Solution of an integer linear programming problem via a primal dual method combined with a heuristic

Mohand Ouamer Bibi, Houria Boussouira and Abdelhek Laouar*

Research Unit LaMOS,
Department of Operations Research,
Faculty of the Exact Sciences,
University of Bejaia,
06000 Bejaia, Algeria
Email: mdouamer.bibi@univ-bejaia.dz
Email: houria.boussouira@univ-bejaia.dz
Email: abdelhek.laouar@univ-bejaia.dz
*Corresponding author

Abstract: In this paper, we propose an algorithm for solving integer linear programming (ILP) problem with upper and lower bounded variables where we combined a cutting plane method with a heuristic. At each iteration, a relaxed problem is solved by the adaptive method and its optimal solution is submitted to a judicious rounding procedure. The concept of β-optimality is used to indicate the quality of the approximate solution obtained by this heuristic. In order to compare our method with the intlinprog method of the MATLAB optimisation toolbox, numerical experiments on randomly generated test problems are presented.

Keywords: integer linear programming; ILP; cutting planes; adaptive method; heuristic method; approximate solution; β-optimality.

Reference to this paper should be made as follows: Bibi, M.O., Boussouira, H. and Laouar, A. (2022) 'Solution of an integer linear programming problem via a primal dual method combined with a heuristic', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 12, No. 1, pp.69–87.

Biographical notes: Mohand Ouamer Bibi received his DES degree in Mathematical Analysis (1980) from the USTHB University, Algeria, Master's (1982) and PhD (1986) in Applied Mathematics from the University of Minsk, Belarus. He is a Full Professor at the Department of Operational Research and Leader of the research group 'Optimisation and optimal control' at the LaMOS Research Unit, University of Bejaia, Algeria.

Houria Boussouira received his Engineering degree in Operational Research (2002) and Master's in Applied Mathematics (2009) from the University of Bejaia, Algeria. She is an Associate Professor at the Department of Initial Formation in Economics, Management and Commercial Sciences 'SEGC-LMD', University of Bejaia, Algeria.

Abdelhek Laouar received his Engineering degree in Operational Research (2007) and Master's in Applied Mathematics (2010) from the University of Bejaia, Algeria. He is an Associate Professor at the Department of Initial Formation in Economics, Management and Commercial Sciences 'SEGC-LMD', University of Bejaia, Algeria.

1 Introduction

Integer linear programming (ILP) techniques are closely related to the modelling of problems whose decision variables are restricted to be integer. Although this restriction, ILP is used to model a wide variety of optimisation problems in extremely diverse application areas. This is one of the most active areas of the mathematical programming, and the volume of publications and the research that has been devoted to it since the early work of Gomory (1958) attest the difficulty of the subject and the importance of its applications. Indeed, ILP problems are difficult to solve, since they belong to the class of NP-hard problems (Karp, 1972; Garey and Johnson, 1979).

Integer programming problems have been extensively studied by many researchers (Hu, 1969; Zionts, 1974; Salkin and Morito, 1975; Taha, 1975; Schrijver, 1986; Nemhauser and Wolsey, 1988; Parker and Rardin, 1988; Glover and Sherali, 2005). Hence, several studies and monographs on the integer programming and combinatorial optimisation have been published after the 80 years (Schrijver, 1986; Nemhauser and Wolsey, 1988; Llewellyn and Ryan, 1993) and recently new results are obtained (Schrijver, 2003; Jünger et al.. 2010; Lodi, 2010; Korte and Vygen, 2012); De Loera et al., 2012; Conforti et al., 2014; Fellah and Kechar, 2017; Eisenbrand and Weismantel, 2018; Mei et al., 2018).

There are three main categories of algorithms for solving integer programming problems: exact algorithms including cutting-planes, branch-and-bound and dynamic programming, completed by heuristic algorithms and approximation algorithms (Owen and Mehrotra, 2001; Marchand et al., 2002; Bertacco et al., 2007; Achterberg et al., 2012; Cornuéjols et al., 2013; Dey and Molinaro, 2018). The cutting plane methods have the disadvantage to provide often an integer optimal solution only at the end, thus they may take an exponential number of iterations, contrary to the heuristic methods that provide an approximate solution in a reasonable time, but without a guarantee on its quality.

In this work, we use the adaptive method (Gabasov, 1993; Gabasov et al., 1995; Bibi and Bentobache, 2011, 2015; Djemai et al., 2016) for solving the relaxed problems. It is a primal-dual method that works with a support feasible solution (SFS) $\{x, J_B\}$, where the vector x and the support J_B (defining a basis) are defined independently, unlike a basic feasible solution. Hence, this SFS can scan all parts of the admissible region. This corresponds exactly to the nature of a linear integer programming problem where the maximum can be attained anywhere in the admissible region. By the way, in this algorithm we will use the interior and the active aspects of this method, which is intermediate between the interior points and active set methods (Roos et al., 1997; Wright, 1997; Vial, 1993; Gill et al., 1983; Fletcher, 1987). In addition, we will define the concept of β-optimality, which is an estimate for the quality of an approximate integer feasible solution. The different integer solutions will be obtained by a rounding procedure that is judiciously chosen over the course of the different cuts that will be added to the original relaxed problem. After proving that the added cuts are valid, we have developed an algorithm of resolution that is illustrated on a numerical example.

This article is organised as follows: in Section 2 the problem is stated with some definitions. A review on the adaptive method is given in Section 3. A rounding heuristic will be the subject of Section 4, followed by a valid cut generating process. These two elements will be incorporated in the resolution algorithm described in Section 5 with a numerical example described in details. At the end, we finish with numerical experiments on generated random ILP problems and a conclusion.

2 Problem and definitions

Let us consider an ILP problem with bounded variables, which is written in the following canonical form:

$$(ILP)\begin{cases} \max z = c'x, \\ Ax \le b, \\ l \le x \le u, x \in \mathbb{Z}^n, \end{cases} \tag{1}$$

where $A \in \mathbb{Z}^{m\times n}$ is an $(m \times n)$-matrix, with $rankA = m < n$; $b \in \mathbb{Z}^m$; c, x, l and u are vectors in $\mathbb{Z}^n$. The symbol (′) represents the transposition operation.

We set

$I = \{1, 2, \dots, m\}$ the rows index set of the matrix A

$J = \{1, 2, \dots, n\}$ the columns index set of the matrix A, with $A = (a_{ij}, i \in I, j \in J) = (a_j, j \in J)$

$J_s = \{1, 2, \dots, n, n + 1, \dots, n + m\}$ the columns index set of the matrix $(A| I_m) = (a_{ij}, i \in I, j \in J_s) = (a_j, j \in J_s)$, where a_j is the j^{th} column, I_m is an identity

m – matrix.

So we consider the associated relaxed linear programming problem in standard form:

$$(PL)\begin{cases} \max z = c'x, \\ Ax + +I_m x^s = b, \\ l \le x \le u, l^s \le x^s \le u^s, x \in \mathbb{R}^n, x^s \in \mathbb{R}^m, \end{cases} \tag{2}$$

where $x = (x_1, \dots, x_n)'$, $x^s = (x_{n+1}, \dots, x_{n+m})'$, $l^s = (l_{n+1}, \dots, l_{n+m})' = 0$, $u^s = (u_{n+1}, \dots, u_{n+m})' \in \mathbb{Z}^m$, with

$$u^s_{n+i} = b_i - \sum_{j\in J, a_{ij}>0} a_{ij} l_j - \sum_{j\in J, a_{ij}<0} a_{ij} u_j, i \in I.$$

We give the following definitions:

- A vector x verifying the constraints $Ax \le b$, $l \le x \le u$, $x \in \mathbb{Z}^n$, is a feasible solution of the problem (1). The feasible solutions set is given by:

  $$S^Z = \{x \in \mathbb{Z}^n : Ax \le b, l \le x \le u\}.$$

  Similarly, we define by

  $$S = \{y = (x, x^s) = (x_j, j \in J_s) \in \mathbb{R}^{n+m} : Ax + I_m x^s = b, l \le x \le u, l^s \le x^s \le u^s\},$$

  The feasible solutions set of the relaxed problem (2).

- Let $y^0 = (x^0, x^{s0})$ be an optimal solution for the problem (2). A feasible solution $y^\epsilon = (x^\epsilon, x^{s\epsilon})$ is said to be ϵ-optimal or suboptimal if:

  $$z(y^0) - z(y^\epsilon) = c'x^0 - c'x^\epsilon \le \epsilon,$$

where ϵ is a non-negative number, chosen as an accuracy.

Let $J_B \subset J_s$ be a subset of indices such that $J_s = J_B \cup J_N$, $J_B \cap J_N = \varnothing$, $|J_B| = m$. In virtue of this partition, we can write and partition vectors and matrices as follows:

$$y = \begin{pmatrix} y_B \\ y_N \end{pmatrix}, \; y_B = (x_j,\, j \in J_B),\; y_N = (x_j,\, j \in J_N);$$

$$k = (c, c^s) = (c_j,\, j \in J_s) = \begin{pmatrix} c_B \\ c_N \end{pmatrix}, \; c_B = (c_j,\, j \in J_B),\; c_N = (c_j,\, j \in J_N),$$

where $c^s = (c_j, j = n + 1, \cdots, n + m) = 0 \in \mathbb{Z}^m$;

$$(A|I_m) = A(I, J_s) = (A_B|A_N),\; A_B = (a_j,\, j \in J_B),\; A_N = (a_j,\, j \in J_N).$$

The subset J_B is called a support if $det A_B \neq 0$. The couple $\{y, J_B\}$ formed from the feasible solution y and the support J_B is called a SFS. An SFS $\{y, J_B\}$ is non-degenerate if $l_j < x_j < u_j$, $\forall j \in J_B$. It is called basic if $x_j = l_j \vee u_j$, $\forall j \in J_N$.

3 Solution of the relaxed problem by the adaptive support method

3.1 Resolution algorithm FOR the relaxed problem

Let $\{y, J_B\}$ be a SFS of the problem (2) and let's consider any other feasible solution $\bar{y} = y + \Delta y$, $\bar{y} = (\bar{x}, \bar{x}^s)$. The increment of the objective function is (Bibi and Bentobache, 2015):

$$\Delta z = z(\bar{y}) - z(y) = c'\bar{x} - c'x = -E'_N \Delta y_N,$$

where

$$E' = (E'_B, E'_N) = \pi'(A|I_m) - k' \Leftrightarrow E_j = \pi' a_j - c_j,\; j \in J_s, \tag{3}$$

With $\pi' = c'_B A_B^{-1}$, $E'_B = 0$ and $E'_N = \pi' A_N - c'_N$. The vector E is called a reduced costs vector.

Then we have the following optimality criterion:

Theorem 1: Gabasov (1993)

Let $\{y, J_B\}$ be a SFS for the problem (2). Then the relations:

$$\begin{cases} E_j \geq 0, \text{ for } x_j = l_j, \\ E_j \leq 0, \text{ for } x_j = u_j, \\ E_j = 0, \text{ for } l_j < x_j < u_j,\; j \in J_N, \end{cases} \tag{4}$$

are sufficient, and in the case of non-degeneracy also necessary, for the optimality of the SFS $\{y, J_B\}$.

Remark 1: If the SFS $\{y, J_B\}$ is basic, then the optimal relations (4) take the form:

$$\begin{cases} E_j \geq 0, \text{ for } x_j = l_j, \\ E_j \leq 0, \text{ for } x_j = u_j,\ j \in J_N. \end{cases}$$

The number

$$\beta(y, J_B) = \sum_{E_j > 0\, j \in J_N} E_j(x_j - l_j) + \sum_{E_j < 0\, j \in J_N} E_j(x_j - u_j) \tag{5}$$

Is called a suboptimality estimate of the SFS $\{y, J_B\}$ and we have always the inequality: $z(y^0) - z(y) \leq \beta(y, J_B)$. So, if $\beta(y, J_B) \leq \epsilon$, then y is an ϵ-optimal solution for the problem (2).

Let $\{y, J_B\}$ be an initial SFS of the problem (2) and $\beta(y, J_B) > \epsilon$. An iteration of the adaptive method algorithm (Gabasov, 1993; Bibi and Bentobache, 2015) consists in getting a new SFS $\{\bar{y}, \bar{J}_B\}$ such that $z(\bar{y}) \geq z(y)$ and $\beta(\bar{y}, \bar{J}_B) \leq \beta(y, J_B)$. For this purpose, we calculate an improvement direction $d \in \mathbb{R}^{n+m}$ and a step $\theta^0 \geq 0$ verifying $\bar{y} = y + \theta^0 d$. Thus, we set:

$$\begin{cases} d_j = l_j - x_j, & \text{if } E_j > 0, \\ d_j = u_j - x_j, & \text{if } E_j < 0, \\ d_j = 0, & \text{if } E_j = 0, j \in J_N, \\ d_B = d(d_B) = & -A_B^{-1} A_N d_N. \end{cases} \tag{6}$$

$$\begin{cases} \theta^0 = \min\{1, \theta_{j_1}\}, \text{ with } \theta_{j_1} = \min_{j \in J_B} \theta_j, \text{ where} \\ \qquad \theta_j = \begin{cases} \dfrac{u_j - x_j}{d_j}, & \text{if } d_j > 0, \\ \dfrac{l_j - x_j}{d_j}, & \text{if } d_j < 0, \\ \infty, & \text{if } d_j = 0. \end{cases} \end{cases} \tag{7}$$

Two cases can arise:

1 $\theta^0 = 1$: The feasible solution $\bar{y} = y + d$ is optimal and the resolution process is stopped.

2 $\theta^0 = \theta_{j_1} < 1$: in this case, the suboptimality estimate of the new feasible solution $\{\bar{y}, J_B\}$ is equal to:

$$\beta(\bar{y}, J_B) = (1 - \theta^0)\beta(y, J_B).$$

If $\beta(\bar{y}, J_B) \leq \epsilon$, the new solution is then ϵ-optimal and the process is stopped. Otherwise, we will change the support J_B. For this, we construct the pseudo-solution $\kappa = y + d$ and we set $\alpha_0 = \kappa_{j_1} - \bar{x}_{j_1}$. The dual direction t is constructed as follows:

$$\begin{cases} t_{j_1} = sign\alpha_0, t_j = 0, j \neq j_1, j \in J_B, \\ t'_N = t'_B A_B^{-1} A_N. \end{cases}$$

We compute the dual step $\sigma^0 = \sigma_{j_0} = \min_{j \in J_N} \sigma_j$, where

$$\sigma_j = \begin{cases} \frac{-E_j}{t_j}, & \text{if } E_j t_j < 0; \\ 0, & \text{if } E_j = 0, t_j < 0; \\ \infty, & \text{Otherwise} \end{cases}$$

We set $\bar{J}_B = (J_B \setminus j_1) \cup j_0$. The suboptimality estimate value becomes:

$$\beta(\bar{y}, \bar{J}_B) = (1-\theta^0)\beta(y, J_B) - \sigma^0 |\alpha_0|.$$

If $\beta(\bar{y}, \bar{J}_B) > \epsilon$, then we start a new iteration by setting $y := \bar{y}$, and $J_B := \bar{J}_B$.

3.2 Getting an optimal basic solution from an optimal support solution

Let $\{y^0 = (x^0, x^{s0}), J_B\}$ be an optimal support solution of the problem (2), satisfying the optimality sufficient conditions (4). If $\{y^0, J_B\}$ is not basic, then it is not unique. During the cut construction, a special case can arise, where we must have an optimal basic solution. For this, according to the signs of the reduced costs vector E_N, we define the following sets of indices:

$$J_N^l = \{j \in J_N : E_j \geq 0 \text{ and } x_j^0 = l_j\}, J_N^u = \{j \in J_N : E_j \leq 0 \text{ and } x_j^0 = u_j\},$$
$$J_N^{lu} = \{j \in J_N : E_j = 0 \text{ and } l_j < x_j^0 < u_j\}, J_N = J_N^l \cup J_N^u \cup J_N^{lu}.$$

We construct another feasible solution $\bar{y}$ such that $z(\bar{y}) = z(y^0)$, with $\bar{y} = y + \theta^0 d$ and

$$d_j = \begin{cases} 0, & \text{if } j \in J_N^l \cup J_N^u, \\ l_j - x_j^0, & \text{if } j \in J_N^{lu} \text{ and } 0 < x_j^0 - l_j \leq u_j - x_j^0, \\ u_j - x_j^0, & \text{if } j \in J_N^{lu} \text{ and } 0 < u_j - x_j^0 < x_j^0 - l_j, \end{cases}$$

and $d_B = -A_B^{-1} A_N d_N = -A_B^{-1} \sum_{j \in J_N} a_j d_j$.

The step θ^0 is computed according to formula (7). Two cases can arise:

- if $\theta^0 = 1$: $\bar{y} = y^0 + d$. In this case, we have: $\bar{x}_j = l_j \vee u_j, j \in J_N$, and

  $$z(\bar{y}) - z(y^0) = -\theta^0 \sum_{j \in J_N} E_j d_j = -\theta^0 \sum_{j \in J_N^{lu}} E_j d_j = 0.$$

  Therefore, $\bar{y}$ is another optimal basic solution.

- $\theta^0 = \theta_{j1} < 1: \bar{y} = y^0 + \theta_{j_1} d$, with $\bar{x}_{j_1} = l_{j_1} \vee u_{j_1}$.

  In this case, we repeat the process with the SFS $\{\bar{y}, \bar{J}_B\}$, such that $\bar{J}_B = \{J_B \setminus j_1\} \cup \{j_0\}$ and $\bar{J}_N = \{J_N \setminus j_0\} \cup \{j_1\}$, where j_0 is the index given by the dual iteration.

So the new feasible solution $\overline{y}$ will have a new non-basic critical component at each iteration. This process provides an optimal basic solution after $|J_N|$ iterations at most.

4 Rounding heuristic

4.1 Rounding procedure

Let $y^0 = (x^0, x^{s0})$ be an optimal solution for the problem (2), where x^0 is not integer, and $x^{s0} = b - Ax^0$.

Then we set

$$x_j^0 = \left[x_j^0\right] + \alpha_j, \text{ with } 0 \le \alpha_j < 1,\ j \in J = \{1, 2, ..., n\},$$

where $\left[x_j^0\right]$ is the integer part and $\alpha_j = \left\{x_j^0\right\}$ the fractional part.

In order to obtain an integer feasible solution for the problem (1), we propose a well appropriate rounding process described as follows:

We set $M = z(x^0)$ and $m = z(\hat{x})$, where $\hat{y} = (\hat{x}, \hat{x}^s)$ verifies $z(\hat{y}) = \min_{y \in S} z(x) = z(\hat{x})$.

We define the following sets of indices:

$$J_c^+ = \left\{j \in J : c_j \ge 0\right\} \text{ and } J_c^- = \left\{j \in J : c_j < 0\right\}.$$

1 Rounding of x^0

(A_1) Rounding x^1 according to the variable. We set:

$$x^1 = \left(x_j^1,\ j \in J\right),\ x_j^1 = \begin{cases} \left|x_j^1\right|, & \text{if } 0 \le \alpha_j \le \frac{1}{2}, \\ \left|x_j^0\right| + 1, & \text{if } \frac{1}{2} \le \alpha_j < 1. \end{cases}$$

if x^1 is not feasible for the problem (1), we round according to the function z.

(A_2) Rounding x^2 according to the function z. In this case, we set:

$$x^2 = \begin{cases} \left|x^0\right|, & \text{if } z(\alpha) = \sum_{j \in J} c_j \alpha_j \ge 0, \\ \left|x^0\right| + e, & \text{if } z(\varepsilon) = \sum_{j \in J} c_j \le z(\alpha) < 0, \end{cases}$$

where $e = (1, 1, \ldots, 1' \in \mathbb{R}^n$ and $\alpha = (\alpha_j, j \in J)$:

Else, if this round is not feasible, we set:

$$x^2 = \left(x_j^2,\ j \in J\right),\ x_j^2 = \begin{cases} \left|x_j^0\right|, & j \in J_c^+, \\ \left|x_j^0\right| + 1, & j \in J_c^-. \end{cases}$$

If x^2 is not feasible, we round according to the middle x^m.

2 (A_3) Rounding of the middle x^m. We set $x^m = \frac{1}{2}(x^0 + \hat{x})$ and we round x^m to x^3 as follows:

$$x^3 = \left(x_j^3,\, j \in J\right),\, x_j^3 = \begin{cases} \lfloor x_j^m \rfloor + 1, & j \in J_c^+, \\ \lfloor x_j^m \rfloor, & j \in J_c^-. \end{cases}$$

If this round is not feasible, we set

$$x^3 = \begin{cases} \lfloor x^m \rfloor + 1, & \text{if } |J_c^+| \geq |J_c^-|, \\ \lfloor x^m \rfloor, & \text{if } |J_c^+| < |J_c^-|. \end{cases}$$

If this last round is not feasible, we set

Remark 2: The round x^3 has a greater probability to be feasible, because x^m is situated in the middle of the polytope S or in the middle of one of its faces.

Remark 3: Although at the end of the rounding step, we do not get any feasible integer solution, we continue the cutting plane algorithm until obtaining a satisfactory feasible rounded solution or an integer optimal solution.

Remark 4: If a round x^a verifies $z(x^a) > z(x^0)$, then x^a is not feasible.

Lemma 2: In the case of the round x^2 according to the function z, we have always $z(x^2) \leq z(x^0)$.

Indeed, if $z(\alpha) \geq 0$, we have $x^2 = [x^0]$ and

$$z\left(x^0\right) = \sum_{j \in J} c_j x_j^0 = \sum_{j \in J} c_j \left[x_j^0\right] + \sum_{j \in J} c_j \alpha_j = z\left(x^2\right) + z(\alpha) \geq z\left(x^2\right).$$

If $z(e) \leq z(\alpha) < 0$, we have $x^2 = [x^0] + e$. So

$$z\left(x^0\right) = \sum_{j \in J} c_j \left[x_j^0\right] + \sum_{j \in J} c_j \alpha_j \geq \sum_{j \in J} c_j \left[x_j^0\right] + \sum_{j \in J} c_j = \sum_{j \in J} c_j \left[x_j^0\right] + z(e) = z\left(x^2\right).$$

In the second case, we have

$$z\left(x^0\right) = \sum_{j \in J_c^+} c_j x_j^0 + \sum_{j \in J_c^-} c_j x_j^0 \geq \sum_{j \in J_c^+} c_j \left[x_j^0\right] + \sum_{j \in J_c^-} c_j \left(\left[x_j^0\right] + 1\right) = z\left(x^2\right). \square$$

We illustrate the rounding process on a typical ILP example, where all the neighbours rounds of the optimal solution are not feasible, but the middle is.

Example 4.1: Consider the following linear integer programming problem:

$$\begin{cases} \max\ z = x_1 + x_2, \\ \qquad 14x_1 + 9x_2 \leq 51, \\ \qquad -6x_1 + 3x_2 \leq 1, \\ \qquad x_j \geq 0,\, x_j \ \textit{integer},\ j = 1, 2. \end{cases} \tag{8}$$

The optimal solution after solving the relaxed problem is:

$$x^0 = \left(\frac{3}{2}, \frac{10}{3}\right)', \text{ with } z(x^0) = \frac{29}{6}.$$

Since this solution x0 is not integer, then we apply the rounding process for x^0.

- The round $x^1 = (1, 3)$ according to the variable is not feasible.
- The round according to the function z is $x^2 = (1, 3) = x^1$, which is also not feasible.

The other possible neighbours rounds x^a of the optimal solution x^0 are: (2, 3), (1, 4) and (2, 4), which are not feasible, since $z(x^a) > z(x^0)$. Therefore, we go to the middle round x^m: $x^m = \frac{1}{2}(x^0 + \hat{x})$, where $\hat{x} = (0, 0)$, $x^m = \left(\frac{3}{4}, \frac{5}{3}\right)$. The obtained round is $x^3 = (1, 2)$ which is an integer feasible solution for the problem (8), close enough to the integer optimal solution $x^e = (2, 2)$, with $z(x^e) = 4$ and $z(x^3) = 3$.

4.2 Notion of β-optimality

The concept of β-optimality allows to indicate the quality of an approximate integer solution. Let's x^0 be an optimal solution for the problem (2) and x^{0e} an optimal solution for the problem (1). Then the β-optimality estimate is defined as follows:

$$\beta(x^{0a}) = \frac{z(x^0) - z(x^{0a})}{M - m}, \tag{9}$$

where x^{0a} is an integer feasible solution, obtained eventually after the rounding process of x^0, M and m being defined above.

As the difference between $z(x^0)$ and $z(x^{0e})$ cannot be infinitely small, we reduce this difference by dividing on the quantity $M - m$. Hence, we will have:

$$0 \le \frac{z(x^{0e}) - z(x^{0a})}{M - m} \le \frac{z(x^0) - z(x^{0a})}{M - m} = \beta(x^{0a}) \le 1.$$

Thus, the round x^{0a} is called β-optimal if $\beta(x^{0a}) \le \beta$, where the non-negative real $\beta < 1$ is chosen as an accuracy.

5 ILP resolution algorithm

The purpose of this algorithm is to construct an optimal or an approximate solution for the problem (1). The principle is similar to that of cutting planes algorithms, but in this work we associate in addition a heuristic that tries to get with a high probability an integer feasible solution, for which the β-optimality estimate is evaluated in order to quantify its quality. Thus, we first solve the relaxed problem (2) by the adaptive method (Gabasov et al., 1995), obtaining therefore an optimal support solution $\{y^0, J_B\}$, with $y^0 = (x^0, x^{s0})$. If the vector x^0 contains only integer components, then x^0 is an optimal solution of the problem (1) and the resolution process is stopped. Otherwise, we apply the rounding process described above and calculate the β-optimality estimate, if the round x0a is feasible. If $\beta(x^{0a}) \leq \beta$, where β is chosen in advance as an accuracy, then the algorithm is stopped. Otherwise, we must generate a valid cut. To do this, according to (3) we define the following sets of indices:

$$J_N^+ = \{j \in J_N : E_j > 0\}, J_N^- = \{j \in J_N : E_j < 0\},$$
$$J_N^0 = \{j \in J_N : E_j = 0\} = J^+ \cup J^-,$$
$$J^+ = \{j \in J_N^0 ; x_j^0 - l_j \leq u_j - x_j^0\}, \text{and } J^- = \{j \in J_N^0 ; u_j - x_j^0 < x_j^0 - l_j\}.$$

Then we construct the functions:

$$Z_i(y) = \left\{\tilde{b}_i - \sum_{j \in J_N^+ \cup J^+} x_{ij} l_j - \sum_{j \in J_N^- \cup J^-} x_{ij} u_j\right\} + \sum_{j \in J_N^+ \cup J^+} \{x_{ij}\}(x_j - l_j) + \sum_{j \in J_N^- \cup J^-} \{-x_{ij}\}(x_j + u_j), i \in J_B, \tag{10}$$

where $\tilde{b} = A_B^{-1} b$ and $X = (x_{ij}, i \in J_B, j \in J_N) = A_B^{-1} A_N$.

Proposition 3: Gabasov and Kirillova (1980)

For all integer SFS $\{y^e, J_B\}$ of the problem (2), we have $Z_i(y^e) \geq 0$, $\forall i \in J_B$. In addition, $Z_i(y^e)$ is an integer number for all $i \in J_B$.

Proposition 4: Let $\{y^0, J_B\}$ be an optimal SFS of the problem (2). If there is an index $i_1 \in J_B$ such that $Z_{i_1}(y^0) < 0$, then the inequality gives a valid cut.

$$Z_{i_1}(y) \geq 0, y \in \mathbb{R}^{n+m},$$

Proof 1: Evident, because in virtue of Proposition 3, all integer feasible solutions y^e for the problem (2) verify $Z_i(y^e) \geq 0$, $\forall i \in J_B$, in particular $Z_{i_1}(y^e) \geq 0$. □

Proposition 5: When the support optimal solution $\{y^0, J_B\}$ is basic and not integer, then $\exists\, i_1 \in J_B$ such that $Z_{i_1}(y^0) < 0$.

Proof 2: Let $\{y^0, J_B\}$ be an optimal basic feasible solution for the problem (2). Then in this case, we have:

$$x_j^0 = \begin{cases} l_j, & j \in J_N^+ \cup J^+, \\ u_j, & j \in J_N^- \cup J^-. \end{cases}$$

Let $i_1 \in J_B$ be an index such that the component $x_{i_1}^0$ is not integer, with $x_{i_1}^0 = [x_{i_1}^0] + \{x_{i_1}^0\}, 0 < \{x_{i_1}^0\} < 1$. Then the expression of $Z_{i_1}(y^0)$ takes the following form:

$$Z_{i_1}(y^0) = \left\{\tilde{b}_{i_1} - \sum_{j\in J_N^+ \cup J^+} x_{i_1 j} l_j - \sum_{j\in J_{\overline{N}} \cup J^-} x_{i_1 j} u_j\right\} + \sum_{j\in J_N^+ \cup J^+} \{x_{i_1 j}\}(l_j - l_j)$$
$$+ \sum_{j\in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\}(-u_j + u_j),$$
$$= -\left\{\tilde{b}_{i_1} - \sum_{j\in J_N^+ \cup J^+} x_{i_1 j} l_j - \sum_{j\in J_{\overline{N}} \cup J^-} x_{i_1 j} u_j\right\}.$$

On other hand, we have:

$$(A|In_m)\, y^0 = b \Leftrightarrow A_B y_B^0 + A_N y_N^0 = b,$$
$$so\ y_B^0 = A_B^{-1} b - A_B^{-1} A_N y_N^0 = \tilde{b} - X y_N^0 \Rightarrow \tilde{b} - y_B^0 + X y_N^0.$$

Hence, the component $\tilde{b}_{i_1}$ is equal to:

$$\tilde{b}_{i_1} = x_{i_1}^0 + \sum_{j\in J_N^+ \cup J^+} x_{i_1} l_j + \sum_{j\in J_{\overline{N}} \cup J^-} x_{i_1} u_j,$$

From where we get

$$Z_{i_1} = (y^0) = -\{x_{i_1}^0\} < 0. \ \square$$

Corrolary 6: Let $\{y^0, J_B\}$ be an optimal SFS for the problem (2). If y^0 is not integer and such that $Z_i(y^0) \geq 0$, $\forall i \in J_B$, then y^0 is necessarily a non-basic optimal feasible solution.

Proof 3: Evident, according to the Proposition 5.

Remark 5: In the case of the hypothesis of Corollary 6, i.e., $Z_i(y^0) \geq 0$, $\forall i \in J_B$, we use the procedure of the Subsection 3.2 in order to obtain an optimal basic feasible solution.

Proposition 7: Let $\{y^0 = (x^0, x^{s0}),\ J_B^0\}$ be an optimal support solution for the problem (2), where x^0 is not integer and there exists $i_1 \in J_B^0$ such that $Z_{i_1}(y^0) < 0$. Let $y^* = (x^*, x^{s*})$ be a feasible solution for the augmented problem (2) with the added cut $Z_{i_1}(y) \geq 0$. Then the pair $(\overline{y}, \overline{J}_B)$, where

$$\overline{y} = (\overline{y}, 0) = \left(y^* + \lambda(y^0 - y^*), 0\right) \text{ and } \overline{J}_B = J_B^0 \cup \{n+m+1\},$$

is a SFS for the augmented problem (2), where the parameter λ is such that $Z_{i_1}(\overline{y}) = 0$ and $0 \leq \lambda < 1$.

Proof 4: The vector $\overline{y}$ verifies the constraint $Z_{i_1}(\overline{y}) \geq 0$, since by construction the step λ is such that $Z_{i_1}(\overline{y}) = 0$. Let us show that the parameter λ is such that $0 \leq \lambda < 1$. Indeed, we have:

$$Z_{i_1}(\overline{y}) = \left\{ \tilde{b}_{i_1} - \sum_{j \in J_N^+ \cup J^+} x_{i_1 j} l_j - \sum_{j \in J_{\overline{N}} \cup J^-} x_{i_1 j} u_j \right\} + \sum_{j \in J_N^+ \cup J^+} \{x_{i_1 j}\} \left(x_j^* + \lambda \left(x_j^0 - x_j^* \right) - l_j \right)$$

$$+ \sum_{j \in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\} \left(-x_j^* - \lambda \left(x_j^0 - x_j^* \right) + u_j \right) = 0, \quad (11)$$

$$Z_{i_1}(\overline{y}) = Z_{i_1}(y^*) + \lambda \left(\sum_{j \in J_N^+ \cup J^+} \{x_{i_1 j}\} \left(x_j^0 - x_j^* \right) - \sum_{j \in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\} \left(x_j^0 - x_j^* \right) \right) = 0.$$

On the other hand, we use the definition of the function Z_{i_1} and obtain:

$$\begin{aligned} Z_{i_1}(y^0) - Z_{i_1}(y^*) &= \sum_{j \in J_N^+ \cup J^+} \{x_{i_1 j}\} \left(x_j^0 - l_j \right) + \sum_{j \in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\} \left(-x_j^0 + u_j \right) \\ &\quad - \sum_{j \in J_N^+ \cup J^+} \{x_{i_1 j}\} \left(x_j^* - l_j \right) - \sum_{j \in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\} \left(-x_j^* + u_j \right) \\ &= \sum_{j \in J_N^+ + J^+} \{x_{i_1 j}\} \left(x_j^0 - x_j^* \right) - \sum_{j \in J_{\overline{N}} \cup J^-} \{-x_{i_1 j}\} \left(x_j^0 - x_j^* \right). \end{aligned}$$

From (11), finally we get:

$$\lambda = \frac{Z_{i_1}(y^*)}{Z_{i_1}(y^*) - Z_{i_1}(y^0)}.$$

Since $Z_{i_1}(y^*) \geq 0$ and $Z_{i_1}(y^0) < 0$, then λ satisfies the condition $0 \leq \lambda < 1$.

In addition, $\tilde{y}$ also belongs to S. Indeed, from the convexity of S, it follows that:

$$y^0 \in S,\ y^* \in S \text{ and } \lambda \in [0,1] \Rightarrow \tilde{y} = \lambda y^0 + (1-\lambda) y^* \in S.$$

Furthermore, $\overline{J}_B$ is well a support, since

$$\det A\left(I \cup \{m+1\}, J_B\right) = \det A\left(I \cup \{m+1\}, J_B^0 \cup \{n+m+1\}\right) = -\det A\left(I, J_B^0\right) \neq 0.$$

Therefore, $\{\overline{y}, \overline{J}_B\}$ is a SFS for the augmented problem (2) including the constraints: $Z_{i_1}(y) - x_{n+m+1} = 0$ and $0 = l_{n+m+1} \leq x_{n+m+1} \leq u_{n+m+1}$. □

Algorithm 1 The general scheme of this algorithm is described as follows:

Step 1: Set $t = 0$, $I_t = I = \{1, 2, \ldots, m\}$ and $J_s^t = J_s = \{1, 2, \ldots, n+m\}$.

Let $\{y = (x, x^s), J_B\}$ be an initial SFS for the problem (2). Then we construct an optimal support solution $\{y^t = (x^t, x^{st}), J_B^t\}$ by the adaptive method. If $x^t \in \mathbb{Z}^n$, stop the algorithm: x^t is an optimal solution for the problem (1). Otherwise, go to step 2.

Step 2: Application of the rounding process described in Section 4. Let x^{ta} be the resulting round.

If x^{ta} is feasible, we calculate the β-optimality estimate defined as follows:

$$\beta(x^{ta}) = \frac{-z(x^t) - z(x^{ta})}{M - m}.$$

If $\beta(x^{ta}) \leq \beta$ stop, x^{ta} is β-optimal.

Else Set $x^{t*} = x^{ta}$, go to the step 3.

Else We set $x^{t*} = x^m$ and go to step 3.

Step 3: We construct the functions $Z_i(y^t)$, for all $i \in J_B^t$.

Two cases can occur:

1 $\exists i_1 \in J_B^t : Z_{i_1}(y^t) < 0$. In this case the inequality

$$Z_{i_1}(y) \geq 0 \tag{12}$$

gives a valid cut.

We solve the following augmented problem, after adding the constraint (12):

$$\begin{cases} \max z = c'x, \\ Ax + x^s = b, \\ Z_{i_1}(y) - x_{n+m+t+1} = 0, \\ l_j \leq x_j \leq u_j,\ j \in J_s^t. \end{cases} \tag{13}$$

We start with the SFS $\{\bar{y}^t, \bar{J}_B^t\}$, where

$$\bar{y}^t = \left(y^{t*} + \lambda\left(y^t - y^{t*}\right), 0\right), 0 \leq \lambda \leq 1, y^{t*} = \left(x^{t*}, x^{st*}\right), x^{st*} = b - Ax^{t*},$$

and λ will be found from the equation $Z_{i_1}(\bar{y}^t) = 0$.

We take the support $\bar{J}_B^+ = J_B^+ \cup \{n+m+t+1\}$.

We increment t, $t = t + 1$.

We set $I_t = I_{t-1} \cup \{m + t\}$ and $J_s^t = J_s^{t-1} \cup \{n+m+t\}$.

Let $\{y^t, J_B^t\}$ be an optimal support solution of (13). Based on the latter, the process is repeated like with $\{y^{t-1}, J_B^{t-1}\}$.

2 $Z_i(y^t) \geq 0, \forall i \in J_B^t$: the vector y^t is not basic (the optimal solution y^t is not unique). In this case, using the procedure 3.2, we obtain an optimal basic solution y^{tB}, and construct a regular cut as in the case (1).

Example 5.1: Let's solve the following ILP problem by the proposed method:

$$\begin{cases} \max z = 3x_1 + 5x_2 + 4x_3 + 2x_4, \\ \qquad 2x_1 + 7x_2 + 3x_3 \leq 18, \\ \qquad x_1 + 3x_2 + 2x_3 + x_4 \leq 14, \\ \qquad 3x_2 + 2x_2 + 2x_4 \leq 11, \\ \qquad 0 \leq x_j \leq 10,\ x_j \in \mathbb{N},\ j = \overline{1, 4}. \end{cases} \tag{14}$$

The associated relaxed program written in standard form is:

$$\begin{cases} \max z = 3x_1 + 5x_2 + 4x_3 + 2x_4, \\ \qquad 2x_1 + 7x_2 + 3x_3 + x_5 = 18, \\ \qquad x_1 + 3x_2 + 2x_3 + x_4 + x_6 = 14, \\ \qquad 3x_1 + 2x_2 + 2x_4 + x_7 = 11, \\ \qquad 0 \le x_j \le 10, \ j = \overline{1, 4}. \\ \qquad 0 \le x_5 \le 18, 0 \le x_6 \le 14, 0 \le x_7 \le 11. \end{cases} \tag{15}$$

Step 1 Solution of the relaxed problem by the adaptive method.

Let $y = (x, x^s) = (0, 0, 0, 0, 18, 14, 11)'$ be an initial feasible solution of (15), with $z(x) = 0$. We take the support $J_B = \{1, 2, 3\}$ and construct the optimal support solution:

$$y_0 = \left(x^0 \ x^{s0}\right) = \left(\frac{21}{11}, 0, \frac{59}{11}, \frac{29}{11}, 0, 0, 0\right)', J_B^0 = \{1, 4, 3\},$$

$$M = z(y^0) = z(x^0) = \frac{329}{11} \simeq 29{,}909.$$

In addition, we have $\hat{y} = (\hat{x}, \hat{x}^s) = (0, 0, 0, 0, 18, 14, 11)'$, $m = z(\hat{y}) = 0$. The vector $x^0 = \left(\frac{21}{11}, 0, \frac{52}{11}, \frac{29}{11}\right)'$ being non-integer, the rounding process is applied on x^0. The second round according to the function z is then feasible, with $x^{0a} = (1, 0, 4, 2)$, $x^{s0a} = (4, 3, 4)$, $y^{0a} = (x^{0a}, x^{s0a})$, $z(x^{0a}) = z(y^{0a}) = 23$. The β-optimality estimate is:

$$\beta\left(x^{0a}\right) = \frac{z\left(x^0\right) - z\left(x^{0a}\right)}{M - m} = \frac{76}{329} \simeq 0.231.$$

The round x^{0a} is not satisfactory, so we set $x^{0*} = x^{0a}$, $y^{0*} = (x^{0a}, x^{s0a})$.

Step 2 We add to program (15) a cut.

Iteration 1: The reduced costs vector is:

$$E_N^0 = \left(E_2^0, E_5^0, E_6^0, E_7^0\right)' = \left(\frac{27}{11}, \frac{4}{11}, \frac{16}{11}, \frac{3}{11}\right)', J_N^0 = \{2, 5, 6, 7\}.$$

According to the signs of $E_j^0, j \in J_N^0$, the set J_N^0 is partitioned as follows:

$$J_N^+ = \left\{j \in J_N^0 : E_j^0 > 0\right\} = \{2, 5, 6, 7\}, J_N^+ = J^+ = J^- = \emptyset.$$

We then calculate the value of the functions $Z_i(y^0)$, $i \in J_B^0$:

$$Z_B\left(y^0\right) = \left(-\frac{10}{11}, \frac{7}{11}, \frac{8}{11}\right)' \Rightarrow i_1 = 1.$$

$$Z_{i_1}(y) = Z_1(y) = 5x_6 + 4x_5 + 5x_2 + 3x_7 - 10.$$

We solve the augmented program (15), with the added constraint $Z_1(y) \geq 0$, starting by the following initial solution:

$$\overline{y}^0 = \left(y^{0*} + \lambda\left(y^0 - y^{0*} \right), x_8 \right),$$

where λ is calculated from the equation $Z_1(\overline{y}^0) = 0$. So

$$\lambda = \frac{33}{43} \text{ and } \overline{y}^0 = \left(\frac{83}{43}, \frac{10}{43}, \frac{176}{43}, \frac{87}{43}, \frac{10}{43}, \frac{50}{43}, \frac{30}{43}, 0 \right)'.$$

With the slack variable x_8, the initial SFS $\{\overline{y}^0, \overline{J}_B^0\}$ for the augmented problem will be equal to:

$$\overline{y}^0 = \left(\frac{83}{43}, \frac{10}{43}, \frac{176}{43}, \frac{87}{43}, \frac{10}{43}, \frac{50}{43}, \frac{30}{43}, 0 \right)' \text{ and } \overline{J}_B^0 = \{1, 4, 3, 8\}.$$

The optimal SFS of the augmented program is:

$$y^1 = \left(1, 0, \frac{16}{3}, \frac{7}{3}, 0, 0, \frac{10}{3}, 0 \right)' \text{ and } \overline{J}_B^1 = \{1, 4, 3, 7\} \text{ with } z\left(y^1\right) = 29.$$

Iteration 2: The resulting vector $x^1 = \left(1, 0, \frac{16}{3}, \frac{7}{3} \right)'$ being non-integer, we apply the rounding process on x^1, where the 2nd round according to the function z is feasible. We get $x^{1a} = (1, 0, 5, 2)'$, with $z(x^{1a}) = 27$. The β-optimality is:

$$\beta(x^{1a}) = \frac{22}{329} = 0.067.$$

If one considers that the value of the β-optimality is not yet satisfactory, then we set $x^{1*} = x^{1a}$, $y^{1*} = (x^{1a}, x^{s1a})$; with $x^{s1a} = (1, 1, 4, 11)$, and we add a new cut to the last previous program.

The reduced costs vector is:

$$E_N^1 = \left(E_2^1, E_5^1, E_6^1, E_8^1 \right)' = \{2, 0, 1, 1\}.$$

The partition of the set $J_N^+ = \left\{ j \in J_N^1 : E_j^1 > 0 \right\} = \{6, 2, 8\}$,

$J^+ = \left\{ j \in J_N^1 : E_j^1 = 0, x_j^1 - l_j \leq u_j - x_j^1 \right\} = \{5\}$, $J_{\bar{N}} = J^- = \emptyset$. We then calculate the values $Z_i(y^1)$, $i \in J_B^1$:

$$Z_B\left(y^1\right) = \left(0, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3} \right)'.$$

We're still in the case (1). The new cut is: $Z_2(y) \geq 0$, i.e.

$$2x_2 + x_5 + 2x_6 + x_8 - x_9 = 1.$$

We solve the resulting problem, beginning with the SFS $\{\bar{y}^1, \bar{J}_B^1\}$, where $\bar{J}_B^1 = J_B^1 \cup \{9\} = \{1, 4, 3, 7, 9\}$ and $\bar{y}^1 = (y^{1*} + \lambda(y^1 - y^{1*}), 0)'$

$$= \left(1, 0, \frac{679}{130}, \frac{289}{130}, \frac{43}{130}, \frac{43}{130}, \frac{231}{65}, \frac{473}{130}, 0\right), \lambda = \frac{87}{130}.$$

The optimal solution obtained by the proposed method after adding the cut $Z_2(y) \geq 0$ is:

$$y^2 = (1, 0, 5, 3, 1, 0, 2, 0, 0)', z\left(y^2\right) = 29.$$

This solution is integer, then the process stops. The vector $x^2 = (1, 0, 5, 3)'$ is an optimal solution of the problem (14), with $z(x^2) = 29$.

Remark 6: We have $\beta(x^2) = 0$, but we could stop at the good approximate solution $x^{a1} = (1, 0, 5, 2)$, with $z(x^{a1}) = 27$ and $\beta(x^a) = \frac{2}{29} \simeq 0.067$. In practice, for an approximate solution, we can stop the algorithm as soon as (x^a) no longer decreases.

6 Numerical experiments

In order to test the proposed algorithm and to make sure of its effectiveness, a comparative study is made with intlinprog of the MATLAB optimisation toolbox on randomly generated medium size test problems. The random numbers are uniformly distributed in [–10, 10]. The criterion of the comparison between the two methods is the CPU time in seconds and the number $\Delta F = z_r - z_e$, where z_r is the optimal value of z for the relaxed problem and z_e is the best value for ILP. The obtained results are presented in Table 1.

Table 1 Comparative results

n	*m*	*Intlinprog*		*Proposed algorithm*	
		CPU	*ΔF*	*CPU*	*ΔF*
5	2	0.3416	0.5000	0.1248	0.3000
	3	0.1201	0.4000	0.1362	0.5000
10	5	0.1591	3.0067	0.3744	2.0082
	7	0.1529	1.8822	0.0936	1.1021
15	5	0.1482	2.4389	0.2056	0.6754
	10	0.1232	5.7749	0.2176	3.6771
20	5	0.1326	0.6500	0.1864	0.8710
	10	0.1388	2.3241	0.2076	3.2091
	15	0.2262	5.3819	0.3572	4.0287
30	5	0.1248	0.5825	0.1948	1.5033
	10	0.1061	3.1314	0.1494	3.5022
	20	0.1529	6.6476	0.0919	7.1009

Table 1 Comparative results (continued)

n	*m*	*Intlinprog*		*Proposed algorithm*	
		CPU	ΔF	*CPU*	ΔF
50	10	0.1747	3.1133	0.7800	3.7500
	20	3.3244	6.5694	2.1216	5.2418
	30	57.2446	13.9576	2.1996	12.3343
80	10	0.1544	0.7626	0.1812	0.8503
	30	10.8280	8.5260	4.3056	7.9000
	50	12.0764	12.8872	6.3492	10.5000
100	10	1.0049	2.5012	1.0764	2.8012
	50	11.6524	13.7676	7.2849	10.2033
	70	14.6541	15.8753	8.7956	13.6532

The results show that the proposed algorithm is very competitive with intlinprog and presents good performances mainly for problems where m is large enough.

7 Conclusions

Unlike linear programming problems, integer programming problems are very difficult to solve. In fact, no efficient general algorithm is known for their resolution. The cutting planes algorithms give integer feasible solution only at the end of the resolution, whereas heuristics try to find integer feasible solution in a reasonable time, but we cannot know precisely the quality of these approximate solutions. Here, by combining a cutting plane method with a heuristic, we are able to calculate the β-optimality estimate that informs us about the quality of the approximate solution.

Acknowledgements

The authors are indebted to the Editor-in-Chief Dr. Xin-She Yang and to the anonymous referees whose comments and suggestions have improved the quality of the paper.

References

Achterberg, T., Berthold, T. and Hendel, G. (2012) 'Rounding and propagation heuristics for mixed integer programming', *Operations Research Proceedings*, pp.71–76.

Bertacco, L., Fischetti, M. and Lodi, A. (2007) 'A feasibility pump heuristic for general mixed-integer problems', *Discrete Optimization*, Vol. 4, No. 1, pp.63–76.

Bibi, M.O. and Bentobache, M. (2011) 'The adaptive method with hybrid direction for solving linear programming problems with bounded variables', *Proceedings of COSI'2011*, University of Guelma, Algeria, 24–27 April, pp.80–91.

Bibi, M.O. and Bentobache, M. (2015) 'A hybrid direction algorithm for solving linear programs', *International Journal of Computer Mathematics*, Vol. 92, No. 1, pp.201–216.

Conforti, M., Cornuéjols, G. and Zambelli, G. (2014) *Integer Programming*, Springer, Switzerland.

Cornuéjols, G., Margot, F. and Nannicini, G. (2013) 'On the safety of Gomory cut generators', *Math. Program. Comput.*, Vol. 5, No. 4, pp.345–395.

De Loera, J.A., Hemmecke, R. and Köppe, M. (2012) *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*, SIAM, Philadelphia.

Dey, S.S. and Molinaro, M. (2018) 'Theoretical challenges towards cutting-plane selection', *Mathematical Programming*, Vol. 170, No. 1, pp.237–266.

Djemai, S., Brahmi, B. and Bibi, M.O. (2016) 'A primal-dual method for SVM training', *Neurocomputing*, Vol. 211, pp.34–40.

Eisenbrand, F. and Weismantel, R. (2018) 'Proximity results and faster algorithms for integer programming using the Steinitz lemma', in *Proceedings of the Twenty-Ninth AnnualACMSIAM Symposium on Discrete Algorithms*, SIAM, pp.808–816.

Fellah, K. and Kechar, B. (2017) 'An integer linear programming approach for optimising energy consumption in mobile wireless sensor networks under realistic constraints', *International Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 8, No. 2, pp.162–182.

Fletcher, R. (1987) *Practical Methods of Optimization*, Vol. 2: Constrained Optimization, A Wiley-Interserscience Publication, John Wiley and Sons edition, New York.

Gabasov, R. (1993) *Adaptive Method of Linear Programming*, Preprints of the University of Karlsruhe, Institute of Statistics and Mathematics, Karlsruhe, Germany.

Gabasov, R. and Kirillova, F.M. (1980) *Methods of Linear Programming*, Vol. 3: Special Problems, University Press, Minsk.

Gabasov, R., Kirillova, F.M. and Prischepova, S.V. (1995) *Optimal Feedback Control*, Springer-Verlag, London.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York.

Gill, P.E., Murray, W., Saunders, M.A. and Wright, A.H. (1983) 'Computing forward difference intervals for numerical optimization', *SIAM Journal on Scientific and Statistical Computing*, Vol. 4, No. 2, pp.310–321.

Glover, F. and Sherali, H.D. (2005) 'Chvatal-Gomory-tier cuts for general integer programs', *Discrete Optimization*, Vol. 2, No. 1, pp.51–69.

Gomory, R.E. (1958) 'Outline of an algorithm for integer solutions to linear programs', *Bulletin of the American Mathematical Society*, Vol. 64, No. 5, pp.275–278.

Hu, T.C. (1969) *Integer Programming and Network Flows*, Addison-Wesley, London.

Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G. and Wolsey, L. (Eds.) (2010) *50 Years of Integer Programming 1958-2008*, Springer-Verlag, Berlin Heidelberg.

Karp, R.M. (1972) 'Reducibility among combinatorial problems', in Miller, R.E. and Thatcher, J.W. (Eds.): *Complexity of Computer Computations*, pp.85–103, Plenum Press, New York.

Korte, B. and Vygen, J. (2012) *Combinatorial Optimization*, Vol. 2, Springer, Heidelberg.

Llewellyn, D.C. and Ryan, J. (1993) 'A primal dual integer programming algorithm', *Discrete Applied Mathematics*, Vol. 45, pp.261–275.

Lodi, A. (2010) 'Mixed integer programming computation', in *50 Years of Integer Programming 1958-2008*, Springer, Berlin, Heidelberg.

Marchand, H., Martin, A., Weismantel, R. and Wolsey, L. (2002) 'Cutting planes in integer and mixed integer programming', *Discrete Applied Mathematics*, Vol. 123, Nos. 1–3, pp.397–446.

Mei, L.S., Adi, S., Mohd Rizal, B.S. and Effendi, M. (2018) 'Comparison between linear programming and integer linear programming: a review', *International Journal of Mathematics in Operational Research*, Vol. 13, No. 1, pp.91–106.

Nemhauser, G.L. and Wolsey, L.A. (1988) *Integer and Combinatorial Optimization*, Wiley-Interscience, New Jersey.

Owen, J. and Mehrotra, S. (2001) 'Experimental results using general disjunctions in branch-and-bound for general-integer linear programs', *Computational Optimization and Applications*, Vol. 20, No. 2, pp.159–170.

Parker, R.G. and Rardin, R.L. (1988) *Discrete Optimization*, Academic Press, San Diego.

Roos, C., Terlaky, T. and Vial, J. Ph. (1997) *Theory and Algorithms for Linear Optimization, An Interior-Point Approach*, John Wiley and Sons, Chichester, UK.

Salkin, H.M. and Morito, S. (1975) *Integer Programming by Group Theory, Some Computational Results*, Tech. report, Defense Technical Information Center OAI-PMH Repository, USA.

Schrijver, A. (1986) *Theory of Linear and Integer Programming*, John Wiley, Chichester.

Schrijver, A. (2003) *Combinatorial Optimization: Polyhedra and Efficiency*, Vol. A., Paths, Flows, Matchings, Springer, Berlin.

Taha, H.A. (1975) *Integer Programming Theory, Applications and Computations*, Academic Press, New York.

Vial, J.P. (1994) 'Computational experience with a primal-dual interior-point method for smooth convex programming', *Optimization Methods and Software*, Vol. 3, No. 4, pp. 285–310.

Wright, S.J. (1997) *Primal-Dual Interior Point Methods*, SIAM, Philadelphia, USA.

Zionts, S. (1974) *Linear and Integer Programming*, Prentice-Hall, New York.