Scientific Research Publishing

American Journal of Operations Research, 2018, 8, 92-111
http://www.scirp.org/journal/ajor
ISSN Online: 2160-8849
ISSN Print: 2160-8830

The Pivot Adaptive Method for Solving Linear Programming Problems

Saliha Belahcene[1], Philippe Marthon[2], Mohamed Aidene[3]

[1]Mouloud Mammeri University of Tizi-Ouzou, Tizi-Ouzou, Algeria
[2]Site ENSEEIHT de l'IRIT, Université Fédérale Toulouse Midi-Pyrénées, Toulouse, France
[3]L2CSP Laboratory Design and Control Systems Production, Tizi-Ouzou, Algeria
Email: saliha.belahcene@yahoo.com, philippe.marthon@enseeiht.fr, aidene@mail.ummto.dz

How to cite this paper: Belahcene, S., Marthon, P. and Aidene, M. (2018) The Pivot Adaptive Method for Solving Linear Programming Problems. *American Journal of Operations Research*, **8**, 92-111.
https://doi.org/10.4236/ajor.2018.82008

Received: February 13, 2018
Accepted: March 20, 2018
Published: March 23, 2018

Abstract

A new variant of the Adaptive Method (AM) of Gabasov is presented, to minimize the computation time. Unlike the original method and its some variants, we need not to compute the inverse of the basic matrix at each iteration, or to solve the linear systems with the basic matrix. In fact, to compute the new support feasible solution, the simplex pivoting rule is used by introducing a matrix that we will define. This variant is called "the Pivot Adaptive Method" (PAM); it allows presenting the resolution of a given problem under the shape of successive tables as we will see in example. The proofs that are not given by Gabasov will also be presented here, namely the proofs for the theorem of the optimality criterion and for the theorem of existence of an optimal support, and at the end, a brief comparison between our method and the Simplex Method will be given.

Keywords

Optimization, Linear Programming, Simplex Method, Adaptive Method

1. Introduction

As a branch of mathematics, linear programming is the domain that had the most successful in optimization [1] [2] [3] [4] [5]. Since its formulation from 1930 to 1940 and the development of the Simplex method of Dantzig in 1949 [1] [6], researchers in various fields have been led to formulate and solve linear problems.

Although, the Simplex algorithm is often effective in practice, the fact that it is not a polynomial algorithm, as shown by Klee and Minty [7], has incited researchers to propose other algorithms and led to the birth of the interior point algorithms.

In 1979, L. Khachiyan proposed the first polynomial algorithm for linear programming [8]; it is based on the ellipsoid method, studied by Arkadi Nemirovski and David B. Yudin, a preliminary version of which having been introduced by Naum Z. Shor. Unfortunately, this ellipsoid method has a poor efficiency.

In 1984, N. K. Karmakar published an interior point algorithm which has a polynomial convergence [9], and this caused a renewed interest to interior point methods, as well in linear programming that in nonlinear programming [10] [11] [12] [13] [14].

Gabasov and Kirillova have generalized the Simplex method in 1995 [15] [16] [17], and developed the Adaptive Method (AM), a primal-dual method, for linear programming with bounded variables. As the Simplex method, the Adaptive Method is a support method, but it can start from any support (base) and any feasible solution and can move to the optimal solution by interior or boundary points; this method was extended, later, to solve general linear and convex problems [18] [19] [20], and also optimal control problems [21].

In linear programming, several variants of the AM have been proposed. They generally address the problem of initialization of the method and the choice of the direction [22] [23].

In this work, a new variant of the adaptive method called "Pivot adaptive method" (PAM) is presented. Unlike the original method and its variants, we need not to compute the inverse of the basic matrix at each iteration [16], or to solve the linear systems with the basic matrix [23]. In fact, to compute the new feasible solution and the new support, we only need compute the decomposition of vectors columns of the matrix of the problem in the current basis. For this computation we use the simplex pivoting rule [4] by introducing a new matrix GAMMA which allows to minimize the computation time. This new variant of adaptive method allows us to present the resolution of a given problem under the shape of successive tables (as is done with the Simplex method) as we will see in example. The proofs for the theorem of the optimality criterion and for the theorem of existence of an optimal support that are not given by Gabasov [16] will be presented here, and without giving too many details a brief comparison between our method and the Simplex method will be given at the end of this article.

The paper is organized as follows. In Section 2, after giving the statement of the problem, some important definitions are given. In Section 3, we describe step by step the "Pivot Adaptive Method" and in parallel the proofs of the corresponding theorems are given. In Section 4, an example will be solved to illustrate the PAM, and more details will be given to explain how present the resolution under the shape of successive tables. In Section 5, we give a brief comparison between the PAM and the Simplex Method by solving the Klee-Minty problem. Section 6, concludes the paper.

2. Statement of the Problem and Definitions

In this article we consider the primal linear programming problem with bounded variables presented in the following standard form:

$$(P)\quad\begin{cases}\max F\left(x_1,x_2,\cdots,x_n\right)=\sum_{j=1}^{j=n}c_jx_j\\ \sum_{j=1}^{j=n}a_{ij}x_j=b_i,\, i\in I\\ d_j^-\le x_j\le d_j^+,\, j\in J.\end{cases}\tag{1}$$

where: $J=\{1,\cdots,n\}$ is the index set of the variables $x_1,x_2,\cdots,x_n$, $I=\{1,\cdots,m\}$ is the index set of the parameters $b_1,b_2,\cdots,b_m$ (corresponds to the constraints), put: $J=J_B\bigcup J_N$, $J_B\bigcap J_N=\varnothing$, $|J_B|=m$, and introduce the vectors:

- $x=x(J)=\left(x_j,j\in J\right)=\left(x_1,x_2,\cdots,x_n\right)=\begin{pmatrix}x_B\\x_N\end{pmatrix}$, with:

  $x_B=x\left(J_B\right)=\left(x_j,j\in J_B\right)$, $x_N=x\left(J_N\right)=\left(x_j,j\in J_N\right)$.
- $c=c(J)=\left(c_j,j\in J\right)=\left(c_1,c_2,\cdots,c_n\right)=\begin{pmatrix}c_B\\c_N\end{pmatrix}$, with:

  $c_B=c\left(J_B\right)=\left(c_j,j\in J_B\right)$, $c_N=c\left(J_N\right)=\left(c_j,j\in J_N\right)$.
- $b^{\mathrm{T}}=b^{\mathrm{T}}(J)=\left(b_i,i\in I\right)=\left(b_1,b_2,\cdots,b_m\right)$.
- $d^-=d^-(J)=\left(d_j^-,j\in J\right)=\left(d_1^-,d_2^-,\cdots,d_n^-\right);\left\|d^-\right\|<+\infty$.
- $d^+=d^+(J)=\left(d_j^+,j\in J\right)=\left(d_1^+,d_2^+,\cdots,d_n^+\right);\left\|d^+\right\|<+\infty$.

and the $(m\times n)$ matrix

- $A=A(I,J)=\left(a_{ij},i\in I,j\in J\right)=\left(a_1,a_2,\cdots,a_n\right)=\begin{pmatrix}A_1^{\mathrm{T}}\\\vdots\\A_m^{\mathrm{T}}\end{pmatrix}$, with: $a_j=\begin{pmatrix}a_{1j}\\\vdots\\a_{mj}\end{pmatrix}$,

  $j\in J$ (a_j: the j^{th} column of the matrix *A*), and $A_i^{\mathrm{T}}=\left(a_{i1},a_{i2},\cdots,a_{in}\right),i\in I$ (A_i^{T}: the i^{th} row of the matrix *A*), $A=\left(A_B/A_N\right)$, $A_B=A\left(I,J_B\right)$, $A_N=A\left(I,J_N\right)$.

We assume that $rank(A)=m<n$. Then the problem (1) takes the following form:

$$(P)\quad\begin{cases}\max F(x)=c^{\mathrm{T}}x\\ Ax=b\\ d^-\le x\le d^+\end{cases}\tag{2}$$

$Ax=b$ are called the general constraints of (*P*).

Denote the feasible region of (*P*) as: $H=\left\{x\in R^n,Ax=b,d^-\le x\le d^+\right\}$.

2.1. Definition 1

Each vector of the set *H* is called a feasible solution of (*P*).

2.2. Definition 2

Any vector $x\in R^n$ that satisfies the general constraints $Ax=b$ of the problem (*P*) is called a pseudo-feasible solution.

2.3. Definition 3

A feasible solution x^0 is called optimal if: $F\left(x^0\right)=c^{\mathrm{T}}x^0=\max\limits_{x\in H}c^{\mathrm{T}}x$.

2.4. Definition 4

For given value $\epsilon\geq 0$, the solution x^{ϵ} is called an ϵ-optimal (or suboptimal) if: $\left(F\left(x^0\right)-F\left(x^{\epsilon}\right)\right)\leq\epsilon$, where x^0 is an optimal solution for the problem (*P*).

2.5. Definition 5

The set of *m* indices $J_B\subset J$ ($\left|J_B\right|=m$) is called a support of (*P*) if the submatrix $A_B=A\left(I,J_B\right)$ is non-singular ($\det A_B\neq 0$), then the set $J_N=J\setminus J_B$ of ($n-m$) indices is called non-support, A_B is the matrix support, and $A_N=A\left(I,J_N\right)$ is the matrix non-support.

2.6. Definition 6

The pair $\left\{x,J_B\right\}$ formed by a feasible solution *x* and the support J_B is called a support feasible solution (SF-solution).

2.7. Definition 7

The SF-solution $\left\{x,J_B\right\}$ is called non-degenerate if: $d_j^-<x_j<d_j^+,\forall j\in J_B$.

Recall that, unlike the Simplex Method in which the feasible solution *x* and the basic indices J_B are related intimately, in the adaptive method there are independent.

3. Optimality Criterion

For the smooth running of calculations, in the rest of the article, we assume that the sets J_B and J_N are vectors of indices (*i.e.*: the order of indices in these sets is important and must be respected).

3.1. Formula of the Objective Value Increment

Let $\left\{x,J_B\right\}$ be the initial SF-solution of the problem (*P*), and $\bar{x}=\bar{x}\left(J\right)=\begin{pmatrix}\bar{x}_B\\ \bar{x}_N\end{pmatrix}$ an arbitrary pseudo-feasible solution of (*P*). We set $\Delta x=\Delta x_B+\Delta x_N=\bar{x}-x$, then the increment of the objective function value is given by:

$$\Delta F\left(x\right)=F\left(\bar{x}\right)-F\left(x\right)=c^{\mathrm{T}}\bar{x}-c^{\mathrm{T}}x=c^{\mathrm{T}}\Delta x=c_B^{\mathrm{T}}\Delta x_B+c_N^{\mathrm{T}}\Delta x_N \tag{3}$$

and since:

$$A\Delta x=A_B\Delta x_B+A_N\Delta x_N=A\bar{x}-Ax=0 \tag{4}$$

then

$$\Delta x_B=-A_B^{-1}A_N\Delta x_N \tag{5}$$

Substituting the vector Δx_B in (3) we get:

$$\Delta F(x) = \left(c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N\right) \Delta x_N \tag{6}$$

Here, we define the *m*-vector of multipliers *y* as follows:

$$y^{\mathrm{T}} = c_B^{\mathrm{T}} A_B^{-1} \tag{7}$$

In [16], Gabasov defines the support gradient and noted it Δ as: $\Delta^{\mathrm{T}} = y^{\mathrm{T}} A - c^{\mathrm{T}}$, but he also recognizes that for every $k \in J_N$, the k^{th} support derivative is equal to $-\Delta_k = c_k - y^{\mathrm{T}} a_k$, in fact, $-\Delta_k$ is the rate of change of the objective function when the k^{th} non-support components of the feasible solution x is increased and all the other non-support components are fixed, at the same time the support components are changed in such way to satisfy the constraints $Ax = b$.

Then, in the pivot adaptive method, we define the *n*-vector of reduced gains (or support gradient) δ as follows:

$$\delta^{\mathrm{T}} = -\Delta^{\mathrm{T}} = \left(\delta_j, j \in J\right) = c^{\mathrm{T}} - y^{\mathrm{T}} A = \left(\delta_B^{\mathrm{T}}, \delta_N^{\mathrm{T}}\right), \text{ where}: \delta_B^{\mathrm{T}} = 0, \delta_N^{\mathrm{T}} = c_N^{\mathrm{T}} - y^{\mathrm{T}} A_B \tag{8}$$

To reduce the computation time in the adaptive method of Gabasov [16], we define the ($m \times n$) matrix Γ as follows:

$$\Gamma = A_B^{-1} A = \left(\Gamma_B, \Gamma_N\right), \quad \text{where}: \Gamma_B = I_m, \Gamma_N = A_B^{-1} A_N \tag{9}$$

which is the decomposition of the columns of the matrix A on the support J_B.

We have: $\Gamma_N = \Gamma(I, J_N) = \left(\Gamma_{ij}, i \in I, j \in J_N\right)$, Γ_{ij} is the product of the i^{th} row of the matrix A_B^{-1} and the j^{th} column of the matrix A_N. Then the *n*-vector of support gradient given in (8) can be computed as follows:

$$\begin{aligned}&\delta^{\mathrm{T}} = \left(\delta_j, j \in J\right) = c^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A = c^{\mathrm{T}} - c_B^{\mathrm{T}} \Gamma = \left(\delta_B^{\mathrm{T}}, \delta_N^{\mathrm{T}}\right),\\ &\text{where}: \delta_B^{\mathrm{T}} = 0, \delta_N^{\mathrm{T}} = c_N^{\mathrm{T}} - c_B^{\mathrm{T}} \Gamma_N\end{aligned} \tag{10}$$

We see that, to compute the quantity $c_B^{\mathrm{T}} A_B^{-1} A$, we begin to compute $\Gamma = A_B^{-1} A$ and not $y^{\mathrm{T}} = c_B^{\mathrm{T}} A_B^{-1}$; therefore we need not to compute the inverse of the basic matrix A_B.

From (6) and (10) we obtain

$$\Delta F(x) = \delta_N^{\mathrm{T}} \Delta x_N = \sum_{j \in J_N} \delta_j \Delta x_j \tag{11}$$

3.2. Definition 8

For an given support J_B, any vector $\chi = \chi(J) = \begin{pmatrix} \chi_B \\ \chi_N \end{pmatrix}$ of R^n that satisfies

$$\begin{cases} \chi_j = d_j^-, & \text{if } \delta_j < 0 \\ \chi_j = d_j^+, & \text{if } \delta_j > 0 \\ \chi_j = d_j^- \text{ or } d_j^+, & \text{if } \delta_j = 0 \\ \chi_B = A_B^{-1}\left(b - A_N \chi_N\right) \end{cases} \quad \left(j \in J_N\right) \tag{12}$$

is called a primal pseudo-feasible solution accompanying the support J_B.

3.3. Proposition 1

A primal pseudo-feasible solution accompanying a given support J_B, χ, verifies just the general constraints of the problem (*P*), ie $A\chi = b$. If more, the support components χ_B of the vector χ verifies: $d_B^- \le \chi_B \le d_B^+$, then χ is feasible and optimal solution for (*P*).

3.4. Theorem 1 (The Optimality Criterion)

Let $\{x, J_B\}$ be an SF-solution of the problem (*P*), and δ^{T} be the support gradient computed by (10), then the relations

$$\begin{cases} \delta_j < 0, & \text{for } x_j = d_j^- \\ \delta_j > 0, & \text{for } x_j = d_j^+ \\ \delta_j = 0, & \text{for } d_j^- \le x_j \le d_j^+ \end{cases} \qquad \left(j \in J_N\right) \tag{13}$$

are sufficient, and in the case of non-degeneracy of the SF-solution $\{x, J_B\}$ also necessary for the optimality of the feasible solution *x*.

Proof

1) The sufficient condition:

for the SF-solution $\{x, J_B\}$, suppose that the relation (13) are verified, we must proof that x is optimal, it amounts to show that:

$$\forall \bar{x} \in H, \Delta F\left(x\right) = F\left(x\right) - F\left(\bar{x}\right) \ge 0 \tag{14}$$

Let $\bar{x} \in H$, and $\Delta x = x - \bar{x}$, then:

$$\begin{aligned} \Delta F\left(x\right) = F\left(x\right) - F\left(\bar{x}\right) &= \sum_{j \in J_N} \delta_j \Delta x_j \\ &= \sum_{\substack{j \in J_N \\ \delta_j \le 0}} \delta_j \left(x_j - \bar{x}_j\right) + \sum_{\substack{j \in J_N \\ \delta_j \ge 0}} \delta_j \left(x_j - \bar{x}_j\right) + \sum_{\substack{j \in J_N \\ \delta_j = 0}} \delta_j \left(x_j - \bar{x}_j\right) \\ &= \sum_{\substack{j \in J_N \\ \delta_j \le 0}} \delta_j \left(d_j^- - \bar{x}_j\right) + \sum_{\substack{j \in J_N \\ \delta_j \ge 0}} \delta_j \left(d_j^+ - \bar{x}_j\right) \ge 0 \end{aligned}$$

then x is optimal.

2) The necessary condition:

Suppose that $\{x, J_B\}$ an SF-solution non-degenerate, *i.e.*:

$$\forall j \in J_B,\ d_j^- < x_j < d_j^+ \tag{15}$$

and that x optimal, *i.e.*:

$$\forall \bar{x} \in H,\ \Delta F\left(x\right) = F\left(x\right) - F\left(\bar{x}\right) = \sum_{j \in J_N} \delta_j \Delta x_j \ge 0. \tag{16}$$

Reasoning by absurdity:

Suppose that the relations (13) are not verified, thus we have, at least, one of the following three cases:

1) $\exists k \in J_N / \delta_k \le 0$ and $x_k > d_k^-$.

2) $\exists k \in J_N / \delta_k \ge 0$ and $x_k < d_k^+$.

3) $\exists k \in J_N / \delta_k = 0$ and $x_k = d_j^-$ or $x_k = d_j^+$.

Suppose then we have the first case, *i.e.*

$$\exists k \in J_N / \delta_k \le 0 \text{ and } x_k > d_k^- \tag{17}$$

(the proof uses the same reasoning for the second and the third cases). Construct the *n*-vector $\bar{x}(\theta) = \left((\bar{x}(\theta))_1, (\bar{x}(\theta))_2, \cdots, (\bar{x}(\theta))_n\right)$ as follows:

$$\begin{cases} (\bar{x}(\theta))_j = x_j, & \forall j \in J_N \setminus \{k\} \\ (\bar{x}(\theta))_k = x_k - \theta, & \theta \in \left[0, x_k - d_k^-\right] \\ (\bar{x}(\theta))_j = x_j + \theta\Gamma_{jk}, & \forall j \in J_B \end{cases} \tag{18}$$

where Γ_{jk} is the product of the *j*th row of the matrix A_B^{-1} and the *k*th column of the matrix A_N. Let $\theta_1 = x_k - d_k^- > 0$.

we have

$$A\bar{x}(\theta) = A_B(\bar{x}(\theta))_B + A_N(\bar{x}(\theta))_N = 0 \tag{19}$$

and

$$\forall j \in J_N, d_j^- \le (\bar{x}(\theta))_j \le d_j^+ \tag{20}$$

For $j \in J_B, (\bar{x}(\theta))_j = x_j + \theta\Gamma_{jk}$, then for have:

$$\forall j \in J_B, d_j^- \le (\bar{x}(\theta))_j \le d_j^+ \tag{21}$$

a value of θ is chosen such that:

$$\forall j \in J_B, d_j^- \le x_j + \theta\Gamma_{jk} \le d_j^+ \tag{22}$$

i.e.

$$\begin{cases} 0 < \theta \le \dfrac{d_j^+ - x_j}{\Gamma_{jk}}, \text{ for } \Gamma_{jk} > 0 \\ 0 < \theta \le \dfrac{d_j^- - x_j}{\Gamma_{jk}}, \text{ for } \Gamma_{jk} < 0 \end{cases} \quad (j \in J_B) \tag{23}$$

Put: $\theta_2 = \min\limits_{\substack{\Gamma_{jk}>0 \\ j\in J_B}} \dfrac{d_j^+ - x_j}{\Gamma_{jk}}$, $\theta_3 = \min\limits_{\substack{\Gamma_{jk}<0 \\ j\in J_B}} \dfrac{d_j^- - x_j}{\Gamma_{jk}}$.

we have: $\theta_2 > 0$, $\theta_3 > 0$, because $\{x, J_B\}$ is non-degenerate.

Let $\theta^* = \min(\theta_1, \theta_2, \theta_3)$, then

$$\bar{x}(\theta^*) \in H \tag{24}$$

or

$$F(x) - F(\bar{x}(\theta^*)) = \sum_{j\in J_N} \delta_j \Delta x_j = \delta_k \Delta x_k = \delta_k \theta^* < 0 \tag{25}$$

then

$$F(\bar{x}(\theta^*)) > F(x) \tag{26}$$

contradiction with (16).

3.5. Definition 9

The pair $\{x^0, J_B^0\}$ satisfying the relations (13) will be called an optimal SF-solution of the problem (*P*).

3.6. The Suboptimality Criterion

Let x^0 an optimal solution of (*P*), and *x* a feasible solutions of (*P*).

From (11) we obtained:

$$\begin{aligned}\max_{\bar{x}\in H} \Delta F(x) &= F(x^0) - F(x)\\ &= \max_{\bar{x}\in H} \sum_{\substack{j\in J_N\\ \delta_j>0}} \delta_j(\bar{x}_j - x_j) + \max_{\bar{x}\in H} \sum_{\substack{j\in J_N\\ \delta_j<0}} \delta_j(\bar{x}_j - x_j)\\ &\le \sum_{\substack{j\in J_N\\ \delta_j>0}} \delta_j(d_j^+ - x_j) + \sum_{\substack{j\in J_N\\ \delta_j<0}} \delta_j(d_j^- - x_j)\end{aligned}$$

Let

$$\beta(x, J_B) = \sum_{\substack{j\in J_N\\ \delta_j>0}} \delta_j(d_j^+ - x_j) + \sum_{\substack{j\in J_N\\ \delta_j<0}} \delta_j(d_j^- - x_j) \tag{27}$$

$\beta(x, J_B)$ is called the suboptimality estimate of the SF-solution $\{x, J_B\}$ since

$$\left(F(x^0) - F(x)\right) \le \beta(x, J_B) \tag{28}$$

We have the following theorem of suboptimality.

3.7. Theorem 2 (Sufficient Condition of Suboptimality)

For a feasible solution *x* to be ϵ-optimal for given positive number ϵ, it is sufficient of the existence of a support J_B such that $\beta(x, J_B) \le \epsilon$.

3.8. Proposition 2

From (12) and (27) we deduce:

$$\beta(x, J_B) = \delta_N^{\mathrm{T}}(\chi_N - x_N) = \delta^{\mathrm{T}}(\chi - x) \tag{29}$$

3.9. The Dual Problem

The dual of the primal problem (*P*) is given by the following linear problem

$$(D)\quad \begin{cases}\min \phi(y, v, w) = b^{\mathrm{T}} y - (d^-)^{\mathrm{T}} v + (d^+)^{\mathrm{T}} w\\ A^{\mathrm{T}} y - v + w = c\\ y \in R^m, v \ge 0, w \ge 0.\end{cases} \tag{30}$$

where $A^{\mathrm{T}} = \begin{pmatrix} a_1^{\mathrm{T}} \\ \vdots \\ a_n^{\mathrm{T}} \end{pmatrix}$, with $\forall j \in J$, a_j^{T} is the transpose of the j^{th} column a_j of the matrix *A*. The problem *D* has *n* general constraints, and $(m + 2n)$ variables.

Like (*P*), the problem (*D*) has an optimal solution $\lambda^0=\left(y^0,v^0,w^0\right)$, and $F\left(x^0\right)=\phi\left(\lambda^0\right)$, where x^0 is the optimal solution of (*P*).

3.10. Definition 10

Any $(m+2n)$-vector $\lambda=(y,v,w)$ satisfying all the constraints of the problem D is called a dual feasible solution.

3.11. Definition 11

Let J_B be a support of the problem (*P*), and δ the corresponding support gradient defined in (10), the $(m+2n)$-vector $\lambda=(y,v,w)$ which satisfies:

$$\begin{cases} y^{\mathrm{T}}=c_B^{\mathrm{T}}A_B^{-1} \\ v_j=-\delta_j,\ w_j=0 \quad \text{if } \delta_j\le 0 \quad (j\in J) \\ v_j=0,\ w_j=\delta_j \quad \text{if } \delta_j>0 \end{cases} \tag{31}$$

is called the dual feasible solution accompanying the support J_B.

3.12. Proposition 3

- A dual feasible solution $\lambda=(y,v,w)$ accompanying a given support J_B, is dual feasible, *i.e.*: $A^{\mathrm{T}}y-v+w=c,\ v\ge 0,\ w\ge 0,\ y\in R^m$.
- For any support J_B we have

$$c^{\mathrm{T}}\chi=\phi(\lambda) \tag{32}$$

where λ and χ are respectively the dual feasible solution and an primal pseudo-feasible solution accompanying the support J_B.

3.13. Definition 12

Let y be dual feasible solution: $A^{\mathrm{T}}y-v+w=c,\ v\ge 0,\ w\ge 0$. The $(m+2n)$-vector $\lambda^*=\left(y,v^*,w^*\right)$ such that:

$$\begin{cases} v_j^*=0,\ w_j^*=\left(c_j-a_j^{\mathrm{T}}y\right) \quad \text{if } \left(c_j-a_j^{\mathrm{T}}y\right)\ge 0 \\ v_j^*=-\left(c_j-a_j^{\mathrm{T}}y\right),\ w_j^*=0 \quad \text{if } \left(c_j-a_j^{\mathrm{T}}y\right)\le 0 \end{cases} \quad (j\in J) \tag{33}$$

is called a coordinated dual feasible point to the *m*-vector *y*.

3.14. Proposition 4

The coordinated dual feasible point $\lambda^*=\left(y,v^*,w^*\right)$ to y is dual feasible, *i.e.*: $A^{\mathrm{T}}y-v^*+w^*=c$, $v^*\ge 0$, $w^*\ge 0$.

3.15. Decomposition of the Suboptimality Estimate of an SF-Solution $\{x,J_B\}$

Let $\{x,J_B\}$ be a SF-solution of the problem (*P*), $\beta\left(x,J_B\right)$ be its suboptimality estimate calculated by (27), $\lambda=(y,v,w)$ be the dual feasible solution accompanying the support J_B, χ the primal pseudo-feasible solution accompanying the support J_B, x^0 be the optimal solution for the problem (*P*), λ^0 be the optimal solution for the dual problem (*D*).

From (9), (10), (29), (32), and proposition 1 we have:

$$\begin{aligned}\beta(x,J_B)&=\delta_N^{\mathrm{T}}(\chi_N-x_N)=\delta^{\mathrm{T}}(\chi-x)=\left(c^{\mathrm{T}}-c_B^{\mathrm{T}}\Gamma\right)(\chi-x)\\&=c^{\mathrm{T}}\chi-c^{\mathrm{T}}x-c_B^{\mathrm{T}}A_B^{-1}A\chi+c_B^{\mathrm{T}}A_B^{-1}Ax=c^{\mathrm{T}}\chi-c^{\mathrm{T}}x\quad.\\&=\phi(\lambda)-F(x)=\phi(\lambda)-\phi\left(\lambda^0\right)+F\left(x^0\right)-F(x)\end{aligned}$$

then

$$\beta(x,J_B)=\beta(x)+\beta(J_B)\tag{34}$$

where: $\beta(J_B)=\phi(\lambda)-\phi\left(\lambda^0\right)$ is the degree of non-optimality of the support J_B, and $\beta(x)=F\left(x^0\right)-F(x)$ is the degree of non-optimality of the feasible solution *x*.

3.16. Proposition 5

- The feasible solution *x* will be optimal if $\beta(x)=0$, and the support J_B will be optimal if $\beta(J_B)=0$, but the pair $\{x,J_B\}$ will be an optimal SF-solution if $\beta(x)=\beta(J_B)=0$, so the optimality of the feasible solution *x* can be not identified if it is examined with an unfit support, because in this case there will be $\beta(J_B)\neq 0$, and $\beta(x,J_B)\neq 0$.
- As $\beta(J_B)=\phi(\lambda)-\phi\left(\lambda^0\right)$, then the support J_B will be optimal if its accompanying dual feasible solution λ is optimal solution for the dual problem *D*.

3.17. Theorem 3 (Existence of an Optimal Support)

For each problem (*P*), there is always an optimal support.

Proof

As (*P*) has an optimal solution, its dual (*D*) given by (30) has also an optimal solution, let $\bar{\lambda}=(\bar{y},\bar{v},\bar{w})$ this solution, and define the sets:

$$I^+=\left\{j\in J/\left(c_j-a_j^{\mathrm{T}}\bar{y}\right)\geq 0\right\},\quad I^-=\left\{j\in J/\left(c_j-a_j^{\mathrm{T}}\bar{y}\right)\leq 0\right\},$$

and

$$\bar{C}=\left\{y\in R^m/\left(c_j-a_j^{\mathrm{T}}y\right)\geq 0,\forall i\in I^+\text{ and }\left(c_j-a_j^{\mathrm{T}}y\right)\leq 0,\forall i\in I^-\right\},$$

we have $\bar{y}\in\bar{C}$.

Let $y\in\bar{C}$, and define the following linear problem:

$$\left(\bar{D}\right)\begin{cases}\min\psi(y)=b^{\mathrm{T}}y+\sum\limits_{j\in J^+}d_j^+\left(c_j-a_j^{\mathrm{T}}y\right)+\sum\limits_{j\in J^-}d_j^-\left(c_j-a_j^{\mathrm{T}}y\right)\\ y\in\bar{C}\subset R^m\end{cases}\tag{35}$$

We have $\psi(y)=\phi(y,v,w)$, where (y,v,w) is the coordinated dual feasible point to the vector *y*, and $\bar{y}$ is an optimal feasible solution of the problem ($\bar{D}$).

On the other hand, there exists a "vertex" y^* of the set $\bar{C}$ that is an optimal feasible solution for ($\bar{D}$), and this vertex is the intersection of at least *m* hyperplane $a_j^{\mathrm{T}}y^*=c_j,\ j\in\left(I^+\cup I^-\right)$.

Put $J_B=\left\{i\in\left(I^+\cup I^-\right)/a_j^{\mathrm{T}}y^*=c_j\right\}$ where $|J_B|=m$.

Then: $A^{\mathrm{T}}(I,J_B)y^* = c_B$, and so: $y^* = c_B^{\mathrm{T}}A_B^{-1}$, further the coordinated dual feasible point to y^* is the dual feasible solution accompanying the support J_B, and its optimal feasible solution for the problem (*D*), then, according to the proposition (4), we conclude that J_B is an optimal support for (*P*).

4. The Description of the Pivot Adaptive Method "PAM"

As the Simplex Method, to solve the problem (*P*), the adaptive method (and therefore also the pivot adaptive method) has two phases: the initialization phase, and the second phase, we will detail each one in the following.

1) The initialization phase:

In this phase an initial support feasible solution SF-solution $\{x, J_B\}$ will be determined, and thus the matrix Γ defined in (9).

We assume that $b_i \ge 0, \forall i \in I$, and $d_j^- = 0, \forall j \in J$.

Remark 1

Notice that each problem (*P*) can be reduced to this case, *i.e.*: $b_i \ge 0, \forall i \in I$, and $d_j^- = 0, \forall j \in J$, in fact:

a) if $\exists i^0 \in I / b_{i^0} < 0$ then both term of the corresponding constraint is multiplied by (−1).

b) if $\exists k \in J / d_k^- \ne 0$, then we do the change of variable: $(x')_k = x_k - d_k^-$, then:

$$0 \le (x')_k \le d_k^+ - d_k^- \tag{36}$$

and the general constraints $Ax = b$ of (*P*) will be write as:

$$a_1x_1 + a_1x_1 + \cdots + a_{k-1}x_{k-1} + a_k\left(x_k' + d_k^-\right) + a_{k+1}x_{k+1} + \cdots + a_nx_n = b \tag{37}$$

where $\forall j \in J$, a_j is the j^{th} column of the matrix *A*, from this relation we obtain:

$$a_1x_1 + a_1x_1 + \cdots + a_{k-1}x_{k-1} + a_kx_k' + a_{k+1}x_{k+1} + \cdots + a_nx_n = b - a_kd_k^- \tag{38}$$

in (36), if $\left(b - a_kd_k^-\right) < 0$ we multiple the two terms by (−1).

Then to determine an initial SF-solution to the problem (*P*), define the artificial problem of (*P*) as follows:

$$\left(\tilde{P}\right) \begin{cases} \max F\left(\tilde{x}\right) = -\sum_{j=1}^{j=m} \tilde{x}_j \\ Ax + \tilde{x} = b \\ d^- \le x \le d^+, 0 \le \tilde{x} \le b \end{cases} \tag{39}$$

where $b_i \ge 0, \forall i \in I$, and $\tilde{x} = \tilde{x}(J) = \left(\tilde{x}_i, i \in I\right) = \left(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_m\right)$ are the *m* artificial variables added to the problem (*P*).

We resolve the artificial problem $\tilde{P}$ with the "pivot adaptive method" starting with the obvious initial feasible solution: $(x, \tilde{x}), x = 0_{R^n}, \tilde{x} = b$, and the canonical support (determined by the indices of the artificial variables), and we take the matrix $\Gamma = A$.

If *H* is not empty, at the optimum of $\tilde{P}$ $\tilde{x} = 0$, $Ax = b$, and $d^- \le x \le d^+$, so x is a feasible solution of (*P*), and the optimal SF-solution of the problem $\tilde{P}$ can be taken as the initial SF-solution for the problem *P*.

2) The second phase:

Assume that an initial SF-solution $\{x, J_B\}$ of the problem (*P*) is found in the initialization phase, and let Γ the correspondent matrix computed by (9), $\epsilon \geq 0$ an arbitrary number, the problem will be solved with the pivot adaptive method that we describe in the following.

At the iterations of the pivot adaptive method the transfer $\{x, J_B\} \to \{\bar{x}, \bar{J}_B\}$ from one SF-solution to a new one is carried out in such a way that:

$$\beta(\bar{x}, \bar{J}_B) \leq \beta(x, J_B) \tag{40}$$

the transfer will be realized as two procedures making up the iteration:

i) The procedure of the feasible solution change $x \to \bar{x}$:

During this procedure we decrease the degree of non-optimality of the feasible solution: $\beta(\bar{x}) \leq \beta(x)$, and then improve the primal criterion $F(\bar{x}) \geq F(x)$:

i.1) compute the support gradient δ by (10), and the suboptimality estimate by (27).

i.2) If $\beta(x, J_B) \leq \varepsilon$, then stop the resolution with: x is an ϵ-optimal feasible solution for (*P*) (*P*).

i.3) If $\beta(x, J_B) > \varepsilon$, we compute the non-support components χ_N of the primal pseudo-feasible solution accompanying the support J_B by (12), and the search direction $l = (l_B, l_N)$ with:

$$\begin{cases} l_N = \chi_N - x_N \\ l_B = -\Gamma_N l_N \end{cases} \tag{41}$$

i.4) Find the primal step length θ^0 with:

$$\theta^0 = \min\{1, \theta_{j_0}\}, \text{where}: \theta_{j_0} = \min_{j \in J_B} \theta_j, \theta_j = \begin{cases} \dfrac{d_j^+ - x_j}{l_j}, & \text{if } l_j > 0; \\ \dfrac{d_j^- - x_j}{l_j}, & \text{if } l_j < 0; \\ \infty; & \text{if } l_j = 0. \end{cases} \quad (j \in J_B) \tag{42}$$

Let j_{00} such that: $J_B(j_{00}) = j_0$ which is the position of j_0 in the vector of indices J_B.

i.5) Compute the new feasible solution: $\bar{x} = x + \theta^0 l$, and $F(\bar{x}) = F(x) + \theta^0 \beta(x, J_B)$.

i.6) If $\theta^0 = 1$, then stop the resolution with: $\{\bar{x}, J_B\}$ is the optimal SF-solution of (*P*).

i.7) If $\theta^0 < 1$, then compute:

$$\beta(\bar{x}, J_B) = (1 - \theta^0)\beta(x, J_B) \tag{43}$$

i.8) If $\beta(\bar{x}, J_B) \leq \epsilon$, then stop the resolution with: $\{\bar{x}, J_B\}$ is the ϵ-optimal SF-solution of (*P*).

i.9) if $\beta(\bar{x}, J_B) > \epsilon$, then go to ii) to change the support J_B to $\bar{J}_B$.

ii) The procedure of the support change $J_B \to \overline{J}_B$:

During this procedure we decrease the degree of non-optimality of the support: $\beta\left(\overline{J}_B\right) \le \beta\left(J_B\right)$, and then improve the dual criterion $\phi\left(\overline{\lambda}\right) \ge \phi\left(\lambda\right)$, where $\overline{\lambda}$ is the dual feasible solution accompanying the support $\overline{J}_B$, and λ is the dual feasible solution accompanying the support J_B.

In this procedure there are two rules to change support: rule of "the short step", and rule of "the long step", the two rules will be presented.

ii.1) to the j_0 found in (i.4), compute $\chi_{j_0} = x_{j_0} + l_{j_0}$, and $\alpha_0 = \chi_{j_0} - \overline{x}_{j_0}$.

ii.2) compute the dual direction t with:

$$\begin{cases} t_{j_0} = -sign\left(\alpha_0\right); \\ t_j = 0,\ j \in J_B \setminus \left\{j_0\right\}; \\ t_N^{\mathrm{T}} = t_B^{\mathrm{T}} \Gamma_N. \end{cases} \tag{44}$$

ii.3) compute:

$$\sigma_j = \begin{cases} \dfrac{\delta_j}{t_j}, & \text{if } \delta_j t_j > 0; \\ 0, & \text{if } \delta_j = 0, t_j < 0, \xi_j = d_j^-; \ \left(j \in J_N\right) \\ 0, & \text{if } \delta_j = 0, t_j > 0, \xi_j = d_j^+; \\ \infty, & \text{in other cases.} \end{cases} \tag{45}$$

and arrange the obtained values in increasing order as:

$$\sigma_{j_1} \le \sigma_{j_2} \le \cdots \le \sigma_{j_p},\ j_k \in J_N, \sigma_{j_k} \ne \infty, k \in \{1, \cdots, p\}. \tag{46}$$

As was said before, the change of support can be done with two different rules, to use the "short step rule" go to a), and to use the "long step rule" go to b).

a) Change of the support with the "short step rule":

a.1) From (46) put:

$$\sigma^0 = \sigma_{j_1} = \sigma_{j^*} = \min_{j \in J_N} \sigma_j. \tag{47}$$

Let j_{**} such that: $J_N\left(j_{**}\right) = j^*$ which is the position of j^* in the vector of indices J_N.

Put: $\overline{J}_B = \left(J_B \setminus \{j_0\}\right) \cup \left\{j^*\right\}$, $\overline{J}_N = \left(J_N \setminus \left\{j^*\right\}\right) \cup \left\{j_0\right\}$, then $\overline{J}_B\left(j_{00}\right) = j^*$, $\overline{J}_N\left(j_{**}\right) = j_0$.

a.2) Compute:

$$\begin{gathered} \overline{\delta} = \delta - \sigma^0 t, \\ \beta\left(\overline{x}, \overline{J}_B\right) = \beta\left(\overline{x}, J_B\right) + \sigma^0 \alpha_0 \end{gathered} \tag{48}$$

go to ii.4).

b) Change of the support with the "long step rule":

b.1) for every $j_k, k \in \{1, \cdots, p\}$ of (47) compute

$$\Delta\alpha_{j_k} = \left|\Gamma_{j_{00} j_k}\right| \left(d_{j_k}^+ - d_{j_k}^-\right) \tag{49}$$

b.2) as j^* we choose j_q such that

$$\alpha_{j_{q-1}} = \left[\alpha_0 + \sum_{k=1}^{k=q-1} \Delta\alpha_{j_k}\right] < 0, \alpha_{j_q} = \left[\alpha_0 + \sum_{k=1}^{k=q} \Delta\alpha_{j_k}\right] \geq 0 \tag{50}$$

Let j_{**} such that: $J_N\left(j_{**}\right) = j^*$ which is the position of j_0 in the vector of indices J_B.

b.3) put: $\sigma^0 = \sigma_{j^*}$, and $\overline{J}_B = \left(J_B \setminus \{j_0\}\right) \cup \left\{j^*\right\}$, $\overline{J}_N = \left(J_N \setminus \left\{j^*\right\}\right) \cup \left\{j_0\right\}$, then $\overline{J}_B\left(j_{00}\right) = j^*$, $\overline{J}_N\left(j_{**}\right) = j_0$.

b.4) Compute:

- $\overline{\delta} = \delta - \sigma^0 t$.
- $\beta\left(\overline{x}, \overline{J}_B\right) = \beta\left(\overline{x}, J_B\right) + \sum_{k=1}^{k=q} \alpha_{j_{k-1}}\left(\sigma_{j_k} - \sigma_{j_{k-1}}\right)$ where: $\alpha_{j_0} = \alpha_0$, $\sigma_{j_0} = 0$.

Go to ii.4). ii.4) compute $\overline{\Gamma}$ as follows:

$$\begin{cases} \overline{\Gamma}_{j_{00}j} = \dfrac{\Gamma_{j_{00}j}}{\Gamma_{j_{00}j^*}}, \ j \in J; \\ \overline{\Gamma}_{ij} = \Gamma_{ij} - \Gamma_{ij^*}\overline{\Gamma}_{j_{00}j}, \ i \in I \setminus \left\{j_{00}\right\}, j \in J. \end{cases} \tag{51}$$

$\Gamma_{j_{00}j^*}$ is "the pivot". Go to ii.5).

ii.5) Set: $x = \overline{x}$, $J_B = \overline{J}_B$, $\beta\left(x, J_B\right) = \beta\left(\overline{x}, \overline{J}_B\right)$, $\delta = \overline{\delta}$, and $\Gamma = \overline{\Gamma}$. Go to i.2).

5. The Pivot Adaptive Method

Let $\left\{x, J_B\right\}$ be an initial SF-solution for the problem (*P*), Γ the matrix computed by (9), and $\epsilon \geq 0$ are given.

The "Pivot Adaptive Method" (rule of the "short step" and rule of "the long step" are presented) is summarized in the following steps:

Algorithm **1 (The Pivot Adaptive Method)**

1) compute the support gradient δ^{T} by (10), and $\beta\left(x, J_B\right)$ by (27).

2) if $\beta\left(x, J_B\right) \leq \epsilon$, then STOP with: *x* is an ϵ-optimal feasible solution for (*P*).

3) if $\beta\left(x, J_B\right) > \epsilon$, then compute the non-support components χ_N of the primal pseudo-feasible solution accompanying the support J_B by (12), and the search direction *l* with (41).

4) find the primal step length θ^0 with (42). Let j_{00} such that: $J_B\left(j_{00}\right) = j_0$ which is the position of j_0 in the vector of indices J_B.

5) Compute the new feasible solution: $\overline{x} = x + \theta^0 l$, and $F\left(\overline{x}\right) = F\left(x\right) + \theta^0 \beta\left(x, J_B\right)$.

6) if $\theta^0 = 1$, then STOP with: $\left\{\overline{x}, J_B\right\}$ is the optimal SF-solution for (*P*).

7) if $\theta^0 < 1$, then compute $\beta\left(\overline{x}, J_B\right) = \left(1 - \theta^0\right)\beta\left(x, J_B\right)$.

8) if $\beta\left(\overline{x}, J_B\right) \leq \epsilon$, then STOP with: $\left\{\overline{x}, J_B\right\}$ is the ϵ-optimal SF-solution for (*P*).

9) if $\beta\left(\overline{x}, J_B\right) > \epsilon$, then change the support J_B to $\overline{J}_B$.

10) compute $\chi_{j_0} = x_{j_0} + l_{j_0}$, and: $\alpha_0 = \chi_{j_0} - \overline{x}_{j_0}$.

11) compute the dual direction *t* with (44).

12) compute σ_j, for $j \in J_N$ with (45), and arrange the obtained values in increasing order.

13) do the change of support by one of the rules:

a) Change of the support with the "short step rule" as follows:

a.1) compute σ^0 with (47), and set: $\bar{J}_B = \left(J_B \setminus \{j_0\}\right) \cup \{j^*\}$, $\bar{J}_N = \left(J_N \setminus \{j^*\}\right) \cup \{j_0\}$.

a.2) compute $\beta\left(\bar{x}, \bar{J}_B\right) = \beta\left(\bar{x}, J_B\right) + \sigma^0 \alpha_0$. Go to (14).

b) Change of the support with the "long step rule" as follows:

b.1) for every $j_k, k \in \{1, \cdots, p\}$ of (46) compute $\Delta\alpha_{j_k} = \left|\Gamma_{j_0 0 j_k}\right| \left(d^+_{j_k} - d^-_{j_k}\right)$.

b.2) as j^* we choose j_q such that (50) is verified.

b.3) set: $\sigma^0 = \sigma_{j^*}$, and $\bar{J}_B = \left(J_B \setminus \{j_0\}\right) \cup \{j^*\}$, $\bar{J}_N = \left(J_N \setminus \{j^*\}\right) \cup \{j_0\}$.

b.4) compute $\beta\left(\bar{x}, \bar{J}_B\right) = \beta\left(\bar{x}, J_B\right) + \sum_{k=1}^{k=q} \alpha_{j_{k-1}} \left(\sigma_{j_k} - \sigma_{j_{k-1}}\right)$ where: $\alpha_{j_0} = \alpha_0$, $\sigma_{j_0} = 0$. Go to (14).

14) compute $\bar{\Gamma}$ with (51) and $\bar{\delta} = \delta - \sigma^0 t$.

15) set: $x = \bar{x}$, $J_B = \bar{J}_B$, $\beta(x, J_B) = \beta\left(\bar{x}, \bar{J}_B\right)$, $\delta = \bar{\delta}$, and $\Gamma = \bar{\Gamma}$. Go to (2).

6. Example

In this section, a linear problem will be resolved with the Pivot Adaptive Method using the "short step rule", and in parallel we will explain how to realize these calculations under the shape of successive tables as is given in Figure 1.

Consider the following linear problem with bounded variables:

$$(P) \quad \begin{cases} \max F(x) = c^{\mathrm{T}} x \\ Ax = b \\ d^- \le x \le d^+. \end{cases} \tag{52}$$

where: $c^{\mathrm{T}} = (65, 115, 0, 0, 0)$, $x = \left(x_1, x_2, x_3, x_4, x_5\right)^{\mathrm{T}}$,

$$A = \begin{pmatrix} 5/2 & 15/2 & 1 & 0 & 0 \\ 1/8 & 1/8 & 0 & 1 & 0 \\ 35/2 & 10 & 0 & 0 & 1 \end{pmatrix}, \quad b = (240, 5, 595)^{\mathrm{T}}, \quad d^- = 0_{R^5},$$

$d^+ = (34, 34, 240, 5, 595)^{\mathrm{T}}$.

Put $J = \{1, 2, 3, 4, 5\}$, $\epsilon = 10^{-3}$.

First phase (Initialization):

Let $\{x^1, J_B^1\}$ be the initial SF-solution of the problem (*P*) where; $x^1 = (11, 27, 10, 1/4, 265/2)^{\mathrm{T}}$, $J_B^1 = \{3, 4, 5\}$, so: $J_N^1 = \{1, 2\}$, $A_B^1 = \left(a_3, a_4, a_5\right) = I_3$, $A_N^1 = \left(a_1, a_2\right)$ then The matrix Γ is given by: $\Gamma = \left(\Gamma_B, \Gamma_N\right)$ where: $\left(\Gamma_1\right)_B = I_3$,

$$\left(\Gamma_1\right)_N = \begin{pmatrix} 5/2 & 15/2 \\ 1/8 & 1/8 \\ 35/2 & 10 \end{pmatrix}.$$

Note that for the initial support we have chosen the canonical support, and for the initial feasible solution we took an interior point of the feasible region of the problem (*P*).

- In the preamble part of the tables (Figure 1) which has four rows we put sequentially: c^{T}, $\left(d^-\right)^{\mathrm{T}}$, $\left(d^+\right)^{\mathrm{T}}$, x^1. The tables have 8 columns.

Second phase:

Starting with the SF-solution $\{x^1, J_B^1\}$, and the matrix Γ to resolve the problem (*P*) with the PAM using the short step rule.

◆ In the first table of PAM (**Figure 1**) (represents the first iteration) there are 8 rows, in the first 3 rows we have the matrix Γ (indicating the vector formed A_B^1 in the first column of the tables), and in the rest rows we have sequentially: δ, l, θ, x^2, σ.

The support gradient $\delta_1^{\mathrm{T}} = \left(\left(\delta_1^{\mathrm{T}}\right)_B, \left(\delta_1^{\mathrm{T}}\right)_N\right)$ where: $\left(\delta_1^{\mathrm{T}}\right)_B = 0_{R^3}$, $\left(\delta_1^{\mathrm{T}}\right)_N = (65, 115)$, and $\beta\left(x^1, J_B^1\right) = 2300 > \epsilon$.

1) Iteration 1:

- Change solution: we have: $(\chi_1)_N = (34, 34)$, and $(l_1)_N = (23, 7)$, $(l_1)_B = (-110, -15/4, -945/2)$, $\theta^0 = \theta_4 = 1/15$, $\left(J_B^1\right)(2) = 4$ (4 is the second component of the vector of index J_B^1).

◆ The case of θ^0 is marked by yellow in tables (**Figure 1**), it correspond at the second vector of J_B^1 which is a_4 (which will come out of the basis.

Then, $\theta^0 < 1$, the new feasible solution $x^2 = (188/15, 412/15, 40/15, 0, 101)$, and $\beta\left(x^2, J_B^1\right) = 6440/3 > \epsilon$.

- Change support (with the short step rule):

$\chi_4 = -7/2$, and: $\alpha_0 = -7/2$, the dual direction $t^{\mathrm{T}} = (1/8, 1/8, 0, 1, 0)$, $\sigma^0 = \sigma_1 = 520$, then $\left(J_N^1\right)(1) = 1$ (1 is the first component of the vector of index J_N^1).

◆ The case of σ^0 is marked by green in tables (**Figure 1**), his row correspond to vector a_1 which will go into the basis.

So $J_B^2 = \left(J_B^1 \setminus \{4\}\right) \cup \{1\} = \{3, 1, 5\}$, $J_N^2 = \left(J_N^1 \setminus \{1\}\right) \cup \{4\} = \{4, 2\}$.

		Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	β	
	c c_B	65	115	0	0	0		Preamble
d^-		0	0	0	0	0		
d^+		34	34	240	5	595		
x^1		11	27	10	1/4	265/2		
a_3	0	5/2	15/2	1	0	0		1st table
a_4	0	1/8	1/8	0	1	0		
a_5	0	35/2	10	0	0	1		
δ		65	115	0	0	0	2300	
l		23	7	-110	-15/4	-945/2		
θ		1	1	1/11	1/15	53/189		
x^2		188/15	412/15	40/15	0	101	6440/3	
σ		520	920				980/3	
a_3	0	0	5	1	-20	0		2nd table
a_1	65	1	1	0	8	0		
a_5	0	0	-15/2	0	-140	1		
δ		0	50	0	-520	0		
l		-98/15	98/15	-98/3	0	49		
θ		94/49	1	4/49	1	494/49		
x^3		12	28	0	0	105	300	
σ			10		26		0	

Figure 1. Tables of the pivot adaptive method.

- For have the 2nd table (**Figure 1**), we applied the pivoting rule where the pivot is marked by red. The new support gradient $\delta_2^{\mathrm{T}} = \delta_1^{\mathrm{T}} - \sigma^0 t = \left(\left(\delta_2^{\mathrm{T}}\right)_B, \left(\delta_2^{\mathrm{T}}\right)_N\right)$ where: $\left(\delta_2^{\mathrm{T}}\right)_B = 0_{R^3}$, $\left(\delta_2^{\mathrm{T}}\right)_N = (-520, 50)$, and $\beta\left(x^2, J_B^2\right) = \beta\left(x^2, J_B^1\right) + \sigma^0 \alpha_0 = 980/3$.

  As $\beta\left(x^2, J_B^2\right) > \epsilon$, we go to another iteration, we compute then: $\Gamma_2 = \left((\Gamma_2)_B, (\Gamma_2)_N\right)$ where:

$$(\Gamma_2)_B = I_3, \ (\Gamma_1)_N = \begin{pmatrix} -20 & 5 \\ 8 & 1 \\ -140 & -15/2 \end{pmatrix}.$$

2) Iteration 2:

- Change solution: we have: $\chi_N = (0, 34)$, and $(l_2)_N = (0, 98/15)$, $(l_2)_B = (-98/3, -98/15, 49)$, $\theta^0 = \theta_3 = 4/49$, $\left(J_B^2\right)(1) = 3$ (3 is the first component of the vector of indice J_B^2).

  Then, $\theta^0 < 1$, the new feasible solution $x^3 = (12, 28, 0, 0, 105)$, and $\beta\left(x^2, J_B^1\right) = 300 > \epsilon$.
- Change support (with the short step rule): $\chi_3 = -30$, and: $\alpha_0 = -30$, the dual direction $t^{\mathrm{T}} = (0, 5, 1, -20, 0)$, $\sigma^0 = \sigma_2 = 10$, then $\left(J_N^2\right)(2) = 2$ (2 is the second component of the vector of index J_N^2).

  So $J_B^3 = \left(J_B^2 \setminus \{3\}\right) \cup \{2\} = \{2, 1, 5\}$, $J_N^3 = \left(J_N^2 \setminus \{2\}\right) \cup \{3\} = \{4, 3\}$.

  The new support gradient $\delta_3^{\mathrm{T}} = \delta_2^{\mathrm{T}} - \sigma^0 t = \left(\left(\delta_3^{\mathrm{T}}\right)_B, \left(\delta_3^{\mathrm{T}}\right)_N\right)$ where:

  $\left(\delta_3^{\mathrm{T}}\right)_B = 0_{R^3}$, $\left(\delta_3^{\mathrm{T}}\right)_N = (10, 320)$, and $\beta\left(x^3, J_B^3\right) = \beta\left(x^3, J_B^2\right) + \sigma^0 \alpha_0 = 0$.

  Then $\left\{x^3, J_B^3\right\}$ is the optimal SF-solution of the problem (*P*), and $F\left(x^3\right) = 4000$.

7. Brief Numerical Comparison between the PAM and the Simplex Method

To realize a numerical comparison between the Simplex algorithm implemented in the function "linprog" under the MATLAB programming language version 7.14, 0.739 (R2012a) and our algorithm (PAM), an implementation for the later with the short step rule, has been developed.

The comparison is carried on the resolution of the Klee-Minty problem which has the following form [7]:

$$(P_3) \begin{cases} \max\ 2^{n-1} x_1 + 2^{n-2} x_2 + \ldots + 2x_{n-1} + x_n \\ x_1 \le 5 \\ 4x_1 + x_2 \le 25 \\ 8x_1 + 4x_2 + x_3 \le 125 \\ \quad \vdots \\ 2^n x_1 + 2^{n-1} x_2 + \cdots + 4x_{n-1} + x_n \le 5^n \\ x \ge 0 \end{cases} \qquad (53)$$

where $x = (x_1, x_2, \cdots, x_n)$. (P_3) has n variables, n constraints and 2^n vertices.

For a fixed n, to write (P_3) in the form of the problem (P) given in (1), we added to each of the n constraints of (P_3) a spread variable, and we found, for each of the $2n$ variables of the result problem, a lower born and an upper born. Then the problem (P_3) can be done as a linear problem with bounded variables as follows:

$$(P_4)\quad \begin{cases} \max\ 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2x_{n-1} + x_n \\ x_1 + x_{n+1} = 5 \\ 4x_1 + x_2 + x_{n+2} = 25 \\ 8x_1 + 4x_2 + x_3 + x_{n+3} = 125 \\ \quad\vdots \\ 2^n x_1 + 2^{n-1}x_2 + \cdots + 4x_{n-1} + x_n + x_{2n} = 5^n \\ 0 \le x_1 \le 5;\ 0 \le x_2 \le 25;\cdots;0 \le x_n \le 5^n; \\ 0 \le x_{n+1} \le 5;\ 0 \le x_{n+2} \le 25;\cdots;0 \le x_{2n} \le 5^n. \end{cases} \tag{54}$$

The Simplex algorithm chosen here takes as a starting solution the original, then in our implementation of the (PAM) we impose the same initial feasible solution. For the initial support we had taken the canonical one $J_B = \{n+1, n+2, \cdots, 2n\}$, then $\Gamma = A$.

We have considered problems with matrix A of size $n \times n$, where $n \in \{3,5,7,10,12,15,17,20,23,25,27,30,33,35,37,40,42,45,47,50\}$, for each size, we obtained the number of iterations, and the time of resolution, where the time is given into milliseconds.

This results are reported in **Figure 2**, and in **Figure 3**, where "Simplex(it)", "PAM(it)", "Simplex(tm)", "PAM(tm)" represent respectively the number of iterations of the Simplex algorithm, the number of iterations of the Pivot Adaptive Method, the time of the Simplex algorithm, the time of the Pivot Adaptive Method. The time is given into milliseconds.

Size	Simplex (it)	PAM (it)	Simplex (tm)	PAM (tm)	Size	Simplex (it)	PAM (it)	Simplex (tm)	PAM (tm)
3 × 3	7	3	8.1	0.65	27 × 27	461	8	325.4	4.8
5 × 5	15	3	15.2	0.65	30 × 30	647	3	467.4	0.83
7 × 7	27	3	22.1	0.66	33 × 33	867	24	654	17.6
10 × 10	51	3	36.6	0.66	35 × 35	1041	9	769.8	12.4
12 × 12	69	3	47.8	0.71	37 × 37	1258	18	888.7	12.6
15 × 15	103	5	75.6	2	40 × 40	1560	81	1095.8	77.9
17 × 17	142	5	101.9	2	42 × 42	1721	367	1236.6	335
20 × 20	204	10	147.1	5.5	45 × 45	1851	825	1346.7	734
23 × 23	291	5	209.7	2.1	47 × 47	1993	1354	1442.1	1307
25 × 25	371	5	268.8	2.2	50 × 50	2169	1185	1561	1211.1

Figure 2. Number of iterations and time for Linprog-Simplex and PAM for Klee-Minty problem.

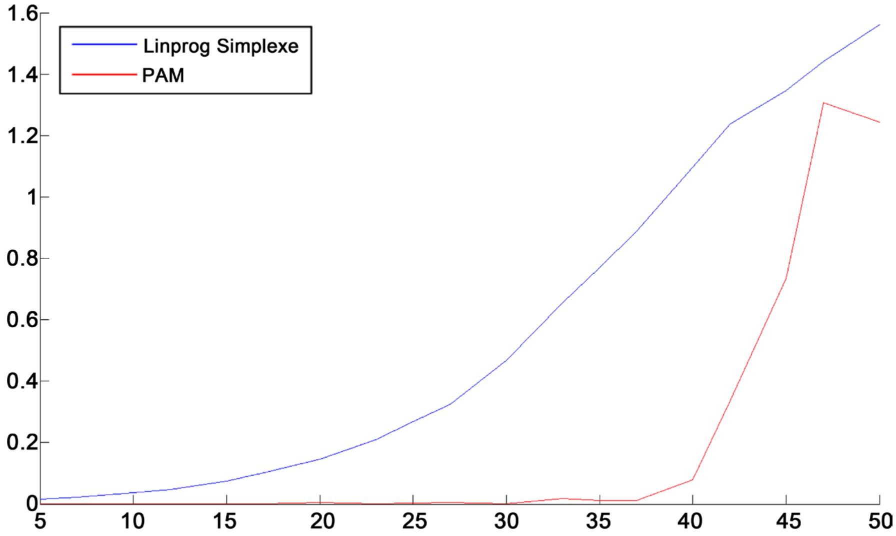

Figure 3. Time comparison between Linprog-Simplex and PAM for the Klee-Minty problem.

We remark that the (PAM) is better than the Simplex algorithm in computation time, and in necessary number of iterations to solve the problems. Recall that the optimal solution of the problem (P_3) is given by $x = \left(0, 0, \cdots, 5^n\right)$.

8. Conclusions

The main contribution of this article is a new variant of the Adaptive Method that we have called "Pivot Adaptive Method" (PAM). In this variant, we use the simplex pivoting rule in order to avoid computing the inverse of the basic matrix in each iteration. As was recognized in this work, the algorithm saves our time compared to the original algorithm (AM), and allows us to present the resolution of a given problem under the shape of successive tables as seen in an example.

We have implemented our algorithm (PAM) ((PAM) with the short step rule, and (PAM) with the long step rule) using MATLAB, and we have done a brief comparison with the primal simplex algorithm (using linprog-simplex in MATLAB) for solving the Klee-Minty problem. Indeed, the (PAM) is more efficient in number of iterations, and in computation time.

In a subsequent work, we shall do a more thorough comparison between the Simplex algorithm of Dantzig and the (PAM), and we shall extend the (PAM) to the problems with a large scale using the constraint selection technique.

References

[1] Dantzig, G.B. (1951) Minimization of a Linear Function of Variables Subject to Linear Inequalities. John Wiley, New York.

[2] Minoux, M. (1983) Programmation mathématique, Théorie et al gorithmes, Tome 1, Dunod.

[3] Culioli, J.C. (1994) Introduction à l'optimisation, Elipse.

[4] Dantzig, G.B. (1963) Linear Programming and Extensions. Princeton University Press, Princeton. https://doi.org/10.1515/9781400884179

[5] Gabasov, R. and Kirillova, F.M. (1977, 1978, 1980) Method of Linear Programming. Vol. 1, 2 and 3, BGU Press, Minsk. (In Russian)

[6] Bland, R.G. (1977) New Finite Pivoting Rules for the Simplex Method. *Mathematics of Operations Research*, **2**, 103-107.

[7] Klee, V. and Minty, G.J. (1972) How Good Is the Simplex Algorithm? In: Shisha III, O., Ed., *Inequalities*, Academic Press, New York, 159-175.

[8] Khachiyan, L.G. (1979) A Polynomial Algorithm in Linear Programming. *Soviet Mathematics Doklady*, **20**, 191-194.

[9] Karmarkar, N.K. (1984) A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, **4**, 373-395.

[10] Kojima, M., Megiddo, N., Noma, T. and Yoshise, A. (1989) A Unified Approach to Interior Point Algorithms for Linear Programming. In: Megiddo, N., Ed., *Progress in Mathematical Programming: Interior Point and Related Methods*, Springer Verlag, New York, 29-47.

[11] Ye, Y. (1997) Interior Point Algorithms, Theory and Analysis. John Wiley and Sons, Chichester.

[12] Vanderbei, R.J. and Shanno, D.F. (1999) An Interior-Point Algorithm for Nonconvex Nonlinear Programming. *Computational Optimization and Applications*, **13**, 231-252.

[13] Peng, J., Roos, C. and Terlaky, T. (2001) A New and Efficient Large-Update Interior-Point Method for Linear Optimization. *Journal of Computational Technologies*, **6**, 61Â-80.

[14] Cafieri, S., Dapuzzo, M., Marino, M., Mucherino, A. and Toraldo, G. (2006) Interior Point Solver for Large-Scale Quadratic Programming Problems with Bound Constraints. *Journal of Optimization Theory and Applications*, **129**, 55-75.

[15] Gabasov, R.F. (1994) Adaptive Method for Solving Linear Programming Problems. University of Bruxelles, Bruxelles.

[16] Gabasov, R.F. (1994) Adaptive Method for Solving Linear Programming Problems. University of Karlsruhe, Institute of Statistics and Mathematics, Karlsruhe.

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

[18] Gabasov, R., Kirillova, F.M. and Kostyukova, O.I. (1979) A Method of Solving General Linear Programming Problems. *Doklady AN BSSR*, **23**, 197-200. (In Russian)

[19] Kostina, E. (2002) The Long Step Rule in the Bounded-Variable Dual Simplex Method: Numerical Experiments. *Mathematical Methods of Operation Research*, **55**, 413-429.

[20] Radjef, S. and Bibi, M.O. (2011) An Effective Generalization of the Direct Support Method. *Mathematical Problems in Engineering*, **2011**, Article ID: 374390.

[21] Balashevich, N.V., Gabasov, R. and Kirillova, F.M. (2000) Numerical Methods of Program and Positional Optimization of the Linear Control Systems. *Zhurnal Vychislitel̉ noi Matematiki i Matematicheskoi Fiziki*, **40**, 838-859.

[22] Bentobache, M. and Bibi, M.O. (2012) A Two-Phase Support Method for Solving Linear Programs: Numerical Experiments. *Mathematical Problems in Engineering*, **2012**, Article ID: 482193.

[23] Bibi, M.O. and Bentobache, M. (2015) A Hybrid Direction Algorithm for Solving Linear Programs. *International Journal of Computer Mathematics*, **92**, 201-216.