

**Recycling valid inequalities for combinatorial
optimization under budgeted uncertainty
Wiederverwertung gultiger Ungleichungen für die
Kombinatorische Optimierung unter
Unsicherheiten mit Budget**

Masterarbeit

Vorgelegt von

Marius Fleuster

aus Aachen

Angefertigt am

Lehrstuhl II für Mathematik

der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule Aachen

29. Oktober, 2021

Erstgutachter:

Prof. Dr. Arie M.C.A. Koster

Lehrstuhl II für Mathematik

RWTH Aachen

Zweitgutachter: :

Prof. Dr.

Lehrstuhl II für Mathematik

RWTH Aachen

Contents

Chapter 1. Introduction	3
Chapter 2. Robust Optimization: Basic formulations	5
1. Basic Formulations	5
2. Relaxation and Strong Inequalities	8
3. Clique inequalities and the Conflict Graph	12
Chapter 3. Clique Partitions in the Conflict Graph	21
1. Application of Clique Partitions to Robust Formulations	21
2. Finding Clique Partitions	23
Chapter 4. Additional inequalities in the robust setting	25
1. Additional inequalities in the robust relaxation of the Knapsack problem	25
2. Analysis of the quality of Clique Inequalities	32
Bibliography	33

CHAPTER 1

Introduction

CHAPTER 2

Robust Optimization: Basic formulations

1. Basic Formulations

Consider a classical Integer Linear Programming formulation of a problem with binary variables (for example a Knapsack instance).

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where $A \in \mathbb{R}^{n \times m}$ for $n, m \in \mathbb{N}$, where $c \in \mathbb{R}^n$ and where for two vectors $y, z \in \mathbb{R}^m$ the relation $y \leq z$ holds if and only if $y_i \leq z_i$ for all $0 \leq i \leq m$.

The above formulation reflects a situation where the modeler faces costs c and matrix coefficients (for example weights in a Knapsack instance) that he is certain of. In some situations though one cannot be exactly sure about the exact values of the cost vector or the matrix coefficients for that matter. It may happen that one only knows intervals of uncertainty $I_{i,j}$ for $i = 0, \dots, m$ and $j = 1, \dots, n$, where $c_j \in I_{0,j}$ and $A_{i,j} \in I_{i,j}$ for $i = 1, \dots, m$ and $j = 1, \dots, n$. This thesis considers exclusively the special case $I_{0,j} = [c_j, c_j + \hat{c}_j]$ for $j = 1, \dots, n$ and $I_{i,j} = [A_{i,j}, A_{i,j}]$ for the uncertainty intervals of the matrix coefficients. Thus only the costs are uncertain.

Since in practice a deviation of all coefficients to their worst case may be considered unrealistic Bertsimas and Sim propose to consider a scenario, where for a fixed number $\Gamma \in [0, n]$ only $\lfloor \Gamma \rfloor$ cost values deviate from their minimal value c_j to their maximal value $c_j + \hat{c}_j$ and an additional value differs by $(\Gamma - \lfloor \Gamma \rfloor)\hat{c}_j$. One then considers the following minimization problem:

DEFINITION 2.1. For $\Gamma \in [0, n]$ and $\hat{c}_j \geq 0$ for $j = 1, \dots, n$ the minimization problem

$$\begin{aligned} (2.1.1) \quad z_{rob} = \min \quad & \sum_{j=1}^n c_j x_j + \max_{S \cup \{t\} \subseteq \{1, \dots, n\}, |S| \leq \lfloor \Gamma \rfloor} (\Gamma - \lfloor \Gamma \rfloor) \hat{c}_t x_t + \sum_{s \in S} \hat{c}_s x_s \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \{0, 1\}^n. \end{aligned}$$

is called a *robust formulation* of the IP

$$\begin{aligned} (2.1.2) \quad \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \{0, 1\}^n. \end{aligned}$$

We denote the set $\{x \in \{0, 1\}^n \mid Ax \leq b\}$ by P .

The above formulation considers for every $y \in P$ the worst case deviation of $\lfloor \Gamma \rfloor$ cost values to their maximum interval bound, and one partial deviation with regards to minimizing the costs. The cost function is therefore not linear anymore. Yet the following Lemma suggest a way to find the solution to the robust formulation by solving a mixed integer problem via dualization. The proof follows (Bertsimas/Stim).

LEMMA 2.2. *Let*

$$\beta(y) := \max_{S \cup \{t\} \subseteq \{1, \dots, n\}, |S| \leq \lfloor \Gamma \rfloor} (\Gamma - \lfloor \Gamma \rfloor) \hat{c}_t y_t + \sum_{s \in S} \hat{c}_s y_s$$

for $y \in P$. Then

$$(2.2.1) \quad \begin{aligned} \beta(y) = \max \quad & \sum_{j=1}^n \hat{c}_j y_j z_j \\ \text{s.t.} \quad & \sum_{j=1}^n z_j \leq \Gamma \\ & z_i \in [0, 1], i = 1, \dots, n \end{aligned}$$

holds and by dualization one gets

$$(2.2.2) \quad \begin{aligned} \beta(y) = \min \quad & \sum_{j=1}^n p_j + \Gamma \cdot z \\ \text{s.t.} \quad & p_j + z \geq \hat{c}_j y_j, j = 1, \dots, n \\ & p_j \geq 0, j = 1, \dots, n \\ & z \geq 0. \end{aligned}$$

PROOF. Let $S \cup \{t\} \subseteq \{1, \dots, n\}$ be such that $|S| \leq \lfloor \Gamma \rfloor$ and such that

$$\beta(y) = \hat{c}_t y_t + \sum_{s \in S} \hat{c}_s y_s.$$

Define $z_s := 1$ for all $s \in S$ and $z_t := (\Gamma - \lfloor \Gamma \rfloor)$ and $z_u := 0$ for all $u \in U := \{1, \dots, n\} \setminus S \cup \{t\}$. By definition of S this is a feasible solution of (2.2.1). Assume this was not optimal. In order to increase another variable z_u for $u \in U$ either a z_s for $s \in S$ or z_t has to be decreased. Since the objective value of 2.2.1 increases, $\hat{c}_u y_u$ must be greater than $\hat{c}_a y_a$ for at least one $a \in S$ or greater than $\hat{c}_t y_t$. In the former case

$$\sum_{s \in S \setminus a} \hat{c}_s y_s + \hat{c}_u y_u > \sum_{s \in S} \hat{c}_s y_s$$

holds and in the latter case

$$(\Gamma - \lfloor \Gamma \rfloor) \hat{c}_u y_u > (\Gamma - \lfloor \Gamma \rfloor) \hat{c}_t y_t.$$

Either way $S \cup \{t\}$ is not an optimal selection and therefore

$$\beta(y) > \hat{c}_t y_t + \sum_{s \in S} \hat{c}_s y_s.$$

We have thus shown the equality of $\beta(y)$ and the solution of 2.2.1.

The second equality is a straight forward application of dualization for the matrix A with $A_{1,j} = 1$ for $j = 1, \dots, n$ and $A_{i,j} = \delta_{i+1,j}$ for $i = 2, \dots, n+1$ and $j = 1, \dots, n$ and right hand side b with $b_1 = \Gamma$ and $b_i = 1$ for $i = 2, \dots, n+1$. \square

With this Lemma and in particular with the formulation (2.2.2), the following Theorem can be stated as the basis for an easier treatment of the robust formulation.

THEOREM 2.3. *The solution of the robust formulation of the IP (2.1.2) is equal to the solution of the mixed integer program*

$$\begin{aligned}
 (2.3.1) \quad z_{rob} = \min \quad & \sum_{j=1}^n c_j x_j + \sum_{j=1}^n p_j + \Gamma \cdot z \\
 \text{s.t.} \quad & Ax \leq b, \\
 & p_j + z \geq \hat{c}_j x_j, \quad j = 1, \dots, n \\
 & p_j \geq 0, \quad j = 1, \dots, n \\
 & z \geq 0, \\
 & x \in \{0, 1\}^n.
 \end{aligned}$$

PROOF. The robust formulation (2.1.1) can be written as

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n c_j x_j + \beta(x) \\
 \text{s.t.} \quad & Ax \leq b, \\
 & x \in \{0, 1\}^n.
 \end{aligned}$$

Substituting $\beta(x)$ by the expression (2.2.2) yields the desired result. \square

The mixed integer formulation (2.3.1) makes the robust optimization problem accessible for the usage of common software for linear programming. Unfortunately the number of variables has more than doubled in comparison to the original non robust formulation. Also the problem has naturally become more complex as the variables x_j have to be balanced against the values of z and the p_j . If the coefficient \hat{c}_j is too large, setting $x_j = 1$ may be unfeasible, even if it was part of a minimal solution to (2.1.2). In that case either p_j or z need to become larger, which increases the objective value. Increasing z is very pricey but allows all p_j at once to be lower.

Robust Optimization can make easy problems hard. Consider the following example:

Example 2.4

Consider the problem of finding a Vertex Cover in a graph $G = (V, E)$ with vertex set V and edge set $E \subseteq \binom{V}{2}$. That is a subset $C \subseteq V$ with $C \cap e \neq \emptyset$ for all $e \in E$. This is a trivial task, since V is already a Vertex Cover. A corresponding

IP formulation would be:

$$\begin{aligned} \min & 0 \\ \text{s.t. } & x_i + x_j \geq 1 \text{ for } \{i, j\} \in E \\ & x \in \{0, 1\}^{|V|}. \end{aligned}$$

Now let $\Gamma = k \leq |V|$ be an integer number and $\hat{c}_i = 1$ for all $i = 1, \dots, |V|$ (while $c_i = 0$). In the original robust formulation this translates to

$$\begin{aligned} \min & \max_{S \subseteq V, |S|=k} \sum_{s \in S} \hat{c}_s x_s \\ \text{s.t. } & x_i + x_j \geq 1 \text{ for all } \{i, j\} \in E \\ & x \in \{0, 1\}^{|V|}. \end{aligned}$$

If a solution to the above robust formulation yields a value smaller than k , then the Graph has a Vertex Cover smaller than k . This problem is known to be NP-hard.

Nevertheless it can be shown, that if for the initial discrete polyhedron P and for every cost function $c^T x$ the corresponding IP can be solved in polynomial time, then the robust formulation can be solved in polynomial time as well.

THEOREM 2.5 (Bertsimas Sim: Robust discrete optimization and network flows). *Let P be the set of feasible vectors from Definition 2.1 and let z_{rob} be the solution of the robust formulation (2.1.1). We assume that $\hat{c}_1 \geq \hat{c}_2 \geq \dots \geq \hat{c}_n$ and we set $\hat{c}_{n+1} = 0$. Then*

$$z_{rob} = \min_{l=1, \dots, n+1} G^l,$$

where

$$(2.5.1) \quad \begin{aligned} G^l &= \min \Gamma \hat{c}_l + c^T x + \sum_{j=1}^l (\hat{c}_j - \hat{c}_l) x_j \\ & \quad x \in P \end{aligned}$$

for all $l = 1, \dots, n+1$.

REMARK 2.6. So to find the solution to the robust formulation it suffices to solve $n+1$ problems over the discrete polyhedron P of the original problem. If this can be done in polynomial time, the robust formulation can be solved in polynomial time as well.

Even though one may just solve $n+1$ instances of the original problem to solve the robust variant, this may pose a risk in case the underlying problem is not known to be solvable in polynomial time or is NP-hard. This thesis will try to give possible improvements for the solution of (2.1.1), if the aim is to solve the MLP directly using techniques like relaxation.

2. Relaxation and Strong Inequalities

Relaxation is an essential tool in modern MLP solvers. Relaxed variants of Mixed Integer Programs are often way easier to solve.

DEFINITION 2.7. The *relaxation* of the robust formulation (2.1.1) is the linear program

$$\begin{aligned}
 z_{rel} = \min & \sum_{j=1}^n c_j x_j + \sum_{j=1}^n p_j + \Gamma \cdot z \\
 \text{s.t. } & Ax \leq b, \\
 (2.7.1) \quad & p_j + z \geq \hat{c}_j x_j, j = 1, \dots, n \\
 & p_j \geq 0, j = 1, \dots, n \\
 & z \geq 0, \\
 & x \in [0, 1]^n.
 \end{aligned}$$

So the binary solution vector can obtain non discrete values between 0 and 1.

A classical result from linear programming guarantees a polynomial time algorithm for (2.7.1).

THEOREM 2.8. *The robust relaxation (2.7.1) can be solved in polynomial time and*

$$z_{rel} \leq z_{rob}$$

holds. (maybe give the state of the art running times...)

A feasible and integer solution to (2.7.1) yields a feasible solution for (2.3.1). But the results can also be arbitrarily bad.

Example 2.9

Consider the problem of choosing the cheapest of several (n) items with the same cost ($c_i = 1$) and with maximal deviation of one item ($\Gamma = 1$) with cost deviation value of 1 ($\hat{c}_i = 1$). The relaxation of the robust formulation of this problem is

$$\begin{aligned}
 \min & \sum_{i=1}^n x_i + \sum_{j=1}^n p_j + z \\
 \text{s.t. } & \sum_{i=1}^n x_i = 1, \\
 & p_j + z \geq x_j, j = 1, \dots, n \\
 & p_j \geq 0, j = 1, \dots, n \\
 & z \geq 0, \\
 & x \in [0, 1]^n.
 \end{aligned}$$

While an obvious solution to the original robust problem would be to choose any item, setting all $p_i = 0$ and $z = 1$ with an objective value of 2, the solution to the relaxation can be drastically improved. One feasible solution, which happens to be optimal as well, is $x_i = \frac{1}{n}$ for $i = 1, \dots, n$, $p_i = 0$ for $i = 1, \dots, n$ and $z = \frac{1}{n}$. This solution has an objective value of $1 + \frac{1}{n} = \frac{n+1}{n}$. This is not a good behaviour of the relaxed problem and a general problem.

Example 2.10

Consider a general knapsack instance (the cost coefficients must be negative and \hat{c}_i is still positive) and its robust relaxation

$$\begin{aligned}
& \min \sum_{j=1}^n c_j x_j + \sum_{j=1}^n p_j + \Gamma \cdot z \\
& \text{s.t.} \quad \sum_{j=1}^n a_j x_j \leq b, \\
& \quad p_j + z \geq \hat{c}_j x_j, \quad j = 1, \dots, n \\
& \quad p_j \geq 0, \quad j = 1, \dots, n \\
& \quad z \geq 0, \\
& \quad x \in [0, 1]^n.
\end{aligned}$$

Let (x^*, p^*, z^*) be an optimal solution to the robust problem and assume that the number of non-zero entries of x^* ($N := \{x_i^* \mid x_i^* = 1\}$) is significantly lower than the number of zero entries of x^* . Assume furthermore $\sum_{j=1}^n a_j x_j^* < b$. Let $x_i^* = 1$ such that there exist an entry $x_k^* = 0$ with $c_k < c_i$. This implies $\sum_{j \in N \setminus \{i\}} a_j x_j^* + a_k > b$, because otherwise x^* would not be optimal.

If we assume additionally that \hat{c}_i is maximal for all \hat{c}_m for $m \in N$. Then a shift of weight from x_i to x_k of ϵ allows for an adjustment of z^* by ϵ , as long as

$$\begin{aligned}
& \epsilon(a_k - a_i) \leq b - \sum_{j \in N} x_j^*, \\
& \epsilon \hat{c}_k \leq \hat{c}_i(1 - \epsilon), \\
& \text{and } \hat{c}_i(1 - \epsilon) \geq \hat{c}_m, m \in N
\end{aligned}$$

hold, meanwhile the costs of the items do not decrease. So the objective value decreases and the relaxed version of the robust problem does probably not obtain an integer (partial) solution x .

The above examples show that in order to solve the relaxed robust problem it is often better to spread little weight on many entries of the x -vector of a given feasible solution (x, p, z) of (2.7.1) instead of concentrating a maximum weight of 1 on fewer entries. That is because z can often be kept lower this way. This behaviour is not desirable, as many techniques of modern MIP solvers rely on relaxation and techniques as the feasibility pump for example. For those techniques to work well, a solution of the relaxation with many integer values is preferred. To tackle this problem a derivation of additional inequalities for the MIP (2.3.1), which often make deviations from integer solutions unfeasible in the relaxed robust optimization problem.

The following general insight will be of great help.

LEMMA 2.11. *The inequalities $p_j + z \geq \hat{c}_j x_j$ from the robust formulation (2.3.1) are equivalent to the non-linear inequalities*

$$p_j + z \cdot x_j \geq \hat{c}_j \cdot x_j$$

for $j = 1, \dots, n$. That is (2.3.1) and

$$\begin{aligned}
 (2.11.1) \quad & \min \sum_{j=1}^n c_j x_j + \sum_{j=1}^n p_j + \Gamma \cdot z \\
 & \text{s.t. } Ax \leq b, \\
 & p_j + z \cdot x_j \geq \hat{c}_j x_j, j = 1, \dots, n \\
 & p_j \geq 0, j = 1, \dots, n \\
 & z \geq 0, \\
 & x \in \{0, 1\}^n
 \end{aligned}$$

yield the exact same solutions.

PROOF. Consider a feasible solution (x, p, z) of (2.3.1). Then either $x_j = 0$ and the inequality $p_j = p_j + z \cdot x_j \geq \hat{c}_j x_j$ is trivially true, since $p_j \geq 0$ was true in the first place, or $x_j = 1$, in which case $\hat{c}_j x_j \leq p_j + z = p_j + z \cdot x_j$ is true as well. On the other hand for a solution (x, p, z) of (2.11.1) again $x_j = 0$ implies $p_j + z \geq \hat{c}_j x_j = 0$ trivially since $p_j, z \geq 0$. But if $x_j = 1$ then $p_j + z = p_j + z \cdot x_j \geq \hat{c}_j x_j$. \square

Lemma 2.11 can be used to derive new feasible inequalities including the variables p_j for $j = 1, \dots, n$ and z from (2.3.1) from certain feasible inequalities for the original problem 2.1.2.

THEOREM 2.12. *Let the inequality*

$$\sum_{i=1}^n d_i x_i \leq e$$

with $d_i \geq 0$ for all $i = 1, \dots, n$ be a feasible inequality for (2.1.2). That means the inequality is valid for all vectors in the polyhedron of (2.1.2). Then the inequality

$$\sum_{i=1}^n d_i p_i + e \cdot z \geq \sum_{i=1}^n d_i \hat{c}_i x_i$$

is feasible for (2.3.1).

PROOF. Since the inequalities $p_j + z \cdot x_j \geq \hat{c}_j x_j$ are feasible for (2.3.1) by Lemma 2.11, and since $d_j \geq 0$,

$$d_j p_j + z \cdot d_j x_j \geq d_j \hat{c}_j x_j$$

is feasible. Summing those inequalities up over all $j = 1, \dots, n$ yields the feasible inequality

$$\sum_{i=1}^n (d_i p_i + z \cdot d_i x_i) \geq \sum_{i=1}^n d_i \hat{c}_i x_i.$$

The left hand side can be written as

$$\sum_{i=1}^n d_i p_i + z \cdot \sum_{i=1}^n d_i x_i.$$

Because

$$\sum_{i=1}^n d_i x_i \leq e$$

is feasible, this yields

$$\sum_{i=1}^n d_i p_i + z \cdot \sum_{i=1}^n d_i x_i \leq \sum_{i=1}^n d_i p_i + e \cdot z$$

and thus proves the claim. \square

With this one can return to Example 2.9.

Example 2.13

In the situation of Example 2.9 the equality (it is also an inequality)

$$\sum_{i=1}^n x_i = 1$$

is feasible for the original IP and the coefficients are positive. Thus

$$\sum_{i=1}^n p_i + 1 \cdot z \geq \sum_{i=1}^n x_i$$

is a feasible inequality for the robust formulation. If one add this inequality to the robust formulation and considers the relaxation

$$\begin{aligned} \min \quad & \sum_{i=1}^n x_i + \sum_{j=1}^n p_j + z \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1, \\ & p_j + z \geq x_j, \quad j = 1, \dots, n \\ & \sum_{i=1}^n p_i + z \geq \sum_{i=1}^n x_i \\ & p_j \geq 0, \quad j = 1, \dots, n \\ & z \geq 0, \\ & x \in [0, 1]^n, \end{aligned}$$

the solution of $x_i = \frac{1}{n}$, $p_i = 0$ and $z = \frac{1}{n}$ is not a valid one anymore. Instead z has to have a value of 1 or $\sum_{i=1}^n p_i = 1$ because $\sum_{i=1}^n x_i = 1$ in every case and the newly added inequality implies

$$\sum_{i=1}^n p_i + z \geq 1.$$

It can be deduced that the optimal objective value is now 2 which can be achieved by setting $x_1 = 1$ and $z = 1$ and every other variable to 0. This problem may still have non integer solutions though, with $x_i = \frac{1}{n}$, $p_i = \frac{1}{n}$ and $z = 0$ for $i = 1, \dots, n$.

3. Clique inequalities and the Conflict Graph

This section will introduce so called Clique inequalities, which can be very useful inequalities in the robust setting. A variant of these inequalities has already appeared in Example 2.13.

DEFINITION 2.14. For variables x_1, \dots, x_n and a subset $Q \subseteq \{1, \dots, n\}$ the inequality

$$\sum_{i \in Q} x_i \leq 1$$

is called *Clique Inequality* of the Clique Q .

Clique Inequalities form cliques in the so called Conflict Graph. Conflict Graphs were introduced into the world of MIP solvers by Achterberg (Conflict Analysis in Mixed Integer Programming).

DEFINITION 2.15. Let

$$\begin{aligned} & \min c^T x \\ & \text{s.t. } Ax \leq b, \\ & \quad x \in \{0, 1\}^n, \end{aligned}$$

be an integer program. The Graph $G_C = (V_C, E_C)$ with vertex set

$$V_C = \{x_i \mid i = 1, \dots, n\} \cup \{\bar{x}_i \mid i = 1, \dots, n\}$$

and edges

$$\begin{aligned} E_C = & \{[x_i, x_j] \mid \text{there is no feasible solution } x \in \{0, 1\}^n \\ & \text{to the integer program with } x_i = x_j = 1\} \\ & \cup \{[x_i, \bar{x}_j] \mid \text{there is no feasible solution } x \in \{0, 1\}^n \\ & \text{to the integer program with } x_i = 1 = 1 - x_j\} \\ & \cup \{[\bar{x}_i, \bar{x}_j] \mid \text{there is no feasible solution } x \in \{0, 1\}^n \\ & \text{to the integer program with } x_i - 1 = 1 = 1 - x_j\} \end{aligned}$$

is called the *Conflict Graph* to the integer program. So the Conflict Graph has all the variables and their complementary variables as vertices and two vertices are connected if it is impossible for both variables (or complementary variables respectively) to be true at the same time in any solution of the integer program.

Often inequalities of the form

$$\sum_{i \in I} d_i x_i + \sum_{j \in J} d_j \bar{x}_j \leq e$$

will be written instead of

$$\sum_{i \in I} d_i x_i + \sum_{j \in J} d_j (1 - x_j) \leq e.$$

The following easy Lemma summarizes the connection between Clique Inequalities and cliques in a Conflict Graph.

LEMMA 2.16. *Let*

$$\begin{aligned} & \min c^T x \\ & \text{s.t. } Ax \leq b, \\ & \quad x \in \{0, 1\}^n, \end{aligned}$$

be an integer program. Then the inequality

$$\sum_{i \in I} x_i + \sum_{j \in J} \bar{x}_j \leq 1$$

for $I \cup J \subseteq \{1, \dots, n\}$ and $I \cap J = \emptyset$ is feasible for the integer program if and only if

$$Q = \bigcup_{i \in I} \{x_i\} \cup \bigcup_{j \in J} \{\bar{x}_j\}$$

forms a clique in the conflict graph G_C .

Clique Inequalities and the Conflict Graph to an integer problem will be very important pieces in possible improvements of the relaxation (2.7.1). As was already demonstrated in Example 2.13 Clique inequalities and the use of Theorem 2.12 may close the gap between the optimal objective value of (2.3.1) and that of (2.7.1). Clique Inequalities have an additional advantage. They often allow a reduction of the number of variables.

THEOREM 2.17. *Let a robust formulation of an integer program (2.1.2) of the form (2.3.1) be given. Let furthermore*

$$\sum_{i \in Q} x_i \leq 1$$

be a feasible Clique Inequality for $Q \subseteq \{1, \dots, n\}$ to the original IP (2.1.2). Let $Q^c := \{1, \dots, n\} \setminus Q$. Then the MIP

$$\begin{aligned} (2.17.1) \quad & \min \sum_{j=1}^n c_j x_j + \sum_{j \in Q^c}^n p_j + p_Q + \Gamma \cdot z \\ & \text{s.t. } Ax \leq b, \\ & p_j + z \geq \hat{c}_j x_j, j \in Q^c \\ & p_Q + z \geq \sum_{i \in Q} \hat{c}_i x_i \\ & p_j \geq 0, j \in Q^c \\ & p_Q \geq 0 \\ & z \geq 0, \\ & x \in \{0, 1\}^n \end{aligned}$$

has the same optimal objective value as (2.3.1). Denote by p_{-S} the vector with entries of p of the index set $S \subseteq \{1, \dots, n\}$. The map

$$f : (x, z, p_{-Q}, p_{-Q^c})^T \mapsto (x, z, p_Q = \sum_{i \in Q} p_i, p_{-Q^c})^T$$

is a surjection of optimal solutions of (2.3.1) to optimal solutions of (2.17.1).

PROOF. It suffices to show that the map f is a surjection between optimal solutions, because the objective values are identical for preimage and image of the map. So let $(x^*, z^*, p_{-Q}^*, p_{-Q^c}^*)^T$ be an optimal solution to (2.3.1). Because

of Lemma 2.12 and since the Clique Inequality is feasible for the original IP and therefore also for (2.3.1), the inequality

$$\sum_{i \in Q} p_i + z \geq \sum_{i \in Q} \hat{c}_i x_i$$

is feasible for (2.3.1). So the inequality

$$p_Q + z \geq \sum_{i \in Q} \hat{c}_i x_i$$

is valid for $f((x^*, z^*, p_{-Q}^*, p_{-Q^c}^*)^T)$. So

$$f((x^*, z^*, p_{-Q}^*, p_{-Q^c}^*)^T) = (x^*, z^*, p_Q^*, p_{-Q^c}^*)^T$$

is a feasible solution to (2.17.1). Now the Clique Inequality implies that $x_i^* = 1$ for at most one index $i \in Q$. But as p_i^* is minimal

$$p_j^* = \max\{0, \hat{c}_j x_j^*\}$$

must hold for all $j \in Q$. Therefore $p_j = 0$ for all but at most one $j \in Q$. Thus $\sum_{j \in Q} p_j^* = p_i^*$ holds for one $i \in Q$. As the Clique Inequality also holds for (2.17.1) one can deduce that an optimal solution $(\tilde{x}, \tilde{z}, \tilde{p}_Q, \tilde{p}_{-Q^c})$ must satisfy

$$p_Q + z \geq \hat{c}_j x_j$$

for all $j \in Q$ as $\tilde{x}_i = 1$ holds for at most one index $i \in Q$. Now there is a preimage

$$(\tilde{x}, \tilde{z}, \tilde{p}_{-Q}, \tilde{p}_{-Q^c}) \in f^{-1}((\tilde{x}, \tilde{z}, \tilde{p}_Q, \tilde{p}_{-Q^c}))$$

that satisfies $p_i = p_{-Q}$ ($p_j = 0$ for all $j \in Q \setminus \{i\}$) and thus satisfies the inequalities

$$p_j + z \geq \hat{c}_j x_j$$

because $\tilde{x}_j = 0$ and satisfies

$$p_i + z \geq \hat{c}_i x_i$$

because $p_i = p_Q$ holds. Thus there is one preimage of any optimal solution, which is a feasible solution to (2.3.1). This solution has the same objective value as its image. Therefore the objective value of $(\tilde{x}, \tilde{z}, \tilde{p}_{-Q}, \tilde{p}_{-Q^c})^T$ is greater or equal than the value of $(x^*, z^*, p_{-Q}^*, p_{-Q^c}^*)^T$, which has the same value as $(x^*, z^*, p_Q^*, p_{-Q^c}^*)^T$, which has an objective value greater or equal to $(\tilde{x}, \tilde{z}, \tilde{p}_{-Q}, \tilde{p}_{-Q^c})^T$. Thus all values are the same and $(x^*, z^*, p_Q^*, p_{-Q^c}^*)^T$ is an optimal solution.

So it was shown that every optimal solution of (2.3.1) is mapped to at least one optimal solution of (2.17.1) and every preimage of an optimal solution of (2.17.1) is an optimal solution to (2.3.1). \square

The effect of Theorem 2.17.1 on the size of a robust formulation can be very meaningful. This is better demonstrated by an example.

Example 2.18

Consider the following Knapsack problem

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & x \in \{0, 1\}^n. \end{aligned}$$

Let $\hat{c}_1 = \hat{c}_2 = \hat{c}_7 = 2$, $\hat{c}_4 = \hat{c}_5 = 1$, $\hat{c}_3 = 0$, $\hat{c}_6 = 3$ and $\Gamma = 1$. The robust formulation (2.3.1) of this problem has six additional inequalities and seven additional variables. A little precomputing shows that the Conflict Graph contains a clique consisting of the variables x_3, x_4, x_5, x_6, x_7 . Thus the Clique inequality

$$x_3 + x_4 + x_5 + x_6 + x_7 \leq 1$$

is feasible for the problem. With Theorem 2.17.1 the robust problem can be reduced to

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 + z + p_Q + p_1 + p_2 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & p_Q + z \geq x_4 + x_5 + 3x_6 + 2x_7 \\ & p_1 + z \geq 2x_1 \\ & p_2 + z \geq 2x_2 \\ & z \geq 0 \\ & p_Q, p_1, p_2 \geq 0 \\ & x \in \{0, 1\}^n. \end{aligned}$$

This formulation has only four additional variables and three inequalities. An optimal solution is for example $x_1 = 1, x_5 = 1, z = 2$ while everything else is zero, which makes for an objective value of -4 .

The effect of the Clique Inequalities are twofold. A reduction of variables can have very positive effects on the computation time modern MIP solvers take. Thus a reduction is worthwhile just for that sake. Furthermore along with the reduction come new Inequalities which are often improving on the original robust formulation as was already demonstrated in Example 2.13. A continuation of the latest example should be enough to show the effectiveness of the combination of variable reduction and the implicit application of Theorem 2.12.

Example 2.19

Returning to Example 2.18. The relaxation of the robust problem without the variable reduction is

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 + z + p_Q + p_1 + p_2 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & p_1 + z \geq 2x_1, p_2 + z \geq 2x_2, \\ & p_4 + z \geq x_4, p_5 + z \geq x_5, \\ & p_6 + z \geq 3x_6, p_7 + z \geq 2x_7, \\ & z \geq 0 \\ & p_1, p_2, p_3, p_4, p_5, p_6, p_7 \geq 0 \\ & x \in [0, 1]^n. \end{aligned}$$

A feasible solution to this formulation would be

$$x = \left(\frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{3}, 0\right)^T$$

and $z = 1$ and $p = 0$. This solution has an objective value of $-5\frac{1}{3}$. Yet the inequality

$$p_4 + p_5 + p_6 + p_7 + z \geq x_4 + x_5 + 3x_6 + 2x_7$$

is not feasible for the above value, since

$$z = 1 < \frac{1}{2} + 3 \cdot \frac{1}{3}.$$

Indeed an optimal value of the relaxed problem

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 + z + p_Q + p_1 + p_2 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & p_Q + z \geq x_4 + x_5 + 3x_6 + 2x_7 \\ & p_1 + z \geq 2x_1 \\ & p_2 + z \geq 2x_2 \\ & z \geq 0 \\ & p_Q, p_1, p_2 \geq 0 \\ & x \in [0, 1]^n, \end{aligned}$$

which makes implicit use of Theorem 2.12, is

$$x = \left(\frac{1}{2}, \frac{1}{2}, 0, 0, 1, 0, 0\right)^T$$

and $z = 1$ and $p = 0$ with objective value -5 . So the robust reformulation (2.17.1) brings the relaxation closer to the integer robust formulation. [there is also an extended clique inequality containing x_1, x_2, x_5 with right side 2, which delivers an optimal solution.]

Clique Inequalities are a key factor for solving the robust problem with the help of relaxation techniques. To use the techniques from Examples 2.18 and 2.19 it is necessary to know at least parts of the Conflict Graph. Unfortunately finding the complete Conflict Graph of an arbitrary IP is NP-hard.

LEMMA 2.20. *The problem of finding the Conflict Graph of*

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \{0, 1\}^n, \end{aligned}$$

for an arbitrary binary integer program is NP-hard.

PROOF. It is well known that 3 – SAT, the problem of deciding the feasibility of a logical formula in conjunctive normal form, where every clause contains a maximum of three variables in negated or unnegated form, is NP-complete. This problem can be formulated as a binary integer program. For every clause $C = x \wedge y \wedge z$, where x, y, z may be negated or unnegated, the inequality

$$x + y + z \geq 1$$

is added to the binary integer program. After all clauses are transformed to inequalities the objective value min 0 is added. One then easily sees that this integer

program is solvable if and only if the logical formula is feasible.

Consider the Conflict Graph of this integer program. If the program is infeasible then every variable is connected to every other variable (negated or unnegated) in the Conflict Graph. Thus the Conflict Graph is complete. If the graph is on the other hand complete there cannot be a feasible solution to the program, which follows directly from Definition 2.15. Given the Conflict Graph one can determine in quadratic time in the number of vertices, if the graph is complete. Therefore the computation of the Conflict Graph cannot be done in polynomial time unless $P = NP$. \square

Even if the computation of the complete Conflict Graph of arbitrary binary integer problems may not be manageable in tolerable time frames the Conflict Graph can be partially constructed in reasonable time. Also for certain important problem classes the Conflict Graph can be completely computed even in low polynomial time. In particular this is the case for arbitrary Knapsack problems. Algorithm 1 does exactly that. The Algorithm is by Souza Britoa and Gambini Santos (Preprocessing and Cutting Planes with Conflict Graphs).

LEMMA 2.21. *Algorithm 1 is correct. It has a running time of $\mathcal{O}(n \log(n))$.*

PROOF. If the coefficient of a variable x_i appears negatively in the constraint. The algorithm implicitly performs a substitution of $a_i x_i$ by $-a_i(1 - x_i) = -a_i \bar{x}_i$. In order to keep the inequality correct the right hand side of the inequality has to be adjusted by an addition of $-a_i$. All of this happens in lines 5 and 6 of the Algorithm. In line 7 only the negated variable is added to the eventual subgraph's vertex set V_C and the unnegated variable will not appear for this particular constraint. Therefore the second remark from line 14 is justified and it should be understood that x_i may also mean \bar{x}_i , whichever one is in the vertex set. Overall the reformulation of the constraint does not exceed linear time.

The main part of the running time comes from sorting the set of coefficients. This can be achieved in $\mathcal{O}(n \log(n))$.

If after sorting the coefficients $a_{n-1} + a_n < b$ holds, no two other coefficients may exceed b in the sum. Then there actually no conflicts from the constraint. Let j be the smallest index as found in line 18 by interval nesting (which takes logarithmic time on the set of sorted coefficients). Then for all $h, i \geq j$ with $h \neq i$ the sum

$$a_h + a_i \geq a_j + a_{j+1} > b$$

as the coefficients are sorted in ascending order. The assignments of edges in line 19 is therefore correct. Similarly

$$a_k + a_i \geq a_k + a_{j_k} > b$$

holds for all $k < j$ and j_k chosen as in line 21, which justifies line 22. The loop from from line 20 itearates a maximum of n times and each step needs $\mathcal{O}(\log(n))$ running time (nested intervals). \square

input : A constraint $a^T x \leq b$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ of a binary integer program B .

output: A subgraph $H_C \leq G_C$ of the Conflict Graph of the binary integer program B .

```

1  $V_C \leftarrow \emptyset$ ;
2  $E_C \leftarrow \emptyset$ ;
3 for  $a_i$  coefficient of  $a$  do
4   if  $a_i < 0$  then
5      $b \leftarrow b - a_i$ ;
6      $a_i \leftarrow -a_i$ ;
7      $V_C \leftarrow V_C \cup \{\bar{x}_i\}$ ;
8   end
9   if  $a_i > 0$  then
10     $V_C \leftarrow V_C \cup \{x_i\}$ ;
11  end
12 end
13 sort the coefficients  $a_1, \dots, a_n$  in ascending order;
14 without loss of generality now  $0 \leq a_1 \leq a_2 \leq \dots \leq a_n$  and  $x_i = \bar{x}_i$ 
   whatever is element of  $V_C$ ;
15 if  $a_{n-1} + a_n < b$  then
16   return  $(V_C, E_C)$ ;
17 end
18 find smallest index  $j$  such that  $a_j + a_{j+1} > b$  by the method of nested
   intervals;
19  $E_C \leftarrow E_C \cup \{[x_h, x_i] \mid h, i \geq j, h \neq i\}$ ;
20 for  $k < j$  do
21   find smallest index  $j_k$  such that  $a_k + a_{j_k} > b$  by the method of
     nested intervals;
22    $E_C \leftarrow E_C \cup \{[x_k, x_i] \mid i \geq j_k\}$ ;
23 end
24 return  $(V_C, E_C)$ ;

```

Algorithm 1: FindConflictGraphofConstraint

Algorithm 1 can be applied to every constraint of a binary integer program and the (partial) Conflict Graph can be extended with every additional constraint. This may take $\mathcal{O}(mn \log(n))$ running time, where m is the number of constraints.

LEMMA 2.22. *The Conflict Graph of an arbitrary Knapsack instance can be found in $\mathcal{O}(n \log(n))$ time.*

PROOF. Let an instance

$$\begin{aligned} \min & c^T x \\ \text{s.t. } & a^T x \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

of the Knapsack problem be given. That is $a \in \mathbb{R}_{\geq 0}^n$, $b \geq 0$ and $c \in \mathbb{R}_{\leq 0}^n$ holds. It is assumed that $a_i \leq b$. Then every negated variable \bar{x}_i is connected to every other variable (negated or unnegated) in the vertex set V_C of the Conflict Graph, because not choosing an item x_i and choosing any other item is always an option as all coefficients are positive and no item exceeds the bound of b on its own. It is furthermore assumed that $a_i > 0$ for all $i \leq n$. Otherwise the corresponding item can always be chosen as part of any solution. Also the indices of a are assumed to be sorted already in ascending order. Algorithm 1 will then find every conflict, that is every edge from the Conflict Graph of the Knapsack problem.

To prove this let $[x_h, x_i] \in E_C$ with $h < i$. That means in particular that x_h and x_i cannot be chosen to be one at the same time, even if all other items are not chosen into the Knapsack (that is $x_l = 0$ for $l \neq h, i$). Then $a_h + a_i > b$ must hold. If $h, i \geq j$, where j is the index found in line 18 from Algorithm 1, then $[x_h, x_i]$ is added to the edge set in line 19. If $h < j$, then $i \geq j_h$, where j_h is the index found in line 21. But then $[x_h, x_i]$ is added to the edge set in line 22. \square

COROLLARY 2.23. *Let*

$$\begin{aligned} \min & c^T x \\ \text{s.t. } & A^T x \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

be a binary integer program with $A \in \mathbb{R}_{\geq 0}^{m \times n}$. Then the Conflict Graph $G_C = (V_C, E_C)$ of that program can be found in $\mathcal{O}(mn \log(n))$ time.

PROOF. Let $[x_h, x_i] \in E_C$. Then in particular $y \in \{0, 1\}^n$ with $y_i = y_h = 1$ and $y_l = 0$ for $l \neq h, i$ is not feasible for the binary integer program. Then y is unfeasible for at least one constraint, which is a constraint of a Knapsack instance. Applying Algorithm 1 to this constraint computes the desired edge of the Conflict Graph. This follows the same way as in Lemma 2.22. \square

Algorithm 1 is a compromise between computability and completeness. It can be considered a good compromise as many important classes of problems are solved exactly by the algorithm. The algorithm can be already used to compute Cliques in the conflict graph. In line 19 the edge set is actually the edge set of a clique in the Conflict Graph. Yet to fully apply the power of Theorem 2.17 it is best to find a clique partition of the Conflict graph.

CHAPTER 3

Clique Partitions in the Conflict Graph

1. Application of Clique Partitions to Robust Formulations

In order to make full use the variable reduction technique introduced in Theorem 2.17 and demonstrated in Example 2.18 and Example 2.19, as well as the expected improvement of the solution of the relaxed problems it seems appropriate to find as many cliques in the Conflict Graph as possible. The aim is to apply Theorem 2.17 several times for different cliques to reduce the number of variables of the robust formulation as much as possible. The definition of a Clique Partition is needed.

DEFINITION 3.1. Let $G = (V, E)$ be a graph. A *Clique Partition* of the graph G is a set

$$P = \{Q_1, Q_2, \dots, Q_k\}$$

such that every Q_i is a clique in the graph for $i = 1, \dots, k$,

$$\bigcup_{i=1, \dots, k} Q_i = V$$

and

$$Q_i \cap Q_j = \emptyset$$

for $i, j \in \{1, \dots, k\}$ and $i \neq j$.

Example 3.2

Consider again the following Knapsack instance

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & x \in \{0, 1\}^n. \end{aligned}$$

and its robust formulation from Example 2.18. Applying Algorithm 1 (the coefficients are already ordered) determines j from line 18 to be 3. Thus there is one clique consisting of variables $\{x_3, x_4, x_5, x_6, x_7\}$. Iterating the loop from line 20 for indices $k_1 = 1$ and $k_2 = 2$ yields cliques $\{x_1, x_6, x_7\}$ and $\{x_2, x_6, x_7\}$ (see Figure 1). The negated variables of the Conflict Graph are not interesting here, as they cannot be used to reduce the variables in the robust formulation. There are many Cliques in this graph and naturally many of them intersect. To apply Theorem 2.17 several times on different cliques these cliques must not intersect. In the leftmost graphic from Figure 1 the Clique Partition $\{\{x_3, x_4, x_5, x_6, x_7\}, \{x_1\}, \{x_2\}\}$ is displayed. Clique two and three of this partition are not proper cliques. One application of Theorem 2.17 yields the reduced robust formulation that was already considered in

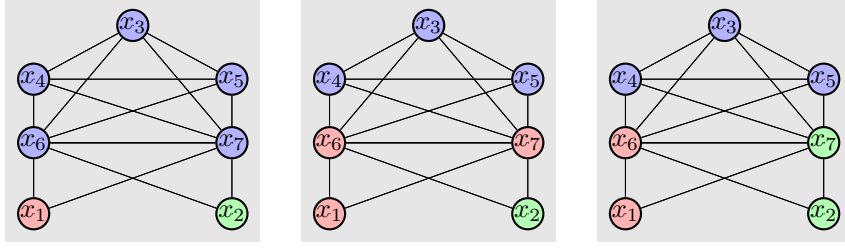


FIGURE 1. Different Clique Partitions of the (partial) Conflict Graph from Example 3.2

Example 2.18. Another Clique Partition would be $\{\{x_3, x_4, x_5\}, \{x_1, x_6, x_7\}, \{x_2\}\}$. Applying Theorem 2.17 to the first clique yields the inequality

$$p_{Q_1} + z \geq x_4 + x_5.$$

Applying the theorem for a second time yields the reduced robust formulation

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 + z + p_{Q_1} + p_{Q_2} + p_2 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & p_{Q_1} + z \geq x_4 + x_5 \\ & p_{Q_2} + z \geq 2x_1 + 3x_6 + 2x_7 \\ & p_2 + z \geq 2x_2 \\ & z \geq 0 \\ & p_{Q_1}, p_{Q_2}, p_2 \geq 0 \\ & x \in \{0, 1\}^n. \end{aligned}$$

One last partition is $\{\{x_3, x_4, x_5\}, \{x_1, x_6\}, \{x_2, x_7\}\}$, which contains only proper cliques. In effect one gets a reduced robust reformulation

$$\begin{aligned} \min \quad & -2x_1 - 2x_2 - 1x_3 - 3x_4 - 4x_5 - 7x_6 - 3x_7 + z + p_{Q_1} + p_{Q_2} + p_{Q_3} \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 \leq 3, \\ & p_{Q_1} + z \geq x_4 + x_5 \\ & p_{Q_2} + z \geq 2x_1 + 3x_6 \\ & p_{Q_3} + z \geq 2x_2 + 2x_7 \\ & z \geq 0 \\ & p_{Q_1}, p_{Q_2}, p_{Q_3} \geq 0 \\ & x \in \{0, 1\}^n. \end{aligned}$$

For the MIP formulation of all the above problems the difference between these Clique Partitions is not that relevant in theory, as all partitions lead to only three p variables and three inequalities instead of seven p variables and six additional inequalities compared to the original (non robust) formulation. The difference between Clique Partitions is more relevant for the relaxed variants of the above MIPs.

Even though it was already used in Example 3.2 an application of Theorem 2.17 for several cliques was not made precise. This will be done in the following corollary to this theorem.

COROLLARY 3.3. *Let a robust formulation of an integer program (2.1.2) of the form (2.3.1) be given. Let furthermore $P = \{Q_1, \dots, Q_k\}$ be a Clique Partition of the conflict graph. Then the MIP*

$$\begin{aligned}
 (3.3.1) \quad & \min \sum_{j=1}^n c_j x_j + \sum_{Q \in P} p_Q + \Gamma \cdot z \\
 & \text{s.t. } Ax \leq b, \\
 & p_Q + z \geq \sum_{i \in Q} \hat{c}_i x_i \text{ for all } Q \in P \\
 & p_Q \geq 0 \text{ for all } Q \in P \\
 & z \geq 0, \\
 & x \in \{0, 1\}^n
 \end{aligned}$$

has the same optimal objective value as (2.3.1). Denote by p_{-S} the vector with entries of p of the index set $S \subseteq \{1, \dots, n\}$. The map

$$f : (x, z, p_{-Q_1}, \dots, p_{-Q_k})^T \mapsto (x, z, p_{Q_1} = \sum_{i \in Q_1} p_i, \dots, p_{Q_k} = \sum_{i \in Q_k} p_i)^T$$

is a surjection of optimal solutions of (2.3.1) to optimal solutions of (3.3.1).

PROOF. The same methods as in Theorem 2.17 can be used on all cliques in the clique partition P . \square

2. Finding Clique Partitions

To reduce the number of variables of a robust formulation a subgraph of the Conflict Graph is needed. Then a Clique Partition as small as possible can be found. Unfortunately the general problem of finding a Minimum Clique Partition is NP-hard.

LEMMA 3.4. *Given a graph $G = (V, E)$, the problem of finding a Clique Partition of minimum cardinality of G is NP-hard.*

PROOF. A Clique Partition of size k of the graph G is a Graph Coloring of size $n - k$ in the complement graph \tilde{G} . That is because a clique in G is an independent set in \tilde{G} . Finding a Minimum Clique Partition is thus equivalent to finding a Maximum Graph Coloring. This is the well known GRAPH – COLORING problem, which is NP-hard. \square

Not only is the general problem of finding a Minimum Clique Partition NP-hard, but this NP-hardness is already relevant to integer programs with constraint matrix $A \in \mathbb{R}_{\geq 0}^{m \times n}$.

LEMMA 3.5. *Let $G = (V, E)$ be an arbitrary graph and $|V| = n$. Consider the ILP*

$$\begin{aligned} & \min c^T x \\ & \text{s.t. } x_i + x_j \leq 1, \text{ for all } \{i, j\} \in E \\ & \quad x \in \{0, 1\}^n. \end{aligned}$$

Then G is a subgraph of the Conflict Graph of the above problem.

There is little hope to find minimal clique partitions of arbitrary Conflict Graphs in reasonable computation time, unless $\text{NP} = \text{P}$. Yet for a simple but important class of ILPs the Conflict Graph has a simpler structure which makes finding a clique partition feasible.

LEMMA 3.6. *The Conflict Graph of an arbitrary Knapsack instance is chordal. A minimal clique partition for the Knapsack problem can be found in linear time.*

CHAPTER 4

Additional inequalities in the robust setting

Theorem 2.12 has already shown to have valuable impact on the treatment of problems in the robust setting and often improves the methods of finding an optimal solution to (2.1.1). Relaxation of (2.3.1) reduces computation times but often resulted in fractional solutions whose objective value is too far off of an optimal integer solution to be very useful. The reformulation (3.3.1) often reduces fractionality in the solution vectors drastically depending on the Clique Partition considered. Unfortunately the Clique Partition approach is often not enough and many non trivial IPs do have trivial Conflict Graphs. Theorem 2.12 offers solutions to that case as well. In this chapter the use of additional inequalities with non-negative coefficients of the original IP formulation in the robust setting will be analyzed. A lemma will be proved which explains the effects many of those additional inequalities have of on fractional solutions. The computational implications of these results will be analyzed.

1. Additional inequalities in the robust relaxation of the Knapsack problem

LEMMA 4.1. *Let the Knapsack problem*

$$\begin{aligned} \min & c^T x \\ \text{s.t.} & a^T x \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

and its robust variant (2.3.1) be given. Then there is an optimal solution (x^, z^*, p^*) to the relaxed robust variant (2.7.1) of this problem, so that the set*

$$N := \{i \in \{1, \dots, n\} \mid 0 < x_i^* < 1, \hat{c}_i > 0, a_i > 0\}$$

contains at the most one index $j \in N$ with

$$\hat{c}_j x_j^* < z^*.$$

PROOF. Let (x^*, z^*, p^*) be the optimal solution chosen so that for

$$M_{x^*} := N \cap \{i \in \{1, \dots, n\} \mid \hat{c}_i x_i^* < z^*\}$$

the value $|M_{x^*}|$ is minimal.

If $|M_{x^*}| \geq 2$ holds then there exist $k, l \in M_{x^*}$, so that $\hat{c}_k x_k^* < z^*$ und $\hat{c}_l x_l^* < z^*$.

Without loss of generality $\frac{c_k}{c_l} \leq \frac{a_k}{a_l}$ is assumed.

Choose now $y_l := x_l^* - \delta$ and $y_k := x_k^* + \alpha\delta$ for

$$\delta \in (0, \min(\alpha^{-1}(1 - x_k^*), \alpha^{-1}(z^* - \hat{c}_k x_k^*), x_l^*))$$

and

$$\alpha \leq \frac{a_l}{a_k}.$$

Define $y_i := x_i$ for all $i \notin \{k, l\}$.

Now

$$a_l y_l + a_k y_k = a_l x_l^* + a_k x_k^* + \delta a_l \left(\frac{a_k}{a_l} \alpha - 1 \right) \leq a_l x_l^* + a_k x_k^*$$

and

$$c_l y_l + c_k y_k = c_l x_l^* + c_k x_k^* + \delta c_l \left(\frac{c_k}{c_l} \alpha - 1 \right) \leq c_l x_l^* + c_k x_k^*.$$

Because of optimality of the solution $\frac{c_k}{c_l} = \frac{a_k}{a_l}$ must hold, as the optimal value could be decreased otherwise.

Now δ can be chosen so that either $y_l = 0$, $y_k = 1$ or $\hat{c}_k y_k = z^*$ holds. But the above inequalities show that (y, z^*, p^*) is an optimal solution to the robust relaxation as well but with $|M_y| < |M_{x^*}|$, which contradicts the choice of (x^*, z^*, p^*) . Therefore $|M_{x^*}| \leq 1$ must hold. So there is at the most one index $j \in M_{x^*}$ with $\hat{c}_j x_j^* < z^*$, which is fully fractional. \square

Lemma 4.1 implies that all, except a maximum of one, fractional entries x_i^* of an optimal x -vector solution x^* of the robust relaxation suffice the equalition $\hat{c}_i x_i^* = z^* + p_i^*$. Now the following lemma restricts the number of indices for which this equality may hold, if a feasible inequality

$$\sum_{i=1}^n d_i \hat{c}_i x_i \leq ez + \sum_{i=1}^n d_i p_i$$

as in Theorem 2.12 is added.

LEMMA 4.2. *Let a relaxation of a robust formulation (2.7.1) be given. Let a feasible inequality as in Theorem 2.12 of the form*

$$\sum_{i=1}^n d_i \hat{c}_i x_i \leq ez + \sum_{i=1}^n d_i p_i$$

with $d_i \geq 0$ for all $i = 1, \dots, n$ be added to the relaxation. Let (x^, z^*, p^*) be an optimal solution to this LP, where $z^* \neq 0$. Then the set*

$$Eq := \{i \in \{1, \dots, n\} \mid \hat{c}_i x_i^* = z^* + p_i^*\}$$

suffices the condition

$$\sum_{i \in Eq} d_i \leq e.$$

If

$$\begin{aligned}
 (4.2.1) \quad \phi(z^*, p^*, Eq) := & \min \sum_{j \notin Eq} \frac{z^* c_j}{\hat{c}_j} y_j \\
 \text{s.t.} \quad & \sum_{j \notin Eq} \frac{z^* A_{i,j}}{\hat{c}_j} y_j \leq b - \sum_{j \in Eq} A_{i,j} x_j^* \text{ for } i = 1, \dots, m \\
 & \sum_{j \notin Eq} d_j y_j \leq e - \sum_{i \in Eq} d_i \\
 & 0 \leq y_j < 1 \text{ for } j \notin Eq \\
 & y_j \leq \frac{\hat{c}_j}{z^*} \text{ for } j \notin Eq
 \end{aligned}$$

is defined, then the objective value of the relaxed LP is

$$\sum_{j \in Eq} c_j x_j^* + \Gamma z^* + \sum_{j=1}^n p_j^* + \phi(z^*, p^*, Eq).$$

PROOF. Take the optimal solution (x^*, z^*, p^*) and consider the sum

$$\sum_{i \in Eq} d_i \hat{c}_i x_i^* = \sum_{i \in Eq} d_i (z^* + p_i^*) = z^* \left(\sum_{i \in Eq} d_i \right) + \sum_{i \in Eq} d_i p_i^*.$$

For indices $j \notin Eq$ $p_j^* = 0$ must hold, because $\hat{c}_j x_j^* < z^*$ and therefore to minimize p_j^* it must be set to 0. So

$$z^* \left(\sum_{i \in Eq} d_i \right) + \sum_{i \in Eq} d_i p_i^* = z^* \left(\sum_{i \in Eq} d_i \right) + \sum_{i=1}^n d_i p_i^*$$

must be valid. As the inequality

$$\sum_{i \in Eq} d_i \hat{c}_i x_i \leq \sum_{i=1}^n d_i \hat{c}_i x_i \leq ez + \sum_{i=1}^n d_i p_i$$

holds, because $d_i \geq 0$ for all $i = 1, \dots, n$, the inequality

$$z^* \left(\sum_{i \in Eq} d_i \right) + \sum_{i=1}^n d_i p_i^* \leq ez^* + \sum_{i=1}^n d_i p_i^*$$

and therefore

$$0 \leq z^* \left(e - \sum_{i \in Eq} d_i \right)$$

must hold. As $z^* \neq 0$, $\sum_{i \in Eq} d_i \leq e$ and therefore $\sum_{i \in Eq} d_i \leq e$ can be obtained. If z^*, p^* and Eq is given the values $\hat{c}_i x_i^* = z^* + p_i^*$ can be computed for $i \in Eq$. Then the values x_j^* for $j \notin Eq$ suffice the inequalities

$$\sum_{j \notin Eq} A_{i,j} x_j^* + \sum_{j \in Eq} A_{i,j} x_j^* \leq b$$

and

$$\sum_{j \notin Eq} d_j \hat{c}_j x_j^* \leq z^* \left(e - \sum_{i \in Eq} d_i \right).$$

And since $\hat{c}_j x_j^* < z^*$ the x_j^* for $j \notin Eq$ form an optimal solution of

$$\begin{aligned} \phi(z^*, p^*, Eq) &:= \min \sum_{j \notin Eq} c_j x_j \\ \text{s.t. } &\sum_{j \notin Eq} A_{i,j} x_j \leq b - \sum_{j \in Eq} A_{i,j} x_j^* \text{ for } i = 1, \dots, m \\ &\sum_{j \notin Eq} d_j \hat{c}_j x_j \leq e - \sum_{i \in Eq} d_i \\ &0 \leq x_j \leq 1, \hat{c}_j x_j < z^* \text{ for } j \notin Eq \end{aligned}$$

and

$$\sum_{j \in Eq} c_j x_j^* + \Gamma z^* + \sum_{j=1}^n p_j^* + \phi(z^*, p^*, Eq)$$

is the objective value of the relaxed LP in consideration. Now a simple substitution of $y_j = \frac{\hat{c}_j}{z^*} x_j$ for $j \notin Eq$ in the above expression yields the claim. \square

REMARK 4.3. $\phi(z, p, Eq)$ really only depends on z^*, p^* and Eq , as $\hat{c}_i x_i^* = z^* + p_i^*$ for all $i \in Eq$. A value $\phi(z, p, Eq)$ can be defined for any $z \geq 0$, vector $p \in \mathbb{R}_{\geq 0}^n$ and set $Eq \subseteq \{1, \dots, n\}$ in the same way as in (4.2.1). As the y_i must be strictly smaller than 1 and as the right hand side of the inequalities in (4.2.1) is not positive for all choices of z, p and Eq the minimum may not exist. In that case a value of ∞ may be assigned. One can then say that if $\phi(z, p, Eq) = \infty$ then z, p and $x_i = z + p_i$ for $i \in Eq$ cannot be part of an optimal solution to the robust relaxation of the Knapsack problem with the additional inequality

$$\sum_{i=1}^n d_i \hat{c}_i x_i \leq ez + \sum_{i=1}^n d_i p_i.$$

Lemma 4.2 is interesting in two ways. First it makes a statement about the number of items in the set Eq , which can be understood as a fractional equivalent of items, which determine the worst case deviation of Γ items from their original values. Second the formulation (4.2.1) has properties which allow to quantify to some extent the connection between z^* and optimal values for x_j^* with $j \notin Eq$. Additional valid inequalities which arise from Theorem 2.12 help to stabilize the optimal solution entries in some sense.

Example 4.4

Recalling Example 2.9 a problem of the robust relaxation (2.7.1) without additional inequalities is a tendency to choose z^* small and the number of items not equal to zero very large. This is to even out the diminished range the values x_j^* may lie in if z^* is small. Lemma 4.1 states that nearly all items with fractional values from an optimal solution of the robust relaxation of a Knapsack problem lie in the set Eq . In the example $|Eq|$ is equal to n . As the inequality

$$\sum_{i=1}^n x_i \leq 1$$

is valid for the original robust problem the inequality

$$\sum_{i=1}^n x_i \leq z + \sum_{i=1}^n p_i$$

is valid for the robust formulation. Lemma 4.2 then projects that for any optimal solution of the relaxation of the robust formulation with the additional inequality, that $|Eq| = \sum_{i \in Eq} 1 \leq 1$. So the optimal solution of the relaxation without additional inequality is not a solution of the relaxation with the addition of

$$\sum_{i=1}^n x_i \leq z + \sum_{i=1}^n p_i.$$

Apart from

$$\sum_{i=1}^n x_i \leq 1$$

the inequalities

$$\sum_{i=1}^{\frac{n}{2}} x_i \leq 1$$

and

$$\sum_{i=\frac{n}{2}}^n x_i \leq 1$$

are valid for the original Knapsack problem as well, if n is even. Adding the inequalities

$$\sum_{i=1}^{\frac{n}{2}} x_i \leq z + \sum_{i=1}^{\frac{n}{2}} p_i$$

and

$$\sum_{i=\frac{n}{2}}^n x_i \leq z + \sum_{i=\frac{n}{2}}^n p_i$$

to the robust relaxation an extension of Lemma 4.2 to several additional equations predicts $|Eq \cap \{1, \dots, \frac{n}{2}\}| \leq 1$ and $|Eq \cap \{\frac{n}{2} + 1, \dots, n\}| \leq 1$. And indeed an optimal solution could be $x_1^* = x_n^* = \frac{1}{2}$ and $z^* = \frac{1}{2}$, while every other variable is zero.

For the classical Knapsack problem there is one valid inequality

$$\sum_{i=1}^n a_i x_i \leq b$$

with $a_i > 0$. According to Theorem 2.12 the inequality

$$\sum_{i=1}^n a_i \hat{c}_i x_i \leq bz + \sum_{i=1}^n a_i p_i$$

is valid for the robust formulation. How powerful is the addition of this inequality to the formulation with regards to the relaxation? The following lemma suggest an improvement over the original problem's relaxation.

LEMMA 4.5. *Let the Knapsack problem*

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^n a_i x_i \leq b \\ & x \in \{0, 1\}^n \end{aligned}$$

and its robust variant (2.3.1) be given, where it is assumed that $0 < \hat{c}_j \leq |c_j|$ for all $j \in \{1, \dots, n\}$. According to Theorem 2.12 the inequality

$$\sum_{i=1}^n a_i \hat{c}_i x_i \leq bz + \sum_{i=1}^n a_i p_i$$

is valid and may be added to the robust formulation. Consider now the relaxation of this problem and an optimal solution $(x^*, z^*, p^*)^T$ of this relaxation, with $z^* \neq 0$. Let

$$Eq := \{i \in \{1, \dots, n\} \mid \hat{c}_i x_i^* = z^* + p_i^*\}$$

be the set of indices as in Lemma 4.2. Then either $z^* > \hat{c}_j$ for at least one $j \notin Eq$ or $x_i^* = 1$ for all $i \in Eq$ and there is a maximum of one index $k \notin Eq$, such that $x_k^* > 0$. In either case there is an index $m \in Eq$ with $\hat{c}_m x_m^* = z^*$. So if $z^* \leq \hat{c}_j$ for all $j \notin Eq$, then $z^* = \hat{c}_m$ for $m \in Eq$.

PROOF. Assume $z^* \leq \hat{c}_j$ for all $j \notin Eq$. Consider $\phi(z^*, p^*, Eq)$ the solution of a linear program as in Lemma 4.2 with two valid inequalities

$$\sum_{i \notin Eq} \frac{z^*}{\hat{c}_i} a_i y_i \leq b - \sum_{i \in Eq} a_i x_i^*$$

and

$$\sum_{i \notin Eq} a_i y_i \leq b - \sum_{i \in Eq} a_i.$$

By assumption $\frac{\hat{c}_j}{z^*} \geq 1$ and therefore it suffices to consider $y_j < 1$ for all $j \notin Eq$. Also by assumption $\frac{z^*}{\hat{c}_j} \leq 1$ for all $j \notin Eq$ and therefore

$$\sum_{i \notin Eq} \frac{z^*}{\hat{c}_i} a_i y_i \leq \sum_{i \notin Eq} a_i y_i \leq b - \sum_{i \in Eq} a_i \leq b - \sum_{i \in Eq} a_i x_i^*.$$

Thus the second inequality dominates the first one and it is enough to use the second inequality only. The problem of finding an optimal solution to the linear program which is $\phi(z^*, p^*, Eq)$ is therefore a relaxed Knapsack problem with the restriction that $y_j < 1$ and objective function

$$z^* \sum_{j \notin Eq} \frac{c_j}{\hat{c}_j} y_j.$$

The factor z^* can be ignored for the computation of an optimal solution. To solve a relaxed Knapsack problem a greedy algorithm can be used, which chooses an index $k \notin Eq$ with $\frac{c_k}{\hat{c}_k a_k}$ maximal and increase y_k from 0 to 1 as long as $a_k y_k \leq b - \sum_{i \in Eq} a_i$. As $y_k < 1$, y_k cannot be increased to a maximal value of 1. Thus there must be a value $\alpha < 1$ with $a_k \alpha = b - \sum_{i \in Eq} a_i$ and y_k can be chosen to

be α . This must be the optimal solution to the minimization problem and thus $\phi(z^*, p^*, Eq) = \frac{z^* c_k}{\hat{c}_k} \alpha$. As demonstrated in Lemma 4.2 the objective value to the relaxed robust problem with additional valid inequality is

$$\sum_{j \in Eq} c_j x_j^* + \Gamma z^* + \sum_{j=1}^n p_j + \phi(z^*, p^*, Eq) = \sum_{j \in Eq} c_j x_j^* + \Gamma z^* + \sum_{j=1}^n p_j + \frac{z^* c_k}{\hat{c}_k} \alpha.$$

As $x_j^* = \frac{z^* + p_j^*}{\hat{c}_j}$ for all $j \in Eq$ and $p_j^* = 0$ for $j \notin Eq$ this becomes

$$\sum_{j \in Eq} (1 + \frac{c_j}{\hat{c}_j}) p_j^* + z^* (\Gamma + \sum_{j \in Eq} \frac{c_j}{\hat{c}_j} + \frac{c_k}{\hat{c}_k} \alpha).$$

Assume $x_j^* < 1$ for one $j \in Eq$. Now set $\tilde{x}_j = 1$ and $\tilde{p}_j := \hat{c}_j - z^*$. Let \tilde{p} be the vector with $\tilde{p}_i := p_i^*$ and $\tilde{x}_i := x_i^*$ for $i \neq j$ and \tilde{p}_j as defined before. The set of indices $\widetilde{Eq} = \{i \in \{1, \dots, n\} \mid \tilde{x}_i = \tilde{p}_i + z^*\} = Eq$. As

$$b - \sum_{i \in \widetilde{Eq}} a_i = b - \sum_{i \in Eq} a_i \leq b - \sum_{i \in Eq} a_i \tilde{x}_i$$

the first inequality of $\phi(z^*, \tilde{p}, \widetilde{Eq})$ is still dominated by the second inequality and thus $y_k = \alpha$ remains an optimal solution to the LP of $\phi(z^*, \tilde{p}, \widetilde{Eq})$ and therefore $\phi(z^*, \tilde{p}, \widetilde{Eq}) = \frac{z^* c_k}{\hat{c}_k} \alpha$. Again the objective value of the relaxed robust problem is

$$\sum_{j \in Eq} (1 + \frac{c_j}{\hat{c}_j}) \tilde{p}_j + z^* (\Gamma + \sum_{j \in Eq} \frac{c_j}{\hat{c}_j} + \frac{c_k}{\hat{c}_k} \alpha) \leq \sum_{j \in Eq} (1 + \frac{c_j}{\hat{c}_j}) p_j^* + z^* (\Gamma + \sum_{j \in Eq} \frac{c_j}{\hat{c}_j} + \frac{c_k}{\hat{c}_k} \alpha),$$

because $(1 + \frac{c_j}{\hat{c}_j}) \leq 0$ as $c_j < 0$ and $\hat{c}_j \leq |c_j|$. Thus x_j^* can be chosen to be 1.

To prove that there exists an $m \in Eq$, such that $\hat{c}_m x_m^* = z^*$, let $m \in Eq$ be the index such that p_m^* is minimal among all values p_i^* with $i \in Eq$. Define $\tilde{p}_i := p_i^* - p_m^* \geq 0$ for all $i \in Eq$, and $\tilde{p}_j := p_j^* = 0$ for all $j \notin Eq$. Define furthermore $\tilde{z} = z^* + p_m^*$. Then

$$\sum_{i \in Eq} \tilde{p}_i + \Gamma \tilde{z} = \sum_{i \in Eq} p_i^* + \Gamma z^* + p_m^* (\Gamma - |Eq|) \geq \sum_{i \in Eq} p_i^* + \Gamma z^*.$$

On the other hand one can define $\bar{z} := 0$, $\bar{p}_i := p_i^* + z^*$ for $i \in Eq$ and $\bar{p}_j := p_j^* = 0$ for $j \notin Eq$. Then

$$\sum_{i \in Eq} \bar{p}_i + \Gamma \bar{z} = \sum_{i \in Eq} p_i^* + \Gamma z^* + z_m^* (|Eq| - \Gamma) \geq \sum_{i \in Eq} p_i^* + \Gamma z^*$$

and from both inequalities it can be concluded that $|Eq| = \Gamma$ or $p_m^* = 0$ or $z^* = 0$. As $z^* > 0$ was assumed, either $|Eq| = \Gamma$ holds, in which case $(x^*, \tilde{z}, \tilde{p})^T$ is an optimal solution as well, or $p_m^* = 0$. In both cases there is a solution and an index $m \in Eq$, such that $p_m^* = 0$ and therefore $\hat{c}_m x_m^* = z^*$ for one index $m \in Eq$. \square

The proof of the previous lemma demonstrates the utility of an inequality

$$\sum_{i=1}^n a_i \hat{c}_i x_i \leq bz + \sum_{i=1}^n a_i p_i$$

in addition to the usual inequalities, in particular the original Knapsack inequality, in the robust relaxation. The function ϕ from Lemma 4.2 shows that the additional inequality stabilizes the values of items outside of the set Eq , if the value of z becomes too small. Without this additional inequality for small values of z the

solution vector of the LP does depend on z . Important statements about the solution structure can be made and there is always a guaranteed lower bound on z^* .

2. Analysis of the quality of Clique Inequalities

In principle the reasoning from the proof of Lemma 4.5 applies to all kinds of valid inequalities derived from Theorem 2.12

$$\sum_{i=1}^n d_i \hat{c}_i x_i \leq ez + \sum_{i=1}^n d_i p_i,$$

where one special case are the Clique Inequalities which were crucial for the variable reduction as in Theorem 2.17. In Example 4.4 it was demonstrated that using Theorem 2.12 on the valid Knapsack inequality directly and adding the resulting inequality to the robust relaxation may have an advantage over applying the same procedure on a number of Clique Inequalities as in (3.3.1). This is not that surprising considering that valid Clique Inequalities often do not carry all the information about the interconnections between the variables, whereas the Knapsack inequality defines those interconnections. This is also suggested by Lemma 4.2, where the set Eq is often much more restricted by the original Knapsack inequality than an arbitrary Clique Inequality.

The same can be said about the robust formulations of IPs (2.3.1), where the matrix $A \in \mathbb{R}_{\geq 0}^{m \times n}$ has only positive entries. In that case it may be better for the quality of a solution to the relaxation to apply Theorem 2.12 to each row of $Ax \leq b$ and adding each inequality to the robust formulation, instead of using the clique approach.

Still, using the approach with Clique Partitions (3.3.1) often reduces the number of variables and inequalities substantially which improves the running time of common solvers for LPs. The number of variables and inequalities becomes a real issue in bigger instances with thousands of variables and inequalities. In case the matrix A does contain negative entries, Theorem 2.12 cannot be used directly on the matrix rows, but in this case the clique approach remains an option.

Bibliography