y12+2y23+2y24+2y45+2y46   -y78- y79- y710 <= 6
( 64)+2x1+3x2+2x3 -x4+x5+x6-5x7           - y12- y23- y24+ y45+y46+y47+2y78+2y79+2y710 <= 7
( 66)     -3x2  - x4+x5+x6+4x7+2x8+2x9+2x10+2y12+2y23+y24+ y45+ y46-y47- y78- y79-y710 <= 9.
```

b) The more we get inequalities, the more complicated inequalities appear. We find a new type of inequality with 3 as coefficients of the y_{ij}:

```
( 56)   -2x2    -5x4     -3x7           + y12+ y23+3y24+ y45+ y46+3y47+ y78+ y79+ y710 <= 0.
```

5. Conclusion

In this project, we have considered the cover inequalities method of generating cuts. We have simulated the valid inequalities generated from cover inequalities by using the software package PORTA. We were able to give the general formulas for some of the valid inequalities and we observed the following:

- With the cover inequality, we got for each graph considered different inequalities that satisfy all the constraints for the QKP such as defined in Chapter 3.
- If we vary the value of alpha by decreasing its value by one, here $\alpha = |C| - 1$. Some inequalities are added such as the tree inequalities.
- The more the value of alpha was decreased, the more inequalities were found. We were able to identify the extended cover inequality.
- We were able to identify the new inequalities unknown until now for which the coefficients of the y_{ij} are greater than one, where there are 2 or 3.

For instance, for the graph of 10 vertices, when the value of $\alpha \leq 10$, we got 36 inequalities, with $\alpha \leq 9$, we got 37, with $\alpha \leq 8$, we got 66, with $\alpha \leq 7$ we got 550, with $\alpha \leq 6$, we got 3262 and for $\alpha \leq 5$, we got 6121 inequalities.

5.1 Future work

The future work will concern of finding the general formulas of the new types of inequalities in which appear the coefficients 2 and 3.

Acknowledgements

I give all thanks to God the almighty, for his infinite blessings and his incomparable love that he has showed me each and every day. I sincerely thank my supervisors Dr Trivikram Dokka Venkata Satyanaraya and Dr Franklin Djeumou Fomeni who made this work a success through their commitment, advice, encouragement and priceless help. I thank the founder of AIMS Prof. Neil Turok for the amazing idea of this institution for African students (AIMS). I thank the AIMS Director Prof. Barry Green and the Academic Director Prof. Jeff Sanders for the advice and the unconditional support. I thank all of my tutors for their help, especially Andry Nirina Rabenantoandro who has been a great help to accomplish this work. I thank all of my classmates and all of my friends who have made my stay at AIMS unforgettable. I cannot end without expressing my deep gratitude to my family who have supported me with their love, advice and prayers every day. To my father, my mother and my siblings, I dedicate my essay.

References

E. Balas. Facets of the knapsack polytope. *Mathematical programming*, 8(1):146–164, 1975.

J. Beasley. Or-notes. *Accessed March*, 22, 1996.

S. Boyd and L. Vandenberghe. Localization and cutting-plane methods. *From Stanford EE 364b lecture notes*, 2007.

A. Caprara, D. Pisinger, and P. Toth. Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11(2):125–137, 1999.

CH. Convex polytopes. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/convex+polytopes, Accessed April 2012.

T. Christof and A. Löbel. Porta–polyhedron representation transformation algorithm, 2008. *Download available at http://typo. zib. de/opt-long_projects/Software/Porta*, 2012.

F. Djeumou Fomeni, K. Kaparis, and A. Letchford. Cutting planes for rlt relaxations of mixed 0-1 polynomial programs. *Mathematical Programming*, 151(2):639–658, 2015.

F. D. Fomeni. A new family of facet defining inequalities for the maximum edge-weighted clique problem. *Optimization Letters*, pages 1–8, 2015.

G. Gallo, P. L. Hammer, and B. Simeone. Quadratic knapsack problems. In *Combinatorial optimization*, pages 132–149. Springer, 1980.

google. optimization. Wikipedia, the Free Encyclopedia, https://www.google.com/google/optimization, Accessed April 2012.

A. Govan. Introduction to optimization. In *North Carolina State University, SAMSI NDHS, Undergraduate workshop*, 2006.

K. Kaparis and A. N. Letchford. Cover inequalities. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

H. Kellerer, U. Pferschy, and D. Pisinger. *Introduction to NP-Completeness of knapsack problems*. Springer, 2004.

R. Mansini and M. G. Speranza. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, 114(2):219–233, 1999.

R. G. Parker and R. L. Rardin. *Discrete optimization*. Elsevier, 2014.

D. Pisinger. The quadratic knapsack problem—a survey. *Discrete applied mathematics*, 155(5):623–648, 2007.

W. D. Pisinger, A. B. Rasmussen, and R. Sandvik. Solution of large quadratic knapsack problems through aggressive reduction. *INFORMS Journal on Computing*, 19(2):280–290, 2007.

D. J. Rader. *Valid inequalities and facets of the quadratic 0-1 knapsack polytope*. Rutgers University. Rutgers Center for Operations Research [RUTCOR], 1997.

M. A. Readdy. Polytopes. 2013.

G. Schäfer. *Discrete Optimization*. Lecture Notes of the Master Course,http://homepages.cwi.nl/~schaefer/courses/lnmb-do11/main.html, January 10, 2012.

WikiKP. Knapsack problem. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/Knapsack_problem, Accessed April 2012.

C. Witzgall. Mathematical methods of site selection for electronic message systems (ems). *NASA STI/Recon Technical Report N*, 76:18321, 1975.