

# A Quick Practical Guide to Polyhedral Analysis in Integer Programming

Vicky Mak-Hau

**Abstract** Polyhedral analysis is one of the most interesting elements of integer programming and has been often overlooked. It plays an important role in finding exact solutions to an integer program. In this paper, we will discuss what polyhedral analysis is, and how some constraints for an integer programming model are “ideal” in the sense that if the model contains all of these “ideal” constraints, then the integer optimal solution can be obtained by simply solving a linear programming relaxation of the integer program. This paper serves as a quick guide for young researchers and PhD students.

**Keywords** Polyhedral Analysis · Integer Programming · Combinatorial optimization

## 1 Introduction

*Integer Programming* (IP) has been widely used in real-life combinatorial optimization problems, such as routing, scheduling, rostering, and hub-locations. There are three major types of integer linear programs. Let  $c$  be a nonnegative vector in  $\mathbb{R}^n$ ,  $g$  be a nonnegative vector in  $\mathbb{R}^p$ ,  $A$  be a  $m \times n$  matrix,  $D$  be a  $m \times p$  matrix, and  $b$  a  $m$ -vector, with all elements in  $c$ ,  $g$ ,  $A$ ,  $D$ , and  $b$  real numbers.

A *mixed integer linear programming problem* (MILP) is of the form:

$$\max\{cx + gy : Ax + Dy \leq b, x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^p\} \quad (1)$$

As with usual practice, we do not distinguish between row and column vectors in our notation. The focus of our paper will be on *pure integer linear programming problems* (ILPs):

---

V. Mak-Hau (✉)  
School of Information Technology, Deakin University, Waurn Ponds,  
Geelong, Australia  
e-mail: vicky.mak@deakin.edu.au

$$\max\{cx : Ax \leq b, x \in \mathbb{Z}_+^n\}, \quad (2)$$

and in particular, *pure binary linear programming problems* (BIPs):

$$\max\{cx : Ax \leq b, x \in \{0, 1\}^n\}. \quad (3)$$

Of all aspects of an integer programming model, perhaps one is most interested in how well it performs in terms of finding the optimal solution(s). Without doubt, the solution methodology used (e.g., branch-and-bound, branch-and-cut, column generation etc.) plays a key role in the computational performance; however, the performance also largely depends on the strength of the constraints of the integer program.

Consider two pure general integer programs described below.

$$\text{IP1 : } \max\{cx + dy : 2x + 2y \leq 5, x, y \in \mathbb{Z}\} \quad (4)$$

$$\text{IP2 : } \max\{cx + dy : x + y \leq 2, x, y \in \mathbb{Z}\} \quad (5)$$

Both IP1 and IP2 contain the same set of feasible integer points:

$$X = \{(0, 0), (0, 1), (0, 2), (1, 0), (2, 0), (1, 1)\}.$$

But IP2 is better in the sense that by solving the LP relaxation of IP2, we get the integer solution for free and hence obtain the optimal solution to the integer program, whereas with IP1, depending on the values of  $c$  and  $d$ , we may get a fractional value when an LP relaxation is solved, in which case we obtain only an upper bound to the optimal value of the integer program. IP2 is, therefore, an *ideal formulation*. We now introduce the mathematical background, definitions, and notation.

Let  $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$  be a polytope (a bounded polyhedron) containing the set of all integer solutions to the system of linear inequalities  $Ax \leq b$ . A point  $x \in P$  is a convex combination of points in  $P$  if there exists  $x^i$ , for  $i \in \{1, \dots, t\}$  such that  $x = \sum_{i=1}^t \lambda_i x^i$ , where  $\sum_{i=1}^t \lambda_i = 1$  and  $\lambda_i \geq 0$  for all  $i = 1, \dots, t$ . Let  $X = \{x \in \mathbb{Z}_+^n : Ax \leq b\}$ . In this case,  $P$  is the linear programming relaxation of  $X$ .

The *convex hull* of  $X$  denoted by  $\text{conv}(X)$  contains all feasible convex combinations of the points in  $X$ . An integer programming model is *ideal* if  $P = \text{conv}(X)$ , in which case solving  $P$  will produce naturally integer solutions, hence the optimal solution for the integer program. In other words, there is no LP-IP (integrality) gap. There are very few classic combinatorial optimization problems that present no integrality gap, except for the ones for which the following property holds.

**Proposition 1** ([1], Proposition 3.3) *A linear program  $\max\{cx : Ax \leq b, x \in \mathbb{R}_+^n\}$  has an integral optimal solution and finite optimal objective value under the necessary and sufficient condition that  $A$  is totally unimodular, and that  $b$  is a vector of integers.*

One classic combinatorial optimization problem with such properties is the assignment problem.

We say that  $P$  is *full-dimensional* if  $\dim(P) = n$ , the number of decision variables, in which case, one can find  $n + 1$  affinely independent points in  $P$ .

Consider again the system of linear inequalities  $Ax \leq b$ . Let  $M^=$  be the index sets of equality constraints in  $Ax \leq b$ , we denote these constraints by  $(A^=, b^=)$ ; and  $M^{\leq}$  be the index sets of inequalities of  $Ax \leq b$ , we denote these constraints by  $(A^{\leq}, b^{\leq})$ . We assume that  $(A^{\leq}, b^{\leq})$  cannot be written as a linear combination of  $(A^=, b^=)$ . We have that  $M^= \cap M^{\leq} = \emptyset$  and that  $M^= \cup M^{\leq} = M$ , with  $M$  the index set of all constraints in  $Ax \leq b$ . Now, if  $P$  is not full-dimensional, then  $\dim(P) = n - \text{rank}(A^=, b^=)$ .

Let  $X = \{x \in \mathbb{Z}_+^n : Ax \leq b\}$ , if  $\text{conv}(X)$  is full-dimensional, then there is a unique set of constraints that describes the  $\text{conv}(X)$ . If such a complete polyhedral description is found, then solving the linear programming relaxation will give us the optimal solution to the integer program for free.

In most real-life integer programming models, however, particularly the classic traveling salesman- and vehicle routing-family problems, as there are equality constraints (the degree constraints), the corresponding polytopes are not full-dimensional. In such cases, there are in fact infinitely many complete polyhedral descriptions. Take the following integer program as an example.

$$X = \{x_1, x_2 \in \mathbb{Z} \mid x_1 + x_2 = 6, 2 \leq x_1 \leq 5, 1 \leq x_2 \leq 4\}. \tag{6}$$

The only feasible points are:  $X = \{(2, 4), (3, 3), (4, 2), (5, 1)\}$  and that  $\text{conv}(X)$  is one-dimensional, with  $(2, 4)$  and  $(5, 1)$  the extreme points. Given these two extreme points given, one possible complete description of  $\text{conv}(X)$  is:

$$X = \{x_1, x_2 \in \mathbb{R} \mid x_1 + x_2 = 6, x_1 \geq 2, x_1 \leq 5\}. \tag{7}$$

and another is:

$$X = \{x_1, x_2 \in \mathbb{R} \mid x_1 + x_2 = 6, x_2 \geq 1, x_2 \leq 4\}. \tag{8}$$

One can easily see that there are in fact infinitely many complete descriptions of  $\text{conv}(X)$ .

We now introduce a few terminologies regarding constraints of an integer program. Again, let  $X = \{x \in \mathbb{Z}_+^n : Ax \leq b\}$ . For simplicity, we will use  $(\pi, \pi_0)$  to represent a constraint defined by  $\pi x \leq \pi_0$ . We say that  $(\pi, \pi_0)$  is a *valid* constraint if  $\pi x \leq \pi_0$  for all  $x \in X$ . We say that  $F$  defines a *face* of  $\text{conv}(X)$  if  $F = \{x \in X : \pi x = \pi_0\}$ . A face is a *proper face* if  $F \neq \emptyset$  and  $F \neq \text{conv}(X)$ .

A face is called a *facet* if  $\dim(F) = \dim(\text{conv}(X)) - 1$ . One way to show that  $(\pi, \pi_0)$  defines a facet is to use the following theorem.

**Theorem 1.1** ([2], Theorem 3.6 in Sect. 1.4) *Let  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ , let  $(A^=, b^=)$  be the equality set of  $P \in \mathbb{R}^n$ , and let  $F = \{x \in P : \pi x = \pi_0\}$  be a proper face. We have that  $F$  is a facet of  $P$ , if and only if there exists  $\alpha \in \mathbb{R}^1$  and  $\mu \in \mathbb{R}^{|M^=|}$  such that  $(\alpha\pi + \mu A^=)x = (\alpha\pi_0 + \mu b^=)$ , for all  $x \in F$ .*

As an illustration, let:

$$X = \{x_1, x_2, x_3 \in \mathbb{Z}_+^3 : x_1 + x_2 \leq 6, x_1 \leq 5, x_2 \leq 4, x_3 = 3\}. \quad (9)$$

Suppose we want to show that  $x_2 \leq 4$  is facet-defining for  $\text{conv}(X)$ . We have that  $F = \{x \in X : x_2 = 4\}$ ,  $(\alpha\pi + \mu A^\top) = (0, \alpha, \mu)$ , with  $\alpha, \mu$  as unknowns, and for every  $x \in F = \{(0, 4, 3), (1, 4, 3), (2, 4, 3)\}$ , we have that  $(0, \alpha, \mu)x = (\alpha\pi_0 + \mu b^\top) = (4\alpha + 3\mu)$ . Hence  $x_2 \leq 4$  is a facet for  $X$ .

The theorem is mathematically very elegant, however in practice, it is easier to just demonstrate that the dimension of  $F$  is one less than the dimension of  $\text{conv}(X)$ , by finding the rank of the matrix with each row representing a distinct  $x \in F$ .

A complete polyhedral description is made up of facets. However hard it may seem, the more facets we can find for an integer program, the more likely we will have a small integrality gap, and consequently the more likely the exact method based on the integer program will perform well.

## 2 The Double Description Methods and Open Source Tools

There are a few good open source software that can be utilized for polyhedral analysis. For polytopes that are full-dimensional, one can use, e.g., `cdd/cdd+` [3] and `PORTA` [4]. However, `PANDA` [5] can handle polytopes that are not full-dimensional. In any case, all software are based on the idea of double description of a polyhedron.

The double description method was first proposed in [6]. Essentially, the idea is that every polyhedron can be described in two equivalent ways: a complete polyhedral description  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ , or the set of all extreme points in  $P$ , and from one, we can deduce the other. The former is referred to as the H-representation and the latter, the V-representation.

To explain how `PANDA` works, consider the following example.

$$X = \{x, y \in \{0, 1\}^2 : 2x + 2y \leq 3\}. \quad (10)$$

We first enter the LP relaxation of (10) into `PANDA`, using the H-representation, as below.

```
Names:
x   y
Inequalities:
2x + 2y <= 3
x <= 1
y <= 1
x >= 0
y >= 0
```

We then run PANDA by entering the command `./panda Toy.in > Toy.ext`. In the output file `Toy.ext`, we have the V-representation of the extreme points of the LP relaxation of (10).

```
Vertices/Rays:
1  2  2
2  1  2
0  1  1
1  0  1
0  0  1
```

The points above are the extreme points  $\{(\frac{1}{2}, 1), (1, \frac{1}{2}), (0, 1), (1, 0), (0, 0)\}$  for the LP relaxation of (10). For binary integer programs, to obtain the extreme points for  $\text{conv}(X)$ , one can simply remove the fractional extreme points.

To obtain the facets for  $\text{conv}(X)$ , one can enter the modified V-representation of the integer extreme points:  $(0, 1)$ ,  $(1, 0)$ , and  $(0, 0)$ , as below:

```
Names:
x  y
Vertices:
0  1
1  0
0  0
```

with the command `./panda Toy1.ext > Toy1.in`. PANDA will then return the facets for  $\text{conv}(X)$ .

```
Inequalities:
-x <= 0
-y <= 0
x + y <= 1
```

Since  $\text{conv}(X)$  is full-dimensional, the three constraints above define the complete description of the convex hull, and the description is unique.

In Fig. 1, we present an example of a basic formulation, in H-representation, of the Cardinality Constrained Multi-cycle Problem (CCMcP) [7] (see Model 4 therein), with only four vertices on the graph, and a cardinality restriction of three. (Readers are referred to the paper for the IP model, hence we do not repeat the description of CCMcP in this paper.)

When we execute PANDA, we obtain all the extreme points for the LP relaxation, including the fractional ones. After removing the fractional points, we obtain the set of extreme points in Fig. 2.

For pure binary integer programming problems, the extreme points of the convex hull are in fact the set of all feasible points. Let  $x_1^*$  and  $x_2^*$  be two distinct extreme points of a BIP; let  $X = \{x \in \{0, 1\}^n : Ax \leq b\}$ ; and let  $x = \alpha x_1^* + (1 - \alpha)x_2^*$ , for  $0 < \alpha < 1$ , then  $0 < x_j < 1$  for at least one  $j \in \{1, \dots, n\}$ , hence  $x \notin X$ . If we are able to systematically generate all the feasible points, then we have the set of all extreme points. (See, e.g., [8] for an example). When we do not know the set of all feasible integer points, we must carry out the steps described above. For most classic combinatorial optimization problems, however, in particular the TSP-family

---

```

Names:
x12    x13    x14    x21    x23    x24    x31    x32    x34    x41    x42    x43

Equations:
x14 +x24 +x34 -x41 -x42 -x43 = 0
x13 +x23 -x31 -x32 -x34 +x43 = 0
x12 -x21 -x23 -x24 +x32 +x42 = 0
-x12 -x13 -x14 +x21 +x31 +x41 = 0

Inequalities:

x12 +x13 +x14 <= 1
x21 +x23 +x24 <= 1
x31 +x32 +x34 <= 1
x41 +x42 +x43 <= 1

x21 +x31 +x41 <= 1
x12 +x32 +x42 <= 1
x13 +x23 +x43 <= 1
x14 +x24 +x34 <= 1

x12 +x23 +x34 <= 2
x23 +x34 +x41 <= 2
x12 +x34 +x41 <= 2
x12 +x23 +x41 <= 2

x12 +x24 +x43 <= 2
x24 +x31 +x43 <= 2
x12 +x31 +x43 <= 2
x12 +x24 +x31 <= 2

x13 +x24 +x32 <= 2
x24 +x32 +x41 <= 2
x13 +x24 +x41 <= 2
x13 +x32 +x41 <= 2

x13 +x34 +x42 <= 2
x21 +x34 +x42 <= 2
x13 +x21 +x42 <= 2
x13 +x21 +x34 <= 2

x14 +x23 +x42 <= 2
x23 +x31 +x42 <= 2
x14 +x23 +x31 <= 2
x14 +x31 +x42 <= 2

x14 +x32 +x43 <= 2
x21 +x32 +x43 <= 2
x14 +x21 +x32 <= 2
x14 +x21 +x43 <= 2

x12 >= 0
x13 >= 0
x14 >= 0
x21 >= 0
x23 >= 0
x24 >= 0
x31 >= 0
x32 >= 0
x34 >= 0
x41 >= 0
x42 >= 0
x43 >= 0

x12 <= 1
x13 <= 1
x14 <= 1
x21 <= 1
x23 <= 1
x24 <= 1
x31 <= 1
x32 <= 1
x34 <= 1
x41 <= 1
x42 <= 1
x43 <= 1

```

**Fig. 1** H-representation of model 4 of CCMcP in [7], with 4 vertices on the graph, and a cardinality restriction of 3

Names:	x12	x13	x14	x21	x23	x24	x31	x32	x34	x41	x42	x43
Vertices:	0	0	0	0	0	1	0	1	0	0	0	1
	0	0	0	0	1	0	0	0	1	0	1	0
	0	0	1	0	0	0	1	0	0	0	0	1
	0	0	1	0	1	0	0	1	0	1	0	0
	0	1	0	0	0	0	0	0	1	1	0	0
	0	1	0	0	0	1	1	0	0	0	1	0
	1	0	0	0	0	1	0	0	0	1	0	0
	1	0	0	0	1	0	1	0	0	0	0	0
	1	0	0	1	0	0	0	0	1	0	0	1
	0	0	0	0	0	0	0	0	1	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	1	0
	0	0	0	0	1	0	0	1	0	0	0	0
	0	0	1	0	0	0	0	0	0	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	1
	0	1	0	0	0	0	1	0	0	0	0	0
	0	1	0	1	0	0	0	1	0	0	0	0
	1	0	0	1	0	0	0	0	0	0	0	0

Fig. 2 Extreme points for the IP presented in Fig. 1

and VRP-family problems, as well as cycle and chain problems, we do know the set of all feasible integer solutions.

When we enter the V-representation of all the extreme points of a BIP, PANDA will return one complete description of the convex hull of the feasible integer points. Recall that if a polytope is full-dimensional, there is only one unique complete polyhedral description, otherwise, there are infinitely many. In Fig. 3, we have one of the complete polyhedral descriptions of the small CCMcP example returned by PANDA. So, even though we may have a class of constraints that we suspect is facet-defining, the fact that it does not appear in the output returned by the software does not mean that the constraint is definitely not facet-defining. The benefit of using a software for obtaining even one of the complete polyhedral description is that it can give us some constraints that we did not know are facet-defining, then we can proceed to establish the proofs.

Notice that the bound constraint  $x \geq 0$  is often facet-defining, see, e.g., [9], these are considered trivial constraints. Notice also that sometimes a constraint is only facet-defining for small problems, so in the end we cannot prove that they are facet-defining for all problem sizes. A constraint can also be facet-defining for larger problems, not toy-sized ones, so they are not found by a software.

### 3 Proof Techniques

*Direct proof* For BIPs of small scale, when we have all the extreme points explicitly generated, to show that a constraint  $(\pi, \pi_0)$  is facet-defining, one needs to show that the set of extreme points (e.g.,  $x^1, \dots, x^t$ , where each of  $x^j$ , for  $j = 1, \dots, t$ , is a  $n$ -dimensional vector) that satisfy  $\pi x = \pi_0$  are of dimension  $\dim(\text{conv}(X)) - 1$ . This can be done by calculating the rank of  $(x^1, -1), \dots, (x^t, -1)$ . If the rank is exactly  $\dim(\text{conv}(X))$ , then there are  $\dim(\text{conv}(X))$  affinely independent feasible solutions

**Fig. 3** One complete polyhedral description of the convex hull of the small CCMcP example

```

Equations:
x14 +x24 +x34 -x41 -x42 -x43 = 0
x13 +x23 -x31 -x32 -x34 +x43 = 0
x12 -x21 -x23 -x24 +x32 +x42 = 0

Inequalities:
-x21 <= 0
x23 +x24 +x31 +x34 +x41 +x43 <= 2
-x32 <= 0
-x42 <= 0
x41 +x42 +x43 <= 1
-x21 -x23 -x24 +x32 +x42 <= 0
x31 +x32 +x34 <= 1
-x21 -x23 +x32 -x43 <= 0
x24 +x34 -x41 -x42 -x43 <= 0
-x21 -x24 -x34 +x42 <= 0
x24 +x31 +x34 -x42 <= 1
-x23 <= 0
x23 -x31 -x32 -x34 +x43 <= 0
-x23 +x31 +x32 +x41 <= 1
x23 -x32 +x41 +x43 <= 1
-x24 <= 0
-x23 -x24 +x31 +x32 +x34 +x42 <= 1
-x23 -x24 +x32 +x41 +x42 +x43 <= 1
x23 +x24 -x32 +x43 <= 1
-x24 +x31 +x41 +x42 <= 1
-x23 -x24 +x31 +2x32 +x34 +x41 +2x42 +x43 <= 2
x23 +x24 +x34 -x42 <= 1
-x31 <= 0
-x23 +x34 -x41 -x43 <= 0
x21 +x24 +2x31 +x32 +x34 +x41 -x42 <= 2
-x23 +x32 +x41 +x42 <= 1
x21 +x23 +x31 -x32 +2x41 +x42 +x43 <= 2
-x24 -x31 -x34 +x43 <= 0
-x34 <= 0
-x23 +x32 +x34 +x42 <= 1
x21 +x23 +2x24 +x31 +x32 +2x34 -x42 -x43 <= 2
-x41 <= 0
x21 +2x23 +x24 -x32 -x34 +x41 +x42 +2x43 <= 2
-x24 +x32 +x42 +x43 <= 1
-x43 <= 0
-x24 +x31 +x32 +x42 <= 1
x24 +x32 +x34 -x43 <= 1
x24 -x32 -x41 -x42 <= 0
x24 +x31 +x32 +x34 -x42 -x43 <= 1
x23 -x32 -x34 +x41 +x42 +x43 <= 1
x23 -x31 -x32 -x42 <= 0
x21 -x23 +x31 +x32 +2x41 +x42 +x43 <= 2
x23 -x34 +x42 +x43 <= 1
x21 +x24 +x32 +x34 +x41 +x42 <= 2
x21 -x24 +2x31 +x32 +x34 +x41 +x42 <= 2
x21 +x31 +x41 <= 1
x21 +x23 +x31 +x32 +x42 +x43 <= 2
x21 -x34 +x41 +x43 <= 1
x21 +x31 +x34 -x43 <= 1
x21 +x24 +x34 -x43 <= 1
x21 +x24 +x31 -x42 <= 1
x21 +x23 -x32 +x41 <= 1
x21 +x23 +x24 <= 1
x21 +x23 -x34 +x43 <= 1
x21 +x23 +x24 -x32 -x34 +x43 <= 1
x21 +x23 +x24 +x34 -x42 -x43 <= 1
2x21 +x23 +x24 +x31 +x34 +x41 -x43 <= 2
2x21 +x23 +x24 +x31 -x34 +x41 +x43 <= 2

```

that satisfy the constraint  $(\pi, \pi_0)$  at equality, that is,  $\dim(F) = \dim(\text{conv}(X)) - 1$ , and therefore  $(\pi, \pi_0)$  defines a facet.

For BIPs of large scale, we have to explicitly construct  $\dim(\text{conv}(X))$  affinely independent feasible integer points that satisfy  $(\pi, \pi_0)$  at equality. One way is to make sure a new variable is used in each new solution. Examples of such direct proofs can be found in [10] for the ATSP and VRPs with time windows or precedence



constraints, and [11] for cardinality constrained quadratic knapsack problem and the quadratic selective TSP.

*Mathematical induction* A one-dimensional mathematical induction framework for the Asymmetric Travelling Salesman Problem was proposed in [12], and the concept is essentially to use vertex insertion and can be applied in many cycle-based combinatorial optimization problems. When there are side constraints, however, e.g., cycle-cardinality constraints, a two-dimensional mathematical induction is required, (see e.g., [13] in the context of the Asymmetric Travelling Salesman Problem with Replenishment Arcs (RATSP)).

*Monotone polytope* Let  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  and lies in the positive orthant. The monotone polytope is defined by  $\tilde{P} = \{\tilde{x} \in \mathbb{R}^n : 0 \leq \tilde{x} \leq x, \text{ for some } x \in P\}$ . The monotone polytope is full-dimensional, hence easier to analyze, though one will still need the results for the original polytope that is not full-dimensional. Fortunately, the theoretical result presented in [14] enabled us to translate results from the monotone polytope to the original polytope of an integer program. See also [15] for background mathematics.

**Definition 1** Let  $\dim(\tilde{P}) = n$ ;  $N$  be the index set of variables,  $(\pi, \pi_0)$  be a valid inequality for  $\tilde{P}$ ;  $N^0 = \{j \in N \mid \pi_j = 0\}$ ;  $A_0^-$  be the  $r$  by  $|N^0|$  submatrix of  $A^-$  with only columns with indices in  $N^0$ . The inequality  $(\pi, \pi_0)$  is said to be support reduced with respect to  $P$  if  $A_0^-$  has a full row rank. Further, w.l.o.g., let  $N^r = \{j_1, \dots, j_r\}$  be the index set of the columns of a basis for  $A_0^-$ . If  $(\pi, \pi_0)$  satisfies the condition that for each  $k \in N^0 \setminus N^r$ , if  $N^0 \setminus N^r \neq \emptyset$ , there exists a pair  $x^1, x^2 \in \{x \in P \mid \pi x = \pi_0\}$ , such that

$$\{j_k\} \subseteq \{j \in N \mid x_j^1 \neq x_j^2\} \subseteq N^0 \setminus N^r$$

then  $(\pi, \pi_0)$  is strongly support reduced.

Consider again (9) and the inequality  $x_2 \leq 4$ . We have that  $A^- = [0, 0, 1]$ ,  $N^0 = \{1, 3\}$ , and  $A_0^- = [0, 1]$ . As  $A_0^-$  has a full row rank, it is support reduced. The index set for its column basis is  $N^r = \{3\}$ . Now, let  $F = \{x \in X \mid x_2 = 4\}$  (with  $X$  the feasible set for (9)). Consider  $x^1, x^2 \in F$  with  $x^1 = (0, 4, 3)$  and  $x^2 = (1, 4, 3)$ , we have that  $\{j \in N \mid x_j^1 \neq x_j^2\} = \{1\}$ , which is  $\supseteq \{1\}$  and  $\subseteq N^0 \setminus N^r = \{1\}$ . Hence,  $x_2 \leq 4$  is strongly support reduced.

**Theorem 3.1** ([14]) *Let  $\pi x \leq \pi_0$ , where  $\pi_0 \neq 0$ , be facet-defining for the monotone polytope  $\text{conv}(\tilde{X})$ . If  $\pi x \leq \pi_0$  is strongly support reduced with respect to  $\text{conv}(X)$ , then it is also facet-defining for  $\text{conv}(X)$ .*

Hence,  $x_2 \leq 4$  is facet-defining for  $X$  in (9). An application of this theorem in the context of minimum-span graph labeling problem with integer distance constraints can be found in [16].

## 4 Literature and Research Directions

There are a number of rather good papers that one can consult before embarking on a polyhedral analysis project, e.g., [17] for the  $p$ -cycle polytope, [18] for the Cardinality Constrained Covering Traveling Salesman Problem, [19] on the cycle polytope of a directed graph and its relaxations, [20] on the  $k$ -cycle polytope, [21] on the circuit polytope, [22] on the Capacitated Vehicle Routing Problem, [12] on ATSP, and many more. One should also consult a few very good textbooks on polyhedral studies, e.g., [1, 2].

For polytopes that are not full-dimensional, to generate an alternative set of facets from the complete polyhedral description provided by a software may be a worthwhile exercise. Firstly, some constraints have better structures that make it possible to develop fast separation algorithms. Secondly, some constraints are simply easier to prove that they are facet-defining. Further, one should not give up so quickly and jump straight to (meta-) heuristics just because the full IP did not perform well on a commercial IP solver. One should always explore stronger constraints and experiment with different relaxations and decomposition methods, or even using these as part of a heuristic method.

## References

1. Wolsey, L.A.: *Integer Program*. Wiley, New York (1998)
2. Nemhauser, G.L., Wolsey, L.A.: *Integer and Combinatorial Optimization*. Wiley, New York (1988)
3. cdd+ reference manual. <https://www.math.ucdavis.edu/~deloera/MISC/BIBLIOTECA/trunk/Fukuda/Fukuda1.pdf> (2010). Accessed 30 Sept 2010
4. Porta vers. 1.4.1. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/> (2010). Accessed 30 Sept 2010
5. Lrwald, S., Reinelt, G.: Panda: a software for polyhedral transformations. *EURO J. Comput. Optim.* **3**(4), 297–308 (2015)
6. Motzkin, T.S., Raiffa, H., Thompson, G.L., Thrall, R.M.: *The Double Description Method*. Princeton University Press, Princeton (1953)
7. Mak-Hau, V.: On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *J. Comb. Optim.* 1–25 (2015)
8. Mak-Hau, V.: Polyhedral results for the cardinality constrained multi-cycle problem (CCMcP) and the cardinality constrained cycles and chains problem (CCCCP). In: *Proceedings of 21st International Congress on Modelling and Simulation (MODSIM, Nov 2015)* (2015)
9. Akartunali, K., Mak-Hau, V., Tran, T.: A unified mixed-integer programming model for simultaneous fluence weight and aperture optimization in vmat, tomotherapy, and cyberknife. *Comput. Oper. Res.* **56**, 134–150 (2015)
10. Mak, V., Ernst, A.T.: New cutting-planes for the time-and/or precedence-constrained ATSP and directed VRP. *Math. Methods Oper. Res.* **66**(1), 69–98 (2007)
11. Mak, V., Thomadsen, T.: Polyhedral combinatorics of the cardinality constrained quadratic knapsack problem and the quadratic selective travelling salesman problem. *J. Comb. Optim.* **11**(4), 421–434 (2006)

12. Balas, E.: The asymmetric assignment problem and some new facets of the traveling salesman polytope on a directed graph. *SIAM J. Discret. Math.* **2**(4), 425–451 (1989)
13. Mak, V., Boland, N.: Facets of the polytope of the asymmetric travelling salesman problem with replenishment arcs. *Discret. Optim.* **3**(1), 33–49 (2006). *The Traveling Salesman Problem*
14. Balas, E., Fischetti, M.: On the monotonization of polyhedra. *Math. Program.* **78**(1), 59–84 (1996)
15. Pulleyblank, W.R., Grötschel, M.: Clique tree inequalities and the symmetric travelling salesman problem. *Math. Oper. Res.* **11**(4), 537–569 (1986)
16. Mak, V.: Polyhedral studies for minimum-span graph labelling with integer distance constraints. *Int. Trans. Oper. Res.* **14**(2), 105–121 (2007)
17. Hartmann, M., Özlük, Ö.: Facets of the  $p$ -cycle polytope. *Discret. Appl. Math.* **112**(13), 147–178 (2001). *Combinatorial Optimization Symposium, Selected Papers*
18. Patterson, R., Rolland, E.: The cardinality constrained covering traveling salesman problem. *Comput. Oper. Res.* **30**(1), 97–116 (2003)
19. Balas, E., Stephan, R.: On the cycle polytope of a directed graph and its relaxations. *Networks* **54**(1), 47–55 (2009)
20. Nguyen, V.H., Maurras, J.F.: On the linear description of the  $k$ -cycle polytope. *Int. Trans. Oper. Res.* **8**(6), 673–692 (2001)
21. Bauer, P.: The circuit polytope: facets. *Math. Oper. Res.* **22**(1), 110–145 (1997)
22. Cornuejols, G., Harche, F.: Polyhedral study of the capacitated vehicle routing problem. *Math. Program.* **60**(1), 21–52 (1993)