

Paweł Kaden, Dariusz Horla\*

## PERFORMANCE EVALUATION OF MINIMIZATION METHODS FOR BINARY AND INTEGER LP PROBLEMS

**Keywords:** integer LP problems, optimization, computational complexity

### 1. INTRODUCTION

There is a vast number of decision-making problems that result in integer optimization problems which is a great challenge because of its well-known exponential worst-case computational complexity. In recent years, the main development in the field took place in binary linear programming (LP).

The paper aims to evaluate the performance of selected binary and integer programming algorithms dedicated to solving LPs [2,3,4,9]. The worst-case assumption concerning the computational complexity should not be applied here, because of many improvements originating from observations of the characteristics of given optimization problems, resulting in speeding up the algorithms.

The paper is composed of two parts, concerning binary, and, subsequently, integer LPs, followed by the Section presenting the results. The latter can be used to select the most appropriate algorithm for the encountered optimization problems, and does not result from pure mathematical inference concerning the structure of the problem, but is based on generating random problems and trying to solve them.

### 2. LINEAR PROGRAMMING IN INTEGER SETS

Linear programs have an optimal point in the convex polyhedron where decision variables from the vector  $\underline{x}$  can have any values, as far as the solution is in the feasible set  $\Phi$ . The only case on the contrary to the former is the degenerate solution, where the set  $\Phi$  contains a single point only.

In many practical problems, however, it is desired that the optimal solution should be contained in a discrete-set, i.e. that  $\underline{x}$  should have a limited number of possible values.

The discrete set usually includes specified elements, i.e. natural numbers. The linear program of the following type:

---

\*Poznań University of Technology, Institute of Control and Information Engineering, Department of Control and Robotics, Piotrowo 3a Str., 60-965 Poznań, e-mail: Dariusz.Horla@put.poznan.pl

$$\begin{aligned}
& \min_{\underline{x}} \quad \underline{c}^T \underline{x} \\
& \text{s.t.} \quad \underline{A}\underline{x} = \underline{b}, \\
& \quad \underline{x} \geq \underline{0}, \\
& \quad x_i \in \mathcal{Z}, i \in I \subseteq \mathcal{N} = \{0, 1, 2, 3, \dots\},
\end{aligned}$$

where some ( $I \subset \mathcal{N}$ ) or all ( $I = \mathcal{N}$ ) components  $x_i$  of the vector of decision variables  $\underline{x}$  are natural numbers, is called a discrete linear program.

A general division of linear programs in discrete sets is as follows:

- binary programming problems (0-1),
- integer programming problems.

### 3. BINARY PROGRAMS

#### 3.1. INTRODUCTION

Apart from standard linear programming constraints, the decision variables may be required to take on values 0 or 1 only, in binary programming problems, what may be treated as logical true or false, and the mathematical problem is formulated as follows:

$$\begin{aligned}
& \min_{\underline{x}} \quad \underline{c}^T \underline{x} \\
& \text{s.t.} \quad \underline{A}\underline{x} = \underline{b}, \\
& \quad x_i = 0 \text{ or } x_i = 1 \quad (i = 1, 2, \dots, n).
\end{aligned}$$

As it can be seen,  $n$  variables of the linear program may have two values only, thus the number of all possible solutions is limited and not greater than  $2^n$ .

By systematic enumeration of the possible solutions, one can verify if the constraints are violated and compute the value of the aim function  $f(\underline{x}) = \underline{c}^T \underline{x}$ , and find the solution in a finite time by choosing solutions that abide constraints and for which the aim function decreases. Such a task is not computationally demanding, because one only needs addition and multiplication operations.

#### 3.2. LEXICOGRAPHIC ORDER METHOD [7]

In the algorithm that performs systematic enumeration of the feasible points from the set  $\Phi$ , consecutive solutions are verified. There is a need, however, to avoid losing any point. As an example, there is a simple lexicographic order search presented in the Figure 1 for  $\underline{x} = [x_1, x_2, x_3]^T$ .

The computational burden of such an algorithm can be reduced by incorporating new, artificial constraints, into the problem, related to the current value of the aim function, what is the basis of the presented algorithms.

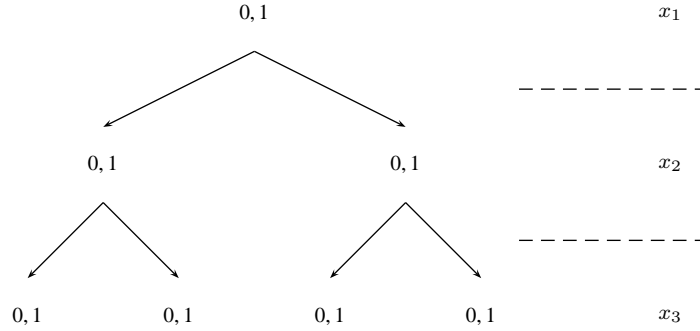


Fig. 1. A lexicographic order example

### 3.3. BALAS METHOD [1,7,10]

The algorithm is expected to perform exhaustive search by enumerating all possible solutions to the problem, and each candidate solution must abide all  $m$  constraints ( $\mathbf{A} \in \mathcal{R}^{m \times n}$ ).

In order to find the optimal solution one needs to perform from  $m2^n$  to  $(m+1)2^n$  operations, assuming that all points satisfy the constraints and the aim function is to be computed at all times. The optimization problem with large  $n$  requires many calculations to find the optimal solution, thus it is often appealing to introduce, as it has been initially written, additional constraints to their matrix representation  $\mathbf{A}$  in the form of static or dynamic filter.

Static filter approach allows one to introduce a single additional constraint once the first feasible solution  $\hat{\underline{x}}$  is found in the lexicographic order, in the form  $\underline{c}^T \underline{x} \leq \underline{c}^T \hat{\underline{x}}$ , allowing to reject all feasible points with no improvement to the initial point.

Dynamic filter version of the algorithm updates the filter every time an improved solution is found, substituting  $\hat{\underline{x}}$  with it. In the both cases, the additional filtering constraint is placed at the top of the constraints (in the first row of  $\mathbf{A}$ ) to improve the performance of the algorithm.

As an example, let us define the following binary programming problem:

$$\begin{aligned}
 \min_{\underline{x}} \quad & 9x_1 + 2x_2 + 8x_3 + 8x_4 \\
 \text{s.t.} \quad & 4x_1 - 9x_2 - 2x_3 - 6x_4 \leq -6, \quad (1) \\
 & 4x_1 - 10x_2 + 10x_3 + 2x_4 \geq -9, \quad (2) \\
 & -4x_1 - 10x_2 + 6x_3 - 7x_4 \leq 2, \quad (3) \\
 & x_i = 0 \text{ or } x_i = 1 \ (i = 1, 2, \dots, n).
 \end{aligned}$$

In the table below the basic Balas algorithm using lexicographic method solution is presented

$x_1$	$x_2$	$x_3$	$x_4$	(1)	(2)	(3)	$f(\underline{x})$
0	0	0	0	—			
0	0	0	1	+	+	+	8
0	0	1	0	—			
0	0	1	1	+	+	+	16
0	1	0	0	+	—		
0	1	0	1	+	+	+	10
0	1	1	0	+	+	+	10
0	1	1	1	+	+	+	18
1	0	0	0	—			
1	0	0	1	—			
1	0	1	0	—			
1	0	1	1	—			
1	1	0	0	—			
1	1	0	1	+	+	+	19
1	1	1	0	+	+	+	19
1	1	1	1	+	+	+	27

with 41 calculations performed out of 64 in maximum. Using static filter approach an additional constraint (0) may be defined as below

$x_1$	$x_2$	$x_3$	$x_4$	(0)	(1)	(2)	(3)	$f(\underline{x})$
0	0	0	0		—			
0	0	0	1		+	+	+	8
0	0	1	0	+	—			
0	0	1	1	—				
0	1	0	0	+	+	—		
0	1	0	1	—				
0	1	1	0	—				

⋮

what allows one to reject solutions with no improvement before evaluating their feasibility, where the additional filtering constraint becomes

$$9x_1 + 2x_2 + 8x_3 + 8x_4 \leq 8 \quad (0).$$

Even in comparison with dynamic filter method, additional improvement can be observed if order in variables' values are changed, i.e. having sorted the vector of decision variables for lexicographic order with respect to increasing absolute values of elements of  $\underline{c}$ .

### 3.4. ALGORITHM FOR BINARY PROGRAMMING WITH PARTIAL ENUMERATION

This algorithm finds the optimal solution among subsets of  $2^n$  element set of possible candidate solutions, generating consecutive partial solutions by prescribing 0/1 values to decision

variables and checking for possible violation of the constraints. For the partial solution, that is feasible, some variables have fixed binary values and remaining possible solutions result by choosing free variables' values.

For the problem of the form

$$\begin{aligned} \min_{\underline{x}} \quad & \underline{c}^T \underline{x} \\ \text{s.t.} \quad & \underline{A}\underline{x} = \underline{b}, \\ & x_i = 0 \text{ or } x_i = 1 \quad (i = 1, 2, \dots, n) \end{aligned}$$

$j$ th constraint can be re-written as ( $j = 1, 2, \dots, m$ )

$$\sum_{i=1}^n a_{j,i} x_i \leq b_j.$$

Let the index  $i$  belong to the set of indexes that can be decomposed into the set  $\Sigma$  of fixed decision variables and the set  $\overline{\Sigma}$  of variables with values to-be-chosen. In order to find a partial solution the  $j$ th constraint must be transformed into

$$\sum_{i \in \Sigma} a_{j,i} x_i + \sum_{i \in \overline{\Sigma}} a_{j,i} x_i \leq b_j,$$

with

$$\sum_{i \in \Sigma} a_{j,i} x_i \leq b_j - \sum_{i \in \overline{\Sigma}} a_{j,i} x_i. \quad (1)$$

The condition (1) is used to check if the candidate solution is feasible, and its negation

$$\sum_{i \in \Sigma} a_{j,i} x_i > b_j - \sum_{i \in \overline{\Sigma}} a_{j,i} x_i$$

is used to eliminate the candidate solution from further consideration.

By choosing an improved partial solution, the solutions that do not assure an improvement in the aim function are eliminated, i.e. for the  $k$ th improved partial solution it must hold that

$$\left( \sum_{i \in \Sigma} c_i x_i + \sum_{i \in \overline{\Sigma}} c_i x_i \right)^{(k)} < \left( \sum_{i \in \Sigma} c_i x_i + \sum_{i \in \overline{\Sigma}} c_i x_i \right)^{(k-1)}. \quad (2)$$

The condition (2) allows one to improve the solution and reject all worse solutions from further considerations.

In the algorithm that is given below, vector  $\underline{u}$  defines the state of the variables for the partial solution. There are three possible states, namely: fixed, with already considered complement, and with variable complement. If the variable is fixed but its complement has not been already considered then the index of this variable is included into  $\underline{u}$ . If the complement has been already considered, then the index with negative sign is included into  $\underline{u}$ . If the variable has no fixed value the zero value is included into  $\underline{u}$ . The order in which the variables are fixed is important in the algorithm.

The algorithm for the problem in the form

$$\begin{aligned} \min_{\underline{x}} \quad & \underline{c}^T \underline{x} \\ \text{s.t.} \quad & \underline{A} \underline{x} \leq \underline{b}, \\ & x_i \in \{0, 1\} \quad (i = 0, 1, \dots, n). \end{aligned}$$

can be described as follows:

1) Initialization step

Verify if  $\underline{b} \geq \underline{0}$ . If so, the optimal solution  $\underline{x} = \underline{0}$  has been found. In the opposite case, set the optimal value of  $f_{\min}$  as sufficiently large number and proceed to step 2.

2) Compute

$$y_i = b_i - \sum_{j \in J} a_{ij} x_j,$$

where  $J$  includes the indexes of non-fixed variables (initially  $J = \emptyset$ ).

Compute

$$y_{\min} = \min_{i=1, \dots, m} y_i,$$

and check the following conditions:  $y_{\min} \geq 0$  and  $\hat{f} < f_{\min}$ , where  $\hat{f}$  is the aim function value for the solution under consideration  $\underline{x}$ . If the latter conditions hold, then  $\underline{x}_{\min} = \underline{x}$ ,  $f_{\min} = \hat{f}$ , and proceed to step 6.

3) Create the set  $T$  comprising free (non-fixed) variables, such that:

$$T = \left\{ j : \hat{f} + c_j < f_{\min}, a_{ij} < 0 \text{ for } i \text{ satisfying } y_i < 0 \right\}.$$

If it holds that  $T = \emptyset$ , proceed to step 6, in the opposite case proceed to step 4.

4) Infeasibility test – if there is an index  $k$  that it holds that

$$y_k - \sum_{j \in T} \min(0, a_{kj}) < 0 \text{ with } y_k < 0,$$

proceed do step 6, in the opposite case, proceed to step 5.

5) Choosing a free variable to fix – for free variables create the sets

$$M_{j \in J} = \{i : y_i - a_{ij} < 0\}.$$

If all the sets are empty, proceed to step 6. In the opposite case compute

$$v_{j \in J} = \sum_{i \in M_j} (y_i - a_{ij}).$$

If it holds that  $M_j = \emptyset$ , assume  $v_j = 0$ . Fix a free variable value related to maximum  $v_j$  (assume that  $x_j = 1$ ), and proceed to step 2.

- 6) For the fixed variable  $x_j$  that refers to positive index in vector  $\underline{u}$ , choose the zero value (consider its complement). Change the sign of the last positive index in vector  $\underline{u}$  and fix all its right-hand side variables (with respect to the zeroed variable). If all complements have already been considered, i.e.  $\underline{u}$  does not contain positive values, stop the algorithm. The optimal solution is defined by the pair  $\underline{x}_{\min}, f_{\min}$ . If  $f_{\min}$  is at its initial large value, the problem is infeasible.

As an example, let us define the following binary programming problem:

$$\begin{aligned} \min_{\underline{x}} \quad & 3x_1 + 2x_2 + x_3 + 8x_4 \\ \text{s.t.} \quad & -x_1 + x_2 + 6x_3 + x_4 \leq 5, \\ & -x_1 - 2x_2 + 3x_3 - x_4 \leq -3, \\ & 2x_1 + 2x_2 - x_3 - 8x_4 \leq -6, \\ & x_i = 0 \text{ or } x_i = 1 \ (i = 1, 2, \dots, n). \end{aligned}$$

#### Iteration no. 1

- 2)  $y_i = [5, -3, -6]^T, y_{\min} = -6$ . Since it holds that  $y_{\min} < 0$  or  $f \geq f_{\min}$ , we proceed to step 3.
- 3) A set  $T$  is created for free variables, namely,  $x_1, x_2, x_3, x_4, T = \{1, 2, 3, 4\}$ .
- 4) The current solution  $\underline{x} = [0, 0, 0, 0]^T, f(\underline{x}) = 0$  is infeasible.
- 5) Minimum violation of the constraints test:  $M_1 = \{2, 3\}, M_2 = \{2, 3\}, M_3 = \{1, 2, 3\}, M_4 = \{2\}, v_1 = -10, v_2 = -9, v_3 = -12, v_4 = -2$ , with the largest  $v_j$  for  $j = 4$ . Fix the value of  $x_4 = 1$ .

#### Iteration no. 2

- 2)  $y_i = [4, -2, 2]^T, y_{\min} = -2$ . Since it holds that  $y_{\min} < 0$  or  $f \geq f_{\min}$ , we proceed to step 3.
- 3) A set  $T$  is created for free variables, namely,  $x_1, x_2, x_3, T = \{1, 2\}$ .
- 4) The current solution  $\underline{x} = [0, 0, 0, 1]^T, f(\underline{x}) = 1$  is infeasible.
- 5) Minimum violation of the constraints test:  $M_1 = \{1\}, M_2 = \emptyset, M_3 = \{1, 2\}, v_1 = -1, v_2 = 0, v_3 = -7$ , with the largest  $v_j$  for  $j = 2$ . Fix the value of  $x_2 = 1$ .

#### Iteration no. 3

- 2)  $y_i = [3, 0, 0]^T, y_{\min} = 0$ . Since it holds that  $y_{\min} \geq 0$  and  $f < f_{\min}$  the improved solution is found, namely  $\underline{x} = [0, 1, 0, 1]^T, f(\underline{x}) = 3$ .
- 6) Consider the complementary value of  $x_2$ , namely  $x_2 = 0$ .

#### Iteration no. 4

- 2)  $y_i = [4, -2, 2]^T, y_{\min} = -2$ . Since it holds that  $y_{\min} < 0$  or  $f \geq f_{\min}$ , we proceed to step 3.

- 3) A set  $T$  is created for free variables, namely,  $x_1, x_3, T = \emptyset$ .
- 6) Consider the complementary value of  $x_4$ , namely  $x_4 = 0$ .

#### Iteration no. 5

- 2)  $y_i = [5, -3, -6]^T$ ,  $y_{\min} = -6$ . Since it holds that  $y_{\min} < 0$  or  $f \geq f_{\min}$ , we proceed to step 3.
- 3) A set  $T$  is created for free variables, namely,  $x_1, x_3, T = \{3\}$ .
- 4) The current solution  $\underline{x} = [0, 0, 0, 0]^T$ ,  $f(\underline{x}) = 0$  is feasible.
- 6) All the remaining complements have been considered, and the algorithm may stop. The optimal solution is

$$\underline{x}^* = [0, 1, 0, 1]^T, \quad f(\underline{x}^*) = 3.$$

## 4. INTEGER PROGRAMMING PROBLEMS

### 4.1. FORMULATION OF THE PROBLEM

In integer LPs, in comparison with standard LP problems, there is an additional requirement that decision variables must have integer values, what is of practical value. The feasible set contains in the specific case a set of points. The problem however, becomes answering the question if the point is feasible or infeasible.

If the feasible solution set is compact and contains a limited number of points, one can perform exhaustive search to systematically reject worse solutions, what can be time-consuming. If we, however, reject the integer requirement, the solution can be found via, e.g. a simplex algorithm, and rounded towards the nearest integer neighbour. This may lead to infeasibility of the solution, as depicted in Figure 2.

Let the following problem be given

$$\begin{aligned} \min_{\underline{x}} \quad & -2x_1 - 5x_2 \\ \text{s.t.} \quad & 2x_1 - x_2 \geq 6, \\ & x_1 - 6x_2 \geq -24, \\ & \underline{x} \geq \underline{0}, \\ & x_1, x_2 \in \mathcal{Z}. \end{aligned}$$

The dashed line in Figure 2 denotes the feasible solution set omitting the last constraint. The large black dots are possible feasible solutions with  $x_1, x_2 \in \mathcal{Z}$ .

The solution of the problem in real numbers is  $\underline{x}_{\text{real}}^* = [\frac{60}{11}, \frac{54}{11}]^T$ ,  $f(\underline{x}_{\text{real}}^*) = -\frac{390}{11}$ , what violates  $x_1, x_2 \in \mathcal{Z}$ .

Having rounded this solution to the nearest neighbour one obtains integer solution  $\hat{\underline{x}} = [5, 5]^T$ ,  $f(\hat{\underline{x}}) = -35$ , however this point is infeasible.

Based on the aim function contour lines, the following point  $\underline{x} = [5, 4]^T$  has the optimal value, i.e.  $\underline{x}^* = [5, 4]^T$  with  $f(\underline{x}^*) = -30$ .



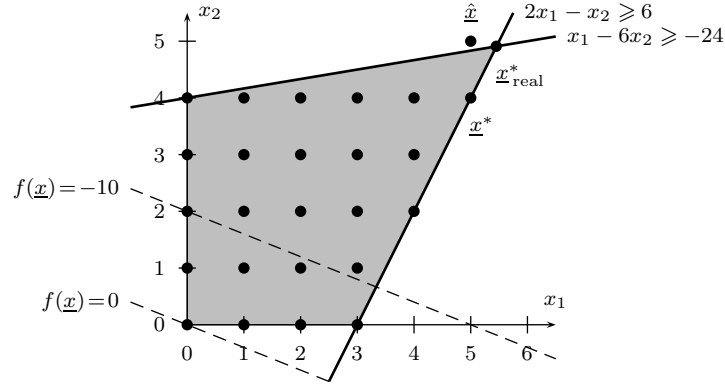


Fig. 2. Feasibility set

#### 4.2. GOMORY ALGORITHM OF THE BASIC CUTS

The formulated problem can be solved with a standard simplex method at first. If the solution does not violate the integer constraints, it becomes the optimal solution. If not, one can introduce additional constraints (Gomory cuts), to reduce the initial set  $\Phi$ , to force the optimal solution of the new problem to respect the integer constraints. Additional constraints are cutting hyperplanes, reducing the size of the initial feasible solutions set.

A cutting hyperplane is constructed so that

- a new feasible set is convex,
- cut-off parts of the feasible set do not contain integer solutions.

Introducing consecutive constraints forces one to solve consecutive LP problems having omitted integer requirement.

Let the following LP be given

$$\begin{aligned} \min_{\underline{x}} \quad & \underline{c}^T \underline{x} \\ \text{s.t.} \quad & \underline{A} \underline{x} \leq \underline{b}, \\ & \underline{x} \geq \underline{0}, \end{aligned}$$

with  $\underline{b} \geq \underline{0}$ , where using basic and non-basic variables one can transform the constraints into

$$\underline{A}_B \underline{x}_B + \underline{A}_N \underline{x}_N \leq \underline{b}.$$

By changing the order of the variables ( $B$  denotes basic and  $N$  non-basic variables):

$$\begin{aligned} \underline{x}_B &= [x_{n-m+1}, x_{n-m+2}, \dots, x_n]^T, \\ \underline{x}_N &= [x_1, x_2, \dots, x_{n-m}]^T \end{aligned}$$

the  $j$ th constraint can be re-written to the form

$$\sum_{i=1}^n a_{j,i} x_i \leq b_j,$$

with  $(j = 1, 2, \dots, m)$

$$\sum_{i=1}^{n-m} a_{j,i} x_i + \sum_{i=n-m+1}^n a_{j,i} x_i = \sum_{i=1}^{n-m} a_{j,i} x_i + \sum_{i=n-m+1}^n a_{j,i} x_{s,i-n+m} = b_j ,$$

where  $x_{s,1}, x_{s,2}, \dots, x_{s,m}$  are slack variables.

By solving the following set of equations

$$\sum_{i=1}^{n-m} a_{j,i} x_i + \sum_{i=1}^m a_{j,i} x_{s,i} = b_j ,$$

one can define the sought  $n-m$  variables as a function of  $m$  slack variables ( $k = 1, 2, \dots, n-m$ )

$$x_k = \bar{b}_j - \sum_{i=1}^m \bar{a}_{j,i} x_{s,i} . \quad (3)$$

If it holds that  $\bar{b}_j, \bar{a}_{j,i} \notin \mathcal{Z}$ , one can write them as a sum of integer and real part

$$\bar{b}_j = \bar{b}_j^{\mathcal{Z}} + \bar{b}_j^{\mathcal{R}} , \quad (4)$$

$$\bar{a}_{j,i} = \bar{a}_{j,i}^{\mathcal{Z}} + \bar{a}_{j,i}^{\mathcal{R}} , \quad (5)$$

$$\bar{b}_j^{\mathcal{Z}}, \bar{a}_{j,i}^{\mathcal{Z}} \in \mathcal{Z} , \quad (6)$$

$$0 \leq \bar{b}_j^{\mathcal{R}}, \bar{a}_{j,i}^{\mathcal{R}} < 1 . \quad (7)$$

Finally one gets:

$$x_k = \bar{b}_j^{\mathcal{Z}} + \bar{b}_j^{\mathcal{R}} - \sum_{i=1}^m (\bar{a}_{j,i}^{\mathcal{Z}} + \bar{a}_{j,i}^{\mathcal{R}}) x_{s,i} ,$$

$$\bar{b}_j^{\mathcal{R}} - \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{R}} x_{s,i} = x_k - \bar{b}_j^{\mathcal{Z}} + \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{Z}} x_{s,i} .$$

Since the variables  $x_k$  i  $x_{s,i}$  must have integer values, it holds that:

$$x_k - \bar{b}_j^{\mathcal{Z}} + \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{Z}} x_{s,i} \in \mathcal{Z} ,$$

$$\bar{b}_j^{\mathcal{R}} - \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{R}} x_{s,i} \in \mathcal{Z} .$$

Since  $\bar{b}_j^{\mathcal{R}} - \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{R}} x_{s,i} \in \mathcal{Z}$  is either zero or negative integer,

$$\bar{b}_j^{\mathcal{R}} - \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{R}} x_{s,i} \leq 0 . \quad (8)$$

Introducing new slack variable  $x_{s,m+1} \geq 0$ ,  $x_{s,m+1} \in \mathcal{Z}$  to (8) one obtains

$$x_{s,m+1} - \sum_{i=1}^m \bar{a}_{j,i}^{\mathcal{R}} x_{s,i} = -\bar{b}_j^{\mathcal{R}}, \quad (9)$$

i.e. the cutting hyperplane.

As an example let us consider the following problem

$$\begin{aligned} \min_{\underline{x}} \quad & -3x_1 - 7x_2 \\ \text{s.t.} \quad & 3x_1 + 8x_2 \leq 24, \\ & 2x_1 + 3x_2 \leq 12, \\ & \underline{x} \geq \underline{0}, \\ & x_1, x_2 \in \mathcal{Z} \end{aligned}$$

that can be re-written into the form

$$\begin{aligned} \min_{\underline{x}} \quad & -3x_1 - 7x_2 \\ \text{s.t.} \quad & 3x_1 + 8x_2 + x_3 = 24, \\ & 2x_1 + 3x_2 + x_4 = 12, \\ & \underline{x} \geq \underline{0}, \\ & x_1, x_2, x_3, x_4 \in \mathcal{Z}, \end{aligned}$$

### Iteration no. 1

The solution of the LP in the real numbers:

$$\underline{x}^{(1)} = [3.429, 1.714]^T$$

violates the integer requirement.

The following Gomory cut

$$2x_1 + 5x_2 \leq 15$$

is defined.

### Iteration no. 2

The new problem becomes

$$\begin{aligned} \min_{\underline{x}} \quad & -3x_1 - 7x_2 \\ \text{s.t.} \quad & 3x_1 + 8x_2 + x_3 = 24, \\ & 2x_1 + 3x_2 + x_4 = 12, \\ & 2x_1 + 5x_2 + x_5 = 15, \\ & \underline{x} \geq \underline{0}, \\ & x_1, x_2, x_3, x_4, x_5 \in \mathcal{Z}, \end{aligned}$$

and its solution in real numbers:

$$\underline{x}^{(2)} = [3.75, 1.50]^T$$

violates the integer requirement.

The following Gomory cut

$$x_1 + 2x_2 \leq 6$$

is defined.

### Iteration no. 3

The new problem becomes

$$\begin{aligned} \min_{\underline{x}} \quad & -3x_1 - 7x_2 \\ \text{s.t.} \quad & 3x_1 + 8x_2 + x_3 = 24, \\ & 2x_1 + 3x_2 + x_4 = 12, \\ & 2x_1 + 5x_2 + x_5 = 15, \\ & x_1 + 2x_2 + x_6 = 15, \\ & \underline{x} \geq \underline{0}, \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \mathcal{Z}, \end{aligned}$$

and its solution in real numbers:

$$\underline{x}^{(3)} = [0, 3]^T$$

satisfies the integer requirement.

The optimal solution to the problem becomes  $\underline{x}^* = [0, 3]^T$ ,  $f(\underline{x}^*) = -21$ .

Tab. 1. Gomory cuts for the given LP problem

iteration	$x_i$	Gomory cut	$\underline{x}^{(k+1)}$	$f(\underline{x}^{(k+1)})$
I	$x_1 = 3.429 - 0.4286x_3$	$2x_1 + 5x_2 \leq 15$	$[3.429, 1.714]^T$	-22.29
II	$x_1 = 3.75 + 1.25x_4$	$x_1 + 2x_2 \leq 6$	$[3.75, 1.5]^T$	-21.75

### 4.3. BRANCH AND BOUND ALGORITHM FOR INTEGER LPS

Another algorithm that can be used to solve integer LPs is based on branch and bound approach and presents the given LP as two separate problems [6, p. 59].

Let P0 be the following problem

$$\begin{aligned} \min_{\underline{x}} \quad & f(\underline{x}) \\ \text{s.t.} \quad & \mathbf{A}\underline{x} = \underline{b}, \\ & \underline{x} \geq \underline{0}. \end{aligned}$$

If the solution to P0 satisfies  $x_i \in \mathcal{Z}$ ,  $i \in I \subseteq \mathcal{N} = \{0, 1, 2, 3, \dots\}$ , then integer solution is found. If P0 is either an infeasible or an unbounded problem, introducing integer requirement will not improve the situation.

If the solution  $\underline{x}_{P0}^*$  to the problem violates integer requirement (or any other solution to the transformed problem), the following approach is adopted:

- if the solution to the subproblem is integer, it becomes candidate solution to the original problem and becomes a leaf in the tree of possible solutions (and is not taken for possible branching),
- in the opposite case, the chosen variable  $x_i \notin \mathcal{Z}$ ,  $x_i \in [a, a+1]$  ( $i \in I$ ,  $a \in \mathcal{Z}$ ) located between two integer numbers, namely,  $a$  and  $a+1$ , is used to branch the problem into two subproblems introducing new bounds, i.e.  $x_i \leq a$  and  $x_i \geq a+1$  to one of the pair of the new problems, as in Figure 3.

The above procedure is repeatedly used to build the complete branching tree.

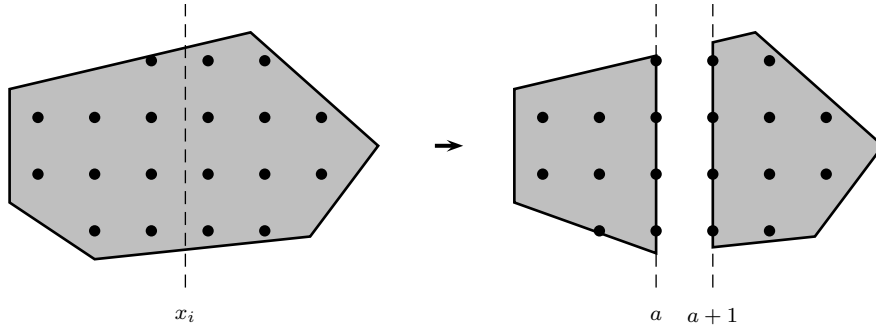


Fig. 3. Branching procedure

Let the following problem be given

$$\begin{aligned}
 \min_{\underline{x}} \quad & -3x_1 - 7x_2 \\
 \text{s.t.} \quad & 3x_1 + 8x_2 \leq 24, \\
 & 2x_1 + 3x_2 \leq 12, \\
 & \underline{x} \geq \underline{0}, \\
 & x_1, x_2 \in \mathcal{Z}.
 \end{aligned}$$

Initially,

$$\begin{aligned}
 \text{(P0)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 \text{s.t.} \quad & 3x_1 + 8x_2 \leq 24, \\
 & 2x_1 + 3x_2 \leq 12, \\
 & \underline{x} \geq \underline{0},
 \end{aligned}$$

has the solution  $\underline{x}_{P0}^* = [\frac{24}{7}, \frac{12}{7}]^T$  with  $f(\underline{x}_{P0}^*) = -\frac{156}{7} \approx -22.2857$ . It needs to be divided into two subproblems:

$$\begin{aligned}
 \text{(P1)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 3, \\
 & \quad \quad \underline{x} \geq \underline{0},
 \end{aligned}$$

$$\begin{aligned}
 \text{(P2)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \geq 4, \\
 & \quad \quad \underline{x} \geq \underline{0}.
 \end{aligned}$$

The solution to P1 becomes  $\underline{x}_{P1}^* = [3, \frac{15}{8}]^T$ ,  $f(\underline{x}_{P1}^*) = -\frac{177}{8} \approx -22.1250$ . Since  $x_{2,P1} \notin \mathcal{Z}$ , the solution is divided again, leading to the following problems:

$$\begin{aligned}
 \text{(P3)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 3, \\
 & \quad \quad x_2 \leq 1, \\
 & \quad \quad \underline{x} \geq \underline{0},
 \end{aligned}$$

$$\begin{aligned}
 \text{(P4)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 3, \\
 & \quad \quad x_2 \geq 2, \\
 & \quad \quad \underline{x} \geq \underline{0}.
 \end{aligned}$$

The solution to P3  $\underline{x}_{P3}^* = [3, 1]^T$ ,  $f(\underline{x}_{P3}^*) = -16$  is integer and becomes the leaf. The solution to P4:  $\underline{x}_{P4}^* = [\frac{8}{3}, 2]^T$ ,  $f(\underline{x}_{P4}^*) = -22$ ,  $x_{1,P4} \notin \mathcal{Z}$ , enables one to divide the

feasibility set again, leading to:

$$\begin{aligned}
 \text{(P5)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 2, \\
 & \quad \quad x_2 \geq 2, \\
 & \quad \quad \underline{x} \geq \underline{0},
 \end{aligned}$$

$$\begin{aligned}
 \text{(P6)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 3, \\
 & \quad \quad x_1 \geq 3 \quad (\text{put together as } x_1 = 3), \\
 & \quad \quad x_2 \geq 2, \\
 & \quad \quad \underline{x} \geq \underline{0}.
 \end{aligned}$$

The solution to P5 is  $\underline{x}_{P5}^* = [2, \frac{9}{4}]^T$ ,  $f(\underline{x}_{P5}^*) = -\frac{87}{4} = 21.75$ ,  $x_{2,P5} \notin \mathcal{Z}$ , and branching again:

$$\begin{aligned}
 \text{(P7)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 2, \\
 & \quad \quad x_2 \leq 2, \\
 & \quad \quad x_2 \geq 2 \quad (\text{put together as } x_2 = 2), \\
 & \quad \quad \underline{x} \geq \underline{0},
 \end{aligned}$$

$$\begin{aligned}
 \text{(P8)} \quad & \min_{\underline{x}} \quad -3x_1 - 7x_2 \\
 & \text{s.t.} \quad 3x_1 + 8x_2 \leq 24, \\
 & \quad \quad 2x_1 + 3x_2 \leq 12, \\
 & \quad \quad x_1 \leq 2, \\
 & \quad \quad x_2 \geq 3, \\
 & \quad \quad \underline{x} \geq \underline{0}.
 \end{aligned}$$

The solution to P7  $\underline{x}_{P7}^* = [2, 2]^T$ ,  $f(\underline{x}_{P7}^*) = -20$  is integer, as  $\underline{x}_{P8}^* = [0, 3]^T$ ,  $f(\underline{x}_{P8}^*) = -21$ . The solution to P6 is  $\underline{x}_{P6}^* = [3, 1]^T$ ,  $f(\underline{x}_{P6}^*) = -16$ , and the solution to P2 becomes  $\underline{x}_{P2}^* = [4, \frac{4}{3}]^T$ ,  $f(\underline{x}_{P2}^*) = -\frac{64}{3} \approx -21.3333$ .

Since both solutions to P1 and P2 violate integer requirement, the initial feasible set is divided into two parts, and since it holds that  $f(\underline{x}_{P2}^*) > f(\underline{x}_{P1}^*)$ , the solution lies in the

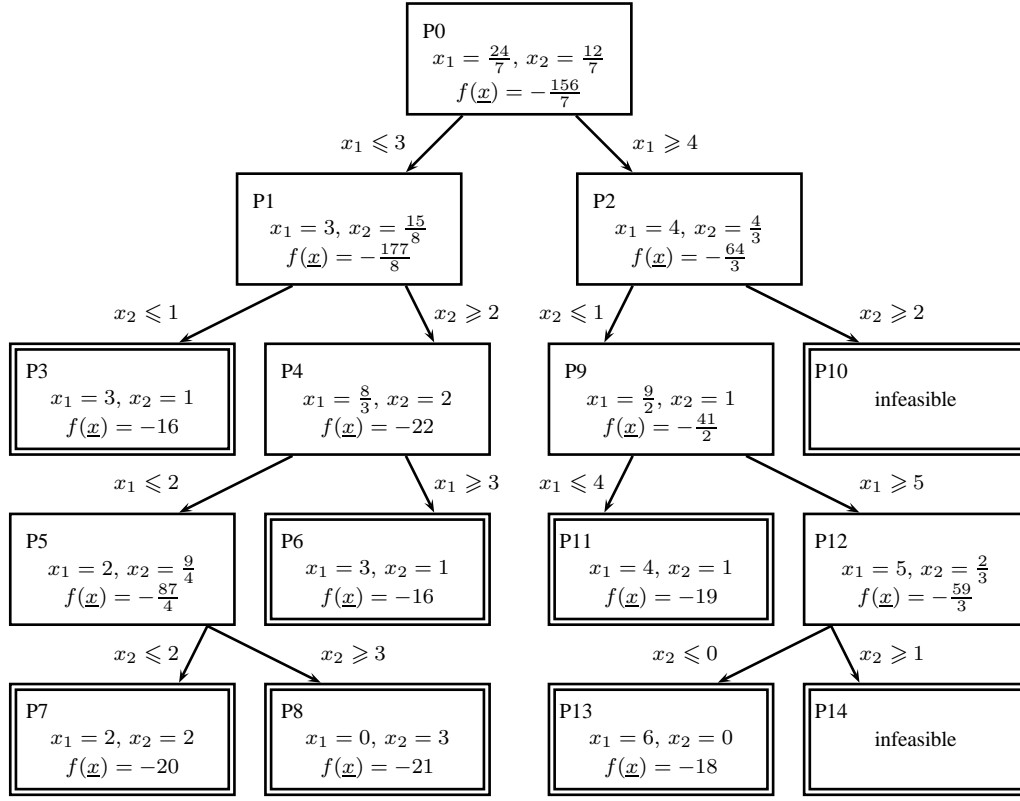


Fig. 4. Branching procedure

branch of P1.

In the Figure 5 successive feasible sets obtained from branch and bound algorithm are presented, leading to the optimal solution (P0-P1-P4-P5-P8).

## 5. A COMPARISON OF PERFORMANCE OF THE ALGORITHMS FOR SELECTED BINARY AND INTEGER LP PROBLEMS

### 5.1. INTRODUCTION

The benchmark has been divided into two parts. The first part is devoted to binary LPs, the second concerns integer LPs.

The first part of the test has been very time-consuming, since Balas algorithms have expected computational complexity proportional to  $2^n$ , on the contrary to partial enumeration method which is an effective algorithm. For the given combination of  $n = 1, 2, \dots, 11$  and  $m = 1, 2, \dots, 19$ , a hundred of random problems have been generated, and for the given combination of  $n = 1, 2, \dots, 15$  and  $m = 1, 2, \dots, 19$  a lesser number of random problems have been generated due to the time needed to solve them. 40 random problems have been



generated for  $n = 12, 30$  for  $n = 13, 20$  for  $n = 14$ , and 10 for  $n = 15$ . Separate test have been carried out for plain Balas method and methods with constant and dynamic filter. It has taken one month to perform all the computation of this part using 8 PC class computers. Partial enumeration algorithm has been run in parallel with Balas method for the same problems

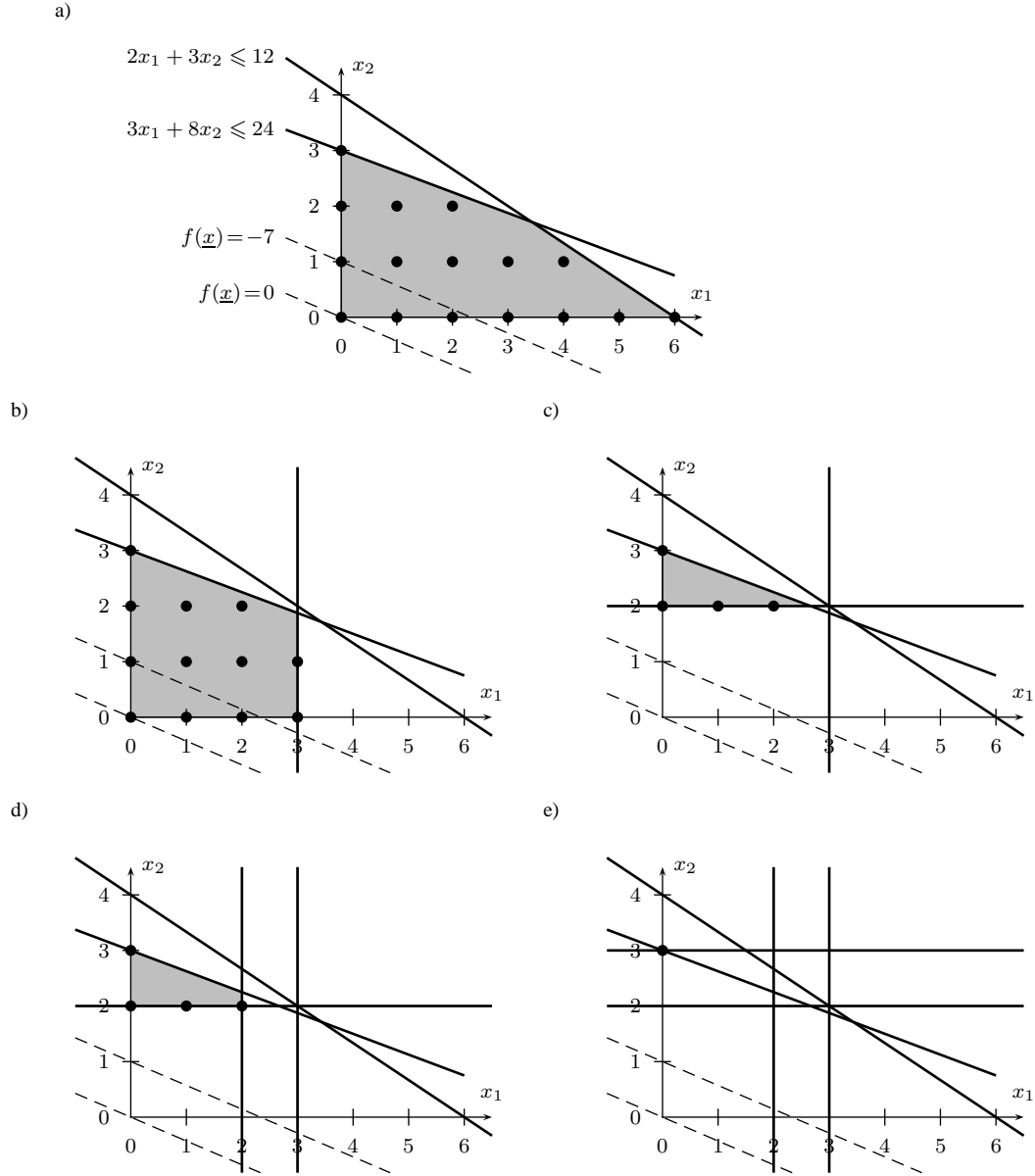


Fig. 5. Branch and bound solutions to the given problem : a) P0, b) P1, c) P4, d) P5, e) P8

generated.

In the second part of the algorithm, the problems for branch and bound method have been generated in the span of  $n = 1, 2, \dots, 20$  and  $m = 1, 2, \dots, 20$ , where for each structure there has been 100 of random problems generated.

## 5.2. THE RESULTS

In the Table 2, performance evaluation results are presented, expressed as mean relative number operations per problem for selected sizes of the LPs with binary constraints. The numbers are given with respect to maximum number of operations to-be-performed, i.e.  $(m+1)2^n$ . The results are presented in the range 0–100%. In comparison, Table 3 presents results for the same problems but for partial enumeration algorithm. In order not to introduce any distortion to the results, the results for this algorithm are also presented in the same scale, but instead of presenting the relative number of operations it refers to the number of main iterations of the algorithm. Selected cases have been presented in Table 4 and depicted in Figure 8.

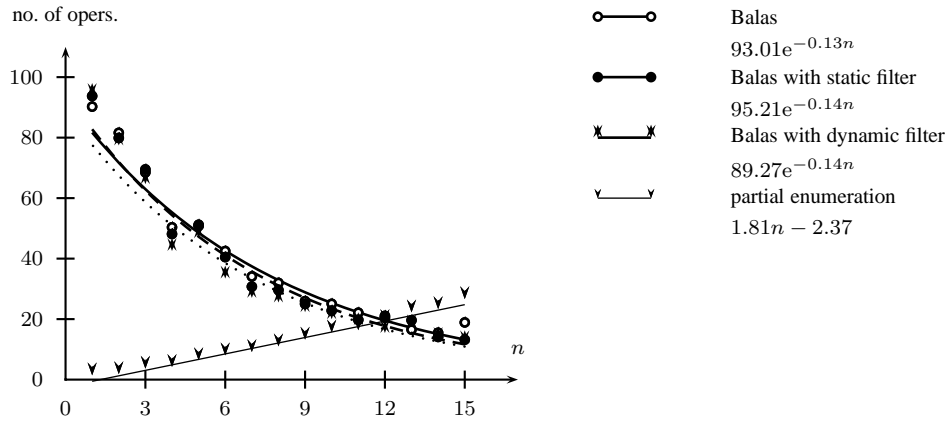
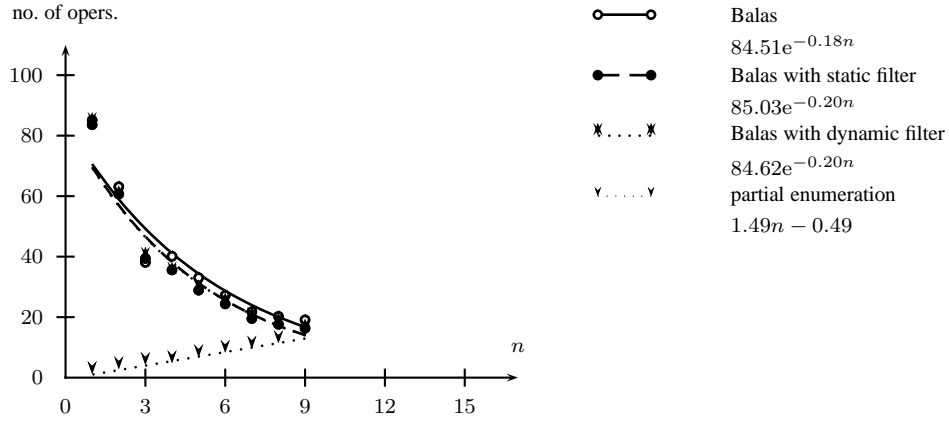
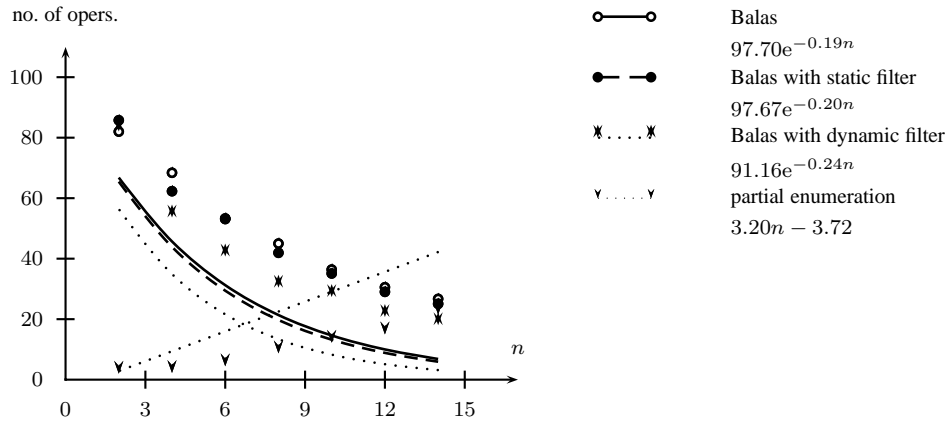
## 6. SUMMARY

Balas methods have unattractive worst-case computational complexity. The relative computational burden decreases monotonically with increase in size of the task. This is because there is only a part of the constraints computed during the run of an algorithm. The greater the number  $m$  of the constraints is, the greater the number of constraints are omitted in feasibility test. By introduction of the constant filter, the situation slightly improves, what is especially visible for large number of constraints. It is to be borne in mind, that still the computational complexity is exponential. On the contrary, partial enumeration method has linear complexity trend line, increasing with the increase in  $n$ .

The branch and bound method has mostly appealing computational complexity, i.e. linear, strongly connected to number  $n$  of variables, and with little dependence on the number  $m$  of the constraints.

## REFERENCES

- [1] Balas E., *An Additive Algorithm for Solving Linear Programs with Zero-one Variables*, Operations Research, Vol. 13(4), 1965, pp. 517–546.
- [2] Bazaraa M.S., Sherali H.D., Shetty C.M., *Nonlinear Programming: Theory and Algorithms*, 3rd ed., New Jersey, Wiley 2006.
- [3] Dantzig G.B., Thapa M.N., *Linear Programming 1: Introduction*, Springer Verlag 1997.
- [4] Dantzig G.B., Thapa M.N., *Linear Programming 2: Theory and Extensions*, Springer 2003.
- [5] Horla D., *Metody obliczeniowe optymalizacji w zadaniach*, Poznań, Wydawnictwo Politechniki Poznańskiej, 2008 (in Polish).
- [6] Pedregal P., *Introduction to Optimization*, New York, Springer-Verlag 2004.
- [7] Stadnicki J., *Teoria i praktyka rozwiązywania zadań optymalizacji z przykładami zastosowań technicznych*, Warszawa, Wydawnictwa Naukowo-Techniczne 2006 (in Polish).
- [8] Sysło M.M., Deo N., Kowalik J.S., *Algorytmy optymalizacji dyskretnej z programami w języku Pascal*, Warszawa, Wydawnictwo Naukowe PWN 1995 (in Polish).

Fig. 6. Mean number of operations for binary LPs and  $\frac{m}{n} = 1$ Fig. 7. Mean number of operations for binary LPs and  $\frac{m}{n} = 2$ Fig. 8. Mean number of operations for binary LPs and  $\frac{m}{n} = 0.5$

Tab. 2. Performance evaluation of a family of Balas methods: a) classical, b) with constant filter, c) with dynamic filter

$m \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a) 1	90.25	82.01	81.63	77.52	77.99	77.31	76.43	76.81	75.09	76.52	73.86	76.42	78.73	85.20	81.00
2	85.10	81.60	70.50	68.40	70.00	71.60	64.50	55.90	54.90	68.60	57.60	60.78	59.97	59.55	53.50
3	79.10	73.20	68.50	66.60	55.80	53.30	47.40	49.10	45.50	46.20	50.90	49.73	53.20	52.13	43.00
4	77.70	63.10	60.27	50.36	50.71	45.85	44.86	45.00	43.43	41.83	40.82	40.32	36.33	40.40	37.70
5	79.20	59.20	51.80	55.60	50.80	39.30	48.00	33.00	33.00	36.40	35.20	36.30	38.30	34.63	34.90
6	77.10	53.10	38.10	49.00	39.20	42.50	31.90	34.50	39.00	34.80	32.60	30.55	31.83	30.10	30.00
7	67.84	53.76	47.24	42.50	37.65	36.37	34.10	32.00	30.23	27.98	30.26	32.34	28.57	26.70	26.20
8	69.32	50.95	44.04	40.12	36.85	35.35	31.21	31.97	29.46	28.22	25.78	27.40	24.70	26.25	31.80
9	65.45	53.16	42.43	37.33	33.54	29.88	29.38	26.16	25.95	25.99	24.95	23.93	20.30	22.80	23.00
10	66.76	49.35	41.77	35.44	33.03	30.23	27.15	26.61	24.18	25.06	24.04	22.51	24.30	21.30	21.60
11	63.40	50.04	39.02	34.29	30.04	27.20	25.79	24.20	22.97	22.22	22.13	20.90	22.63	18.70	18.10
12	65.14	46.58	37.64	32.93	28.74	27.19	24.27	23.43	22.76	21.69	21.83	20.45	20.97	19.90	15.70
13	63.82	43.72	34.56	31.87	26.77	26.75	24.50	20.94	21.46	20.91	19.85	19.59	16.53	18.15	16.00
14	63.24	45.84	33.69	28.04	24.24	22.64	21.78	20.64	20.96	18.01	19.11	18.17	18.60	15.50	17.80
15	59.71	44.18	33.39	28.03	25.02	23.68	20.70	19.93	18.98	18.67	18.69	17.98	18.43	16.70	18.90
16	62.34	40.90	33.14	29.04	24.08	21.67	20.33	20.23	19.42	18.89	17.57	17.94	16.83	14.45	15.50
17	60.30	40.89	33.10	25.63	23.03	21.71	18.73	18.40	17.52	18.05	17.27	16.25	15.17	16.85	12.40
18	59.17	40.48	30.68	23.73	23.36	21.00	19.25	17.49	19.09	15.43	15.21	15.37	16.87	14.95	11.20
19	59.69	41.56	30.40	24.74	21.68	19.10	18.48	17.35	16.94	15.94	15.06	15.46	14.86	14.35	12.90
$m \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b) 1	93.75	85.76	92.10	86.27	87.10	87.31	87.92	91.30	85.44	92.17	93.26	95.65	93.50	83.97	92.85
2	83.60	79.90	70.10	62.30	76.50	58.60	61.00	57.80	66.40	69.20	65.50	67.58	69.60	65.15	61.60
3	82.90	75.20	69.50	57.30	47.20	53.10	49.00	45.60	54.30	48.70	53.50	51.08	53.28	51.80	46.85
4	74.10	60.70	54.77	48.14	45.61	43.01	43.43	41.94	40.93	41.02	42.90	38.95	36.10	34.60	41.80
5	72.40	54.80	43.50	54.30	51.20	38.10	47.20	31.90	32.40	35.10	36.40	36.28	35.28	34.30	33.95
6	73.40	54.00	39.40	47.10	35.60	40.50	30.40	32.40	34.10	32.90	31.10	29.03	31.53	26.25	29.40
7	64.37	51.02	42.32	37.91	34.08	32.17	30.75	30.02	27.97	27.37	27.25	26.53	28.05	25.07	25.00
8	66.96	49.46	40.37	35.60	33.44	30.75	28.21	29.60	27.91	26.04	24.85	24.68	25.30	21.75	24.60
9	64.60	47.31	39.59	32.74	30.77	27.37	27.15	24.52	25.03	24.00	23.84	24.18	22.95	22.93	22.00
10	64.72	45.44	37.34	31.26	28.91	25.69	24.30	23.78	21.34	22.82	20.90	19.58	22.00	20.50	20.40
11	62.48	47.03	34.77	29.64	26.92	24.38	22.39	21.83	21.18	21.43	19.76	21.08	20.13	17.40	21.00
12	62.48	47.03	34.77	29.64	26.92	24.38	22.39	21.83	21.18	21.43	19.76	21.08	20.13	17.40	21.00
13	61.87	42.53	31.65	29.01	25.09	22.32	20.90	18.40	19.21	18.80	19.12	21.60	19.63	15.53	17.85
14	62.68	43.48	30.48	25.44	21.54	20.77	19.52	18.51	18.59	15.95	17.36	16.33	16.97	14.10	15.00
15	58.80	42.15	31.18	24.47	23.27	20.61	18.60	17.77	17.52	16.74	16.83	18.20	17.23	15.33	13.20
16	60.56	39.04	30.70	25.70	21.26	19.57	18.44	17.62	17.42	15.65	16.09	15.80	14.67	14.95	11.20
17	58.89	39.32	31.03	23.30	19.93	19.26	16.79	17.03	16.26	16.26	14.90	8.65	7.93	13.40	11.15
18	57.56	38.63	27.80	21.62	20.82	17.66	17.20	15.84	16.35	13.22	14.20	14.38	13.90	14.10	11.50
19	58.90	39.77	27.57	22.15	19.09	17.86	16.07	15.23	14.05	14.07	13.67	14.50	13.95	13.00	11.90
$m \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c) 1	93.75	82.02	81.60	68.88	65.29	60.78	58.39	56.22	54.94	53.60	53.50	52.68	53.45	50.40	50.25
2	83.60	77.40	67.60	53.60	54.80	43.20	41.10	40.40	42.20	35.40	35.30	36.78	36.40	34.15	34.70
3	82.90	70.20	64.60	50.20	41.40	40.70	37.90	34.60	34.70	33.20	29.10	30.23	30.00	28.50	27.80
4	74.10	59.40	51.46	42.50	38.76	36.45	33.41	30.39	29.71	27.51	26.88	23.60	25.40	21.55	22.90
5	72.40	54.40	43.10	46.50	46.70	34.30	29.40	25.00	27.30	27.30	25.00	22.90	21.43	20.70	20.45
6	73.40	54.00	39.10	46.00	32.90	33.40	28.60	22.40	26.60	24.20	20.10	20.70	22.40	17.80	17.70
7	64.37	50.48	41.67	36.94	32.40	29.19	27.00	25.80	23.69	21.94	18.90	19.18	18.75	18.00	15.13
8	66.96	49.46	39.98	34.13	31.19	28.01	25.56	25.48	23.06	21.27	19.44	18.10	19.33	17.25	13.20
9	64.60	47.31	39.22	32.03	29.44	25.86	25.16	23.01	22.28	19.82	19.58	19.63	19.15	16.30	11.25
10	64.72	45.42	37.25	30.59	28.20	24.2	22.26	21.85	19.19	19.94	17.95	15.65	17.83	16.05	12.60
11	62.48	46.99	34.61	29.28	25.87	23.28	21.04	20.75	18.94	18.74	17.97	18.90	18.15	14.23	15.25
12	63.30	44.03	33.05	28.45	25.17	23.96	20.83	20.27	18.12	17.67	16.89	15.38	16.30	15.90	14.50
13	61.87	42.53	31.57	28.67	24.91	21.62	20.30	17.79	18.12	17.15	17.44	19.53	17.28	12.07	16.00
14	62.68	43.48	30.33	25.22	21.00	20.54	19.14	17.70	17.15	15.09	16.03	15.30	13.70	13.10	13.30
15	58.80	42.12	31.18	24.38	22.60	20.25	18.24	17.38	16.65	16.17	15.78	16.15	16.00	12.83	12.00
16	60.56	39.04	30.61	25.57	21.08	19.35	17.90	17.42	16.48	14.79	15.10	14.18	12.87	13.70	10.30
17	58.89	39.32	30.93	23.19	19.85	19.09	16.48	16.41	15.87	15.55	14.17	7.15	7.45	12.07	9.25
18	57.56	38.63	27.80	21.62	20.81	17.45	16.98	15.56	15.36	12.75	13.81	13.60	12.73	13.40	10.25
19	58.90	39.77	27.57	22.13	19.01	17.74	15.91	15.01	13.85	13.71	13.15	13.525	12.50	11.90	10.95

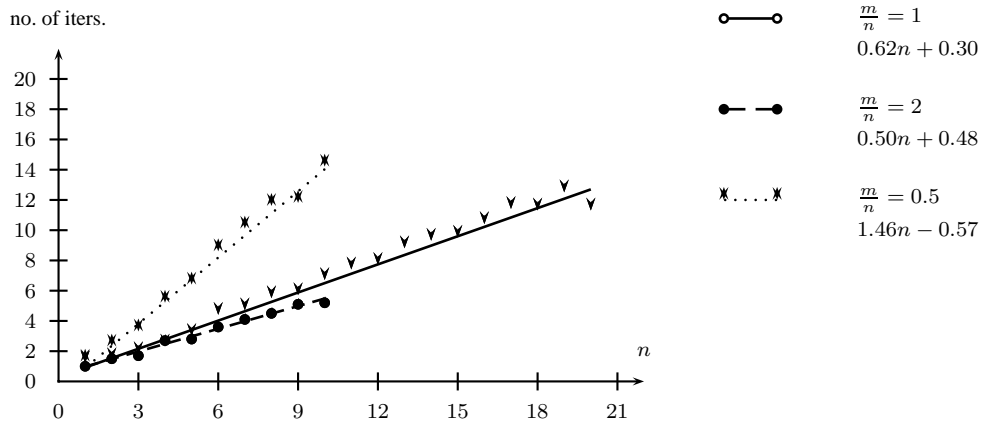


Fig. 9. Mean number of iterations for integer LPs

Tab. 3. Performance evaluation of the partial enumeration algorithm

$m \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.40	1.96	2.20	2.64	2.78	3.06	3.14	3.52	4.22	4.08	4.70	5.00	5.27	4.30	5.60
2	1.20	1.80	2.80	2.20	3.60	3.40	6.80	7.80	5.20	3.00	6.20	8.10	7.87	7.00	7.60
3	2.00	1.60	3.60	4.40	5.40	4.40	7.00	9.20	9.00	11.00	9.00	9.80	10.80	10.40	13.00
4	1.74	2.82	3.18	4.18	5.18	6.66	7.72	8.68	9.52	10.88	11.78	11.02	14.27	12.50	14.00
5	1.80	2.40	2.80	4.20	6.40	7.20	6.60	10.20	10.20	12.20	13.00	14.30	12.40	16.40	15.90
6	1.40	3.00	4.20	5.80	5.40	8.00	10.40	10.60	8.60	13.60	12.20	14.95	18.73	18.50	18.60
7	1.82	2.88	3.82	4.82	6.04	7.30	9.10	10.66	12.36	14.36	13.66	17.22	17.60	21.50	19.00
8	1.98	3.26	4.06	4.80	6.20	7.12	9.64	10.98	12.40	14.06	16.26	17.72	19.87	17.30	19.60
9	1.92	2.76	4.18	4.84	6.40	7.88	9.46	11.54	13.20	13.68	15.92	18.18	19.33	20.80	24.40
10	2.06	2.70	4.20	5.22	6.78	7.24	9.42	11.30	14.08	15.56	16.18	17.74	19.93	20.50	23.40
11	2.04	2.96	3.94	5.22	6.58	7.90	9.08	11.22	13.80	15.32	16.32	19.76	21.60	21.70	23.00
12	2.00	3.14	3.86	5.04	6.40	8.26	9.94	12.14	13.04	15.74	16.68	19.02	20.53	23.50	23.00
13	1.94	3.14	4.20	5.36	7.04	8.08	9.62	11.92	13.16	15.40	17.98	19.14	22.20	23.50	27.00
14	2.02	2.94	3.86	5.30	6.46	8.36	9.62	11.46	13.96	15.44	17.46	19.94	20.00	23.30	22.60
15	1.90	3.06	4.30	5.14	6.48	8.24	9.74	11.50	13.02	16.02	17.70	19.84	21.60	22.00	26.60
16	1.88	3.06	4.00	5.16	7.06	8.44	9.52	11.40	13.76	14.32	17.28	19.52	21.13	24.90	26.60
17	2.04	3.04	4.02	5.34	7.00	7.86	9.56	11.64	13.86	16.40	17.30	19.90	21.53	25.80	27.20
18	1.90	2.96	4.00	5.32	6.62	8.02	9.36	12.08	13.52	14.40	18.44	19.96	20.87	24.40	24.80
19	1.96	3.38	4.06	5.16	6.26	8.12	9.80	11.04	12.88	14.96	17.80	20.20	22.16	24.16	26.40

Tab. 4. Comparison of performance of algorithms for binary LPs a)  $\frac{m}{n} = 1$ , b)  $\frac{m}{n} = 2$ , c)  $\frac{m}{n} = 0.5$ 

a)	$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter	Partial enumeration
	1	1	90.25	93.75	93.75	1.40
	2	2	81.60	79.90	77.40	1.80
	3	3	68.50	69.50	64.60	3.60
	4	4	50.36	48.14	42.50	4.18
	5	5	50.80	51.20	46.70	6.40
	6	6	42.50	40.50	33.40	8.00
	7	7	34.10	30.75	27.00	9.10
	8	8	31.97	29.60	25.48	10.98
	9	9	25.95	25.03	22.28	13.20
	10	10	25.06	22.82	19.94	15.56
	11	11	22.13	19.76	17.97	16.32
	12	12	20.45	21.08	15.38	19.02
	13	13	16.53	19.63	17.28	22.20
	14	14	15.50	14.10	13.10	23.30
	15	15	18.90	13.20	12.00	26.60

b)	$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter	Partial enumeration
	2	1	85.10	83.60	83.60	1.20
	4	2	63.10	60.70	59.40	2.82
	6	3	38.10	39.40	39.10	4.20
	8	4	40.12	35.60	34.13	4.80
	10	5	33.03	28.91	28.20	6.78
	12	6	27.19	24.38	23.96	8.26
	14	7	21.78	19.52	19.14	9.62
	16	8	20.23	17.62	17.42	11.40
	18	9	19.09	16.35	15.36	13.52

c)	$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter	Partial enumeration
	1	2	82.01	85.76	82.02	1.96
	2	4	68.40	62.30	53.60	2.20
	3	6	53.30	53.10	40.70	4.40
	4	8	45.00	41.94	30.39	8.68
	5	10	36.40	35.10	27.30	12.20
	6	12	30.55	29.03	20.70	14.95
	7	14	26.70	25.07	18.00	21.50

Tab. 5. Comparison of computational burden of algorithms for binary LPs a)  $\frac{m}{n} = 1$ , b)  $\frac{m}{n} = 2$ , c)  
 $\frac{m}{n} = 0.5$

$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter
1	1	3.61	3.75	3.75
2	2	9.79	9.59	9.29
3	3	21.92	22.24	20.67
4	4	40.29	38.51	34.00
5	5	97.54	98.30	89.66
6	6	190.40	181.44	149.63
7	7	349.18	314.88	276.48
8	8	736.59	681.98	587.06
9	9	1.328.64	1.281.54	1.140.74
10	10	2.822.76	2.570.44	2.246.04
11	11	5.438.67	4.856.22	4.416.31
12	12	10.889.22	11.222.02	8.186.88
13	13	18.961.75	22.507.52	19.812.35
14	14	38.092.80	34.652.16	32.194.56
15	15	99.090.43	80.600.50	62.914.56

$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter
2	1	5.11	5.02	5.02
4	2	12.62	12.14	11.88
6	3	21.34	22.06	21.90
8	4	57.77	51.26	49.15
10	5	116.27	101.76	99.26
12	6	226.22	202.84	199.35
14	7	418.18	374.78	367.49
16	8	880.41	766.82	758.12
18	9	1.857.08	1.590.53	1.494.22

$m$	$n$	Balas	Balas with constant filter	Balas with dynamic filter
1	2	7.52	7.45	6.69
2	4	31.15	30.53	26.33
3	6	128.87	125.06	103.61
4	8	533.17	512.26	407.74
5	10	2.205.78	2.098.20	1.604.59
6	12	9.125.58	8.594.14	6.314.65
7	14	37.753.61	35.201.24	24.850.38

Tab. 6. Performance evaluation of the branch and bound method for integer LPs

$m \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1.2	1.3	1.9	1.6	2.5	2.6	3.3	4.1	4.0	4.0	4.8	5.8	5.6	5.5	5.5	6.7	5.9	7.5	8.1	8.1
2	1.0	1.4	1.9	2.3	2.4	3.0	3.0	3.4	6.0	4.9	6.0	5.5	5.6	6.0	5.4	6.9	6.2	6.8	7.2	8.3
3	1.0	1.6	1.8	3.5	2.5	3.3	4.1	4.5	4.7	5.7	5.2	6.6	7.1	7.4	8.0	8.5	8.3	9.7	10.6	10.6
4	1.0	1.5	2.1	2.3	3.1	4.3	5.3	5.2	5.1	6.4	6.5	7.2	7.8	8.7	9.3	9.6	10.2	11.0	10.2	10.8
5	1.0	1.3	1.9	2.5	3.0	3.9	4.7	4.9	7.2	6.4	7.8	7.5	8.6	10.1	9.3	9.5	9.5	11.8	12.3	12.1
6	1.0	1.3	1.7	2.5	2.8	4.4	4.9	4.8	6.5	6.9	7.4	8.6	8.3	8.5	10.8	11.0	10.6	11.1	12.1	13.9
7	1.0	1.3	1.8	2.7	3.3	4.5	4.7	5.3	5.5	7.8	7.2	8.2	8.5	10.1	9.3	12.8	12.3	12.0	13.9	13.3
8	1.0	1.3	1.9	2.7	2.9	3.9	4.8	5.5	5.6	6.4	8.0	8.8	8.2	11.2	11.2	11.6	12.9	12.4	14.2	14.4
9	1.0	1.2	1.8	2.9	3.0	4.3	4.0	5.0	5.7	7.1	7.6	7.2	9.3	10.4	10.3	10.1	12.8	11.8	12.2	14.0
10	1.0	1.2	1.6	2.2	2.8	3.8	3.9	5.3	6.3	6.7	6.7	7.9	8.6	9.4	10.7	11.4	12.3	13.6	13.4	14.2
11	1.0	1.2	1.6	2.3	2.7	3.3	4.1	5.5	5.4	6.5	7.4	9.6	9.4	9.9	10.1	10.8	12.5	12.3	12.4	15.6
12	1.0	1.2	1.7	2.1	2.6	3.6	4.4	4.7	6.1	6.3	7.2	7.7	9.5	10.5	11.0	10.2	11.3	12.7	12.7	14.1
13	1.0	1.2	1.5	2.1	2.5	3.5	4.2	4.2	5.2	6.2	7.1	8.0	8.8	9.5	9.8	10.5	12.6	12.1	13.9	14.0
14	1.0	1.2	1.8	2.1	2.6	3.5	4.1	4.8	5.5	7.1	6.7	7.4	9.0	9.3	10.5	11.2	12.6	12.6	12.7	14.0
15	1.0	1.2	1.6	2.0	2.5	3.5	3.9	4.2	5.6	6.4	6.3	6.8	8.8	9.4	9.5	10.0	10.8	11.9	12.8	13.9
16	1.0	1.1	1.5	2.4	2.3	3.7	3.5	4.5	5.5	6.2	6.4	8.0	8.5	9.4	8.7	10.4	11.0	11.4	12.9	14.1
17	1.0	1.1	1.5	1.9	2.4	3.0	3.9	4.7	4.8	5.7	6.4	7.1	8.7	9.0	9.5	9.6	11.4	12.3	12.9	12.9
18	1.0	1.1	1.5	1.8	2.5	3.0	3.4	4.4	5.1	5.8	5.9	7.2	7.8	8.9	9.4	9.3	11.3	11.3	11.6	13.5
19	1.0	1.2	1.3	1.8	2.4	2.9	3.2	4.0	4.5	5.6	6.4	7.2	7.4	8.7	8.5	9.5	10.7	11.0	12.5	11.7
20	1.0	1.1	1.4	2.0	2.5	2.9	3.8	4.3	4.5	5.2	5.8	6.7	7.1	8.7	8.8	9.5	9.8	11.1	10.8	11.3

Tab. 7. Performance evaluation for integer LPs a)  $\frac{m}{n} = 1$ , b)  $\frac{m}{n} = 2$ , c)  $\frac{m}{n} = 0.5$

$m$	$n$	$m+n$	iterations
1	1	2	1.2
2	2	4	1.4
3	3	6	1.8
4	4	8	2.3
5	5	10	3.0
6	6	12	4.4
7	7	14	4.7
8	8	16	5.5
9	9	18	5.7
10	10	20	6.7
11	11	22	7.4
12	12	24	7.7
13	13	26	8.8
14	14	28	9.3
15	15	30	9.5
16	16	32	10.4
17	17	34	11.4
18	18	36	11.3
19	19	38	12.5
20	20	40	11.3

$m$	$n$	$m+n$	iterations
2	1	3	1.0
4	2	6	1.5
6	3	9	1.7
8	4	12	2.7
10	5	15	2.8
12	6	18	3.6
14	7	21	4.1
16	8	24	4.5
18	9	27	5.1
20	10	30	5.2

$m$	$n$	$m+n$	iterations
1	2	3	1.3
2	4	6	2.3
3	6	9	3.3
4	8	12	5.2
5	10	15	6.4
6	12	18	8.6
7	14	21	10.1
8	16	24	11.6
9	18	27	11.8
10	20	30	14.2



- [9] Vanderbei R.J., *Linear Programming: Foundations and Extensions*, 2nd ed, Springer 2001.  
[10] Venkataraman P., *Applied Optimization with Matlab Programming*, Wiley 2009.

#### ABSTRACT

The paper considers performance issues of a class of iterative minimization methods of binary and linear programs. Problem structures that assure superior performance of a specific method have been stipulated with appropriate conclusions drawn.

#### OCENA SZYBKOŚCI DZIAŁANIA METOD MINIMALIZACJI DLA ZADAŃ PROGRAMOWANIA LINIOWEGO W ZBIORACH DYSKRETNYCH

Paweł Kaden, Dariusz Horla

W artykule poruszono zagadnienie szybkości działania metod minimalizacji w zbiorach dyskretnych (binarne i całkowitoliczbowe) dla zadań programowania liniowego. Wskazano przypadki, dla których konkretna metoda działa szybciej niż pozostałe oraz wyciągnięto wnioski odnośnie takiego stanu rzeczy.

Received: 2014-12-01

Accepted: 2015-02-16