STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, March 2025, pp 922–948.
Published online in International Academic Press (www.IAPress.org)

A Hybrid Direction Method for Linear Fractional Programming

Mohammed Amin Hakmi [1,2,3,*], Mohand Bentobache [1,2], Mohand Ouamer Bibi [1]

[1] *University of Bejaia, Faculty of Exact Sciences, LaMOS Research Unit (Modeling and Optimization of Systems), 06000 Bejaia, Algeria*
[2] *Laboratory of Pure and Applied Mathematics, University of Laghouat, 03000, Algeria*
[3] *Department of Mathematics and Computer Science, University Center Nour Bachir, El-Bayadh, Algeria*

Abstract In this article, we propose a new method for solving Linear Fractional Programming (LFP) problems with bounded variables. The proposed algorithm passes from a support feasible solution to a better one following the feasible direction proposed in [K. Djeloud, M. Bentobache and M. O. Bibi, A new method with hybrid direction for linear programming, Concurrency and Computation, Practice and Experience 33 (1), 2021]. Optimality and suboptimality criteria which allow to stop the algorithm when an optimal or suboptimal solution is achieved were stated and proved. Then, a new method called a Hybrid Direction Method (HDM) is described and a numerical example is given for illustration purpose. In order to compare our method to the classical approaches, we develop an implementation with the Matlab programming language. The obtained numerical results on solving 120 randomly generated LFP test problems show that HDM with long step rule is competitive with the primal simplex method and the interior-points method implemented in Matlab.

Keywords Linear Fractional Programming, Simplex Method, Support Method, Hybrid Direction, Bounded Variables, Numerical Experiments.

AMS 2010 subject classifications 90C32, 65K05.

DOI: 10.19139/soic-2310-5070-2245

1. Introduction

Linear fractional programming is a very important subdiscipline of optimization. It consists in optimizing (maximizing or minimizing) an objective function which is a ratio of two linear functions, subject to linear constraints on the decision variables. These variables can be nonnegative, bounded, continuous, integer or mixed-integer. Many practical problems which arise on several fields can be modeled as LFP problems. We can cite resource allocation, machine learning, optimal cutting stock problems, blending problems, minimum-risk problems in stochastic programming, production efficiency problems, data envelopment analysis (DEA) [40, 49, 3, 19, 50, 41, 21].

The theoretical and practical importance of LFP lead many researchers to develop a variety of numerical methods for solving the problem. Among the existing approaches, we can cite the method of Charnes and Cooper (1962) [18]. These authors have shown that the original LFP problem can be transformed to a Linear Programming (LP) problem by introducing additional variables and constraints. This approach allows to exploit all the LP theoretical and numerical results, in particular the obtained problem can be solved by the simplex method [20], interior-points methods [52], the support or adaptive method, the hybrid direction methods [25, 28, 12, 13, 7, 22], etc.

The simplex algorithm was initially developed by Dantzig in 1947 [20] for solving LP problems. Due to its practical efficiency, it was generalized for solving many optimization problems, such as convex quadratic

*Correspondence to: Mohammed Amin Hakmi (Email: mohamedamin.hakmi@univ-bejaia.dz). LaMOS Research Unit (Modeling and Optimization of Systems), Operational Research Department, University of Bejaia, 06000, Bejaia, Algeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

programming problems [53] and concave quadratic minimization problems [8]. In the sixties, the simplex method was generalized by Martos for solving LFP problems written in their original form [39, 40, 4]. Later, many methods were developed for LFP, such as the dual simplex method [4], the parametric approach of Dinkelbach [23], which consists in reducing the resolution of an LFP problem to the resolution of a sequence of LP problems, the interior-points method called "method of analytic centers" [42], the criss-cross method [33]. There exist also other methods for solving LFP problems, see [43, 48, 51].

In 1977, Gabasov and Kirillova developed the support method for solving LP problems [25]. Contrarily to the simplex method, this latter can use basic or nonbasic feasible solutions which correspond to extreme, boundary or interior points. Later, Gabasov and Kirillova developed the adaptive method which uses an improvement direction depending on the current solution. This method is then applied for the resolution of a variety of optimization problems: convex and nonconvex quadratic programming [1, 17, 27, 36, 45, 32], multiobjective programming [46, 35], optimal control [2, 10, 11, 28, 29, 34], integer and mixed-integer programming [15, 16, 47], etc.

Recently, several algorithms based on the concept of hybrid direction were proposed for solving LP problems [7, 9, 13, 22, 30], linear optimal control problems [54], convex quadratic problems [14, 38]. In this article, we generalize the hybrid direction method proposed in [22], for the resolution of LFP problems with bounded variables. We state and prove optimality and suboptimality criteria, then we describe a complete algorithm which handles LFP problems as they are presented in their original form without transforming them into LP problems and we illustrate it with a numerical example. In order to compare our method to the primal simplex method and the interior-points method implemented in Matlab, we develop an implementation with the Matlab programming language and finally we present some numerical results on 120 randomly generated LFP test problems with n constraints and $2n$ bounded variables, where n is varying from 100 to 1400. The numerical study carried out shows the efficiency of our method and its superiority over the primal simplex algorithm and the interior-points method of Matlab, particularly when it uses the long step rule for changing the current support (basis).

This paper is organized as follows: in Section 2, we give some notations and definitions. In Section 3, we state and prove the optimality and suboptimality criteria for LFP, which will be used to stop the proposed algorithm, then we present the primal support method for solving LFP problems with bounded variables written in standard form. In the fourth section, we present the proposed hybrid direction method for solving the considered problem. Section 5 is devoted to the presentation of some numerical results obtained for 120 randomly generated LFP test problems. Finally, we conclude the paper and give some future works.

2. Statement of the problem and definitions

The Linear Fractional Programming problem with Bounded Variables (LFPBV) is presented in the following standard form:

$$\max\ F(x) = \frac{P(x)}{Q(x)} = \frac{p^T x + p_0}{q^T x + q_0}, \tag{1}$$

$$\text{subject to } Ax = b,\ l \leq x \leq u, \tag{2}$$

where p, q, x, l and u are n-vectors, with $\|l\| < \infty$, $\|u\| < \infty$; b is an m-vector; A is a matrix of dimension $(m \times n)$, with $rank(A) = m < n$, p_0 and q_0 are two real numbers. We suppose that $Q(x) > 0$ for all x verifying the constraints of the LFP problem (1)-(2).

• Let us define the following index sets:

$$I = \{1, 2, \ldots, m\},\ J = \{1, 2, \ldots, n\},\ J = J_B \cup J_N,\ J_B \cap J_N = \emptyset,\ |J_B| = m.$$

So we can partition the different vectors and the matrix A as follows:

$$l = l\,(J) = (l_j, j \in J)\,,\quad u = u\,(J) = (u_j, j \in J)\,,$$

$$x = x\,(J) = (x_j, j \in J) = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, x_B = x\,(J_B) = (x_j, j \in J_B), x_N = x\,(J_N) = (x_j, j \in J_N),$$

$$p = (p_j,\ j \in J) = \begin{pmatrix} p_B \\ p_N \end{pmatrix}, q = (q_j,\ j \in J) = \begin{pmatrix} q_B \\ q_N \end{pmatrix},$$

$$A = A\,(I,J) = (a_{ij}, i \in I, j \in J) = (a_j, j \in J),\, a_j = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix},$$
$$A = (A_B, A_N)\,,\; A_B = A\,(I, J_B),\, A_N = A\,(I, J_N).$$

• A vector x verifying the constraints (2) is called a feasible solution (FS) of the problem (1)-(2). We suppose in the following that the feasible set $S = \{x \in \mathbb{R}^n : Ax = b,\; l \leq x \leq u\}$ is nonempty.
• Define the real number $\alpha > 0$ as follows: $\alpha = \min\limits_{x \in S} Q(x)$.
• A feasible solution x^* is called optimal if $F(x^*) \geq F(x),\; \forall x \in S$.
• A feasible solution x^ϵ is called ϵ-optimal or suboptimal if $F(x^*) - F(x^\epsilon) \leq \epsilon$,
where x^* is an optimal solution for the problem (1)-(2) and ϵ is a nonnegative number chosen in advance.
• The index set J_B is called a support if $\det A_B = \det A(I, J_B) \neq 0$.
• The pair $\{x, J_B\}$ formed with the FS x and the support J_B is called a Support Feasible Solution (SFS) and it is called nondegenerate if $l_j < x_j < u_j,\; j \in J_B$.
• Let us define the multipliers vectors π_P^T and π_Q^T:

$$\pi_p^T = p_B^T A_B^{-1}, \pi_Q^T = q_B^T A_B^{-1}, \tag{3}$$

and the reduced costs vectors:

$$\Delta' = \begin{pmatrix} \Delta'_B \\ \Delta'_N \end{pmatrix},\; \Delta'' = \begin{pmatrix} \Delta''_B \\ \Delta''_N \end{pmatrix},\; \Delta(x) = \begin{pmatrix} \Delta_B(x) \\ \Delta_N(x) \end{pmatrix},$$

where

$$\Delta'_N = A_N^T \pi_P - p_N,\; \Delta''_N = A_N^T \pi_Q - q_N,\; \Delta_N(x) = \Delta'_N - F(x)\Delta''_N, \tag{4}$$

and

$$\Delta'_B = A_B^T \pi_P - p_B = A_B^T (A_B^{-1})^T p_B - p_B = 0,\; \Delta''_B = 0,\; \Delta_B^T(x) = 0.$$

3. Optimality and suboptimality criteria

3.1. Increment formula of the objective function

Let $\{x, J_B\}$ be an SFS of the problem (1)-(2). We consider an other arbitrary FS $\bar{x} = x\,(\theta) = x + \theta d$, with $\theta \geq 0$ and $d \in \mathbb{R}^n$. The increment of the objective function can be written as follows:

$$\begin{aligned}
F\,(\bar{x}) - F\,(x) &= \frac{p^T \bar{x} + p_0}{q^T \bar{x} + q_0} - \frac{p^T x + p_0}{q^T x + q_0} \\
&= \frac{p^T \bar{x} q^T x + p^T \bar{x} q_0 + p_0 q^T x + p_0 q_0 - p^T x q^T \bar{x} - p^T x q_0 - p_0 q^T \bar{x} - p_0 q_0}{(q^T \bar{x} + q_0)\,(q^T x + q_0)} \\
&= \frac{q_0\left(p^T \bar{x} - p^T x\right) - p_0\left(q^T \bar{x} - q^T x\right) - p^T x q^T \bar{x} + p^T \bar{x} q^T x}{(q^T \bar{x} + q_0)\,(q^T x + q_0)} \\
&= \frac{q_0\left(p^T \theta d\right) - p_0\left(q^T \theta d\right) - p^T x q^T \bar{x} + p^T \bar{x} q^T x + p^T x q^T x - p^T x q^T x}{(q^T \bar{x} + q_0)\,(q^T x + q_0)} \\
&= \frac{q_0\left(p^T \theta d\right) - p_0\left(q^T \theta d\right) - p^T x\left(q^T \theta d\right) + q^T x\left(p^T \theta d\right)}{(q^T \bar{x} + q_0)\,(q^T x + q_0)} \\
&= \frac{-\left(p^T x + p_0\right)\left(q^T \theta d\right) + \left(q^T x + q_0\right)\left(p^T \theta d\right)}{(q^T \bar{x} + q_0)\,(q^T x + q_0)} \\
&= \frac{\frac{q^T x + q_0}{q^T x + q_0} p^T \theta d - \frac{p^T x + p_0}{q^T x + q_0} q^T \theta d}{q^T \bar{x} + q_0}.
\end{aligned}$$

Thus

$$F(\bar{x}) - F(x) = \frac{\left(\theta p^T d\right) - F(x)\left(\theta q^T d\right)}{Q(\bar{x})}. \tag{5}$$

In the other hand, we have

$$\begin{cases} Ax = b \\ \\ A\bar{x} = b \end{cases} \Rightarrow \theta A d = 0.$$

By setting $d = \begin{pmatrix} d_B \\ d_N \end{pmatrix}$, $d_B = d(J_B)$, $d_N = d(J_N)$, the equality $Ad = 0$ can also be written as follows:

$$A_B d_B + A_N d_N = 0 \Leftrightarrow d_B = -A_B^{-1} A_N d_N. \tag{6}$$

The increment of the function P corresponding to the numerator is given by:

$$\begin{aligned} P(\bar{x}) - P(x) &= \theta p^T d \\ &= \theta\left(p_B^T d_B + p_N^T . d_N\right) \\ &= -\theta\left(p_B^T A_B^{-1} A_N d_N - p_N^T d_N\right) \\ &= -\theta\left(p_B^T A_B^{-1} A_N - p_N^T\right) d_N \\ &= -\theta\left(\pi_P^T A_N - p_N^T\right) d_N. \end{aligned}$$

Hence

$$P(\bar{x}) - P(x) = \theta p^T d = -\theta d_N^T \Delta'_N. \tag{7}$$

Similarly, we calculate the increment of the denominator function Q and obtain:

$$Q(\bar{x}) - Q(x) = \theta q^T d = -\theta d_N^T \Delta''_N. \tag{8}$$

Hence, the increment (5) becomes

$$F(\bar{x}) - F(x) = \frac{-\theta d_N^T \Delta'_N - F(x)\left(-\theta d_N^T \Delta''_N\right)}{Q(\bar{x})} = \frac{-\theta d_N^T\left[\Delta'_N - F(x)\Delta''_N\right]}{Q(\bar{x})}.$$

Therefore,

$$F(\bar{x}) - F(x) = \frac{-\theta \Delta_N^T(x) d_N}{Q(\bar{x})} = \frac{-\sum_{j \in J_N} \Delta_j(x)(\bar{x}_j - x_j)}{Q(\bar{x})}. \tag{9}$$

3.2. Optimality criterion

We have the following theorem [24, 26, 31]:

Theorem 3.1
(Optimality criterion)
Let $\{x, J_B\}$ be an SFS of the problem (1)-(2). Then the relationships

$$\begin{cases} \Delta_j(x) \geq 0, \ \text{for} \ \ x_j = l_j; & \\ \Delta_j(x) \leq 0, \ \text{for} \ \ x_j = u_j; & j \in J_N \\ \Delta_j(x) = 0, \ \text{for} \ \ l_j < x_j < u_j; & \end{cases} \tag{10}$$

are sufficient for the optimality of the FS x. The same relationships are also necessary when the SFS $\{x, J_B\}$ is nondegenerate.

Proof
Sufficient condition.
Let $\{x, J_B\}$ be an SFS verifying the relationships (10). For any FS $\bar{x}$ of the problem (1)-(2), the increment formula (9) gives

$$F(\bar{x}) - F(x) = \frac{-\sum\limits_{\Delta_j(x)>0, j\in J_N} \Delta_j(x)(\bar{x}_j - l_j) - \sum\limits_{\Delta_j(x)<0, j\in J_N} \Delta_j(x)(\bar{x}_j - u_j)}{Q(\bar{x})}.$$

Since we have $Q(\bar{x}) > 0$ and

$$l_j \le \bar{x}_j \le u_j \;\Rightarrow\; \bar{x}_j - l_j \ge 0 \text{ and } \bar{x}_j - u_j \le 0,$$

then we deduce that

$$F(\bar{x}) - F(x) \le 0 \Rightarrow F(\bar{x}) \le F(x).$$

Therefore, the vector x is an optimal solution of the problem (1)-(2).
Necessary condition.
Let $\{x, J_B\}$ be a nondegenerate optimal SFS for the problem (1)-(2) and suppose that the relationships (10) are not satisfied: there exists at least an index $j_0 \in J_N$, such that

$$\Delta_{j_0}(x) > 0 \text{ and } x_{j_0} > l_{j_0} \text{ or } \Delta_{j_0}(x) < 0 \text{ and } x_{j_0} < u_{j_0}. \tag{11}$$

Then, we construct an other FS $\bar{x} = x + \theta d$, where θ is a real positive number and $d = d(J)$ an n-vector verifying

$$\begin{aligned} &d_{j_0} = -\text{sign}\Delta_{j_0}(x), \\ &d_j = 0,\; j \ne j_0,\; j \in J_N, \\ &d(J_B) = -A_B^{-1} A_N d(J_N) = -A_B^{-1} a_{j_0} d_{j_0} = A_B^{-1} a_{j_0} \text{sign}\Delta_{j_0}(x). \end{aligned} \tag{12}$$

So we have

$$A_B d(J_B) + A_N d(J_N) = Ad = 0 \text{ and } A\bar{x} = A(x + \theta d) = Ax + \theta Ad = b.$$

The vector $\bar{x}$ will be a feasible solution to the problem (1)-(2), if it verifies in addition the following inequality:

$$l \le \bar{x} \le u \Leftrightarrow l \le x + \theta d \le u \Leftrightarrow l - x \le \theta d \le u - x. \tag{13}$$

The previous inequality can also be written as follows:

$$l_j - x_j \le \theta d_j \le u_j - x_j,\;\; j \in J_B \Rightarrow \theta \in [0,\; \theta_{j_1}],$$

where

$$\theta_{j_1} = \min_{j\in J_B} \theta_j, \text{ with } \theta_j = \begin{cases} \frac{u_j - x_j}{d_j}, & \text{if } d_j > 0; \\ \frac{l_j - x_j}{d_j}, & \text{if } d_j < 0; \\ \infty, & \text{if } d_j = 0 \end{cases}$$

and

$$l_{j_0} - x_{j_0} \le -\theta \text{sign}\Delta_{j_0}(x) \le u_{j_0} - x_{j_0} \Rightarrow \theta \in [0, \theta_{j_0}],$$

where

$$\theta_{j_0} = \begin{cases} x_{j_0} - l_{j_0}, & \text{if } \Delta_{j_0}(x) > 0; \\ u_{j_0} - x_{j_0}, & \text{if } \Delta_{j_0}(x) < 0. \end{cases}$$

Since the SFS $\{x, J_B\}$ is nondegenerate ($l_j < x_j < u_j,\; j \in J_B$, i.e., $\theta_{j_1} > 0$) and also $\theta_{j_0} > 0$ from (11), we can choose a positive number θ in the interval $]0,\; \theta^0]$, where $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}$. Thus, the relationships (13) will be verified and the vector $\bar{x}$ will be a feasible solution of the problem (1)-(2). So, the increment formula (9) becomes

$$F(\bar{x}) - F(x) = \frac{-\theta \sum_{j \in J_N} \Delta_j(x) d_j}{Q(\bar{x})} = \frac{-\theta \Delta_{j_0}(x) d_{j_0}}{Q(\bar{x})} = \frac{\theta \Delta_{j_0}(x) \operatorname{sign} \Delta_{j_0}(x)}{Q(\bar{x})} = \frac{\theta |\Delta_{j_0}(x)|}{Q(\bar{x})} > 0.$$

Thus, we have find an SFS $\bar{x} \neq x$, such that $F(\bar{x}) > F(x)$. This contradicts the fact that x is an optimal FS. The relationships (10) are therefore verified. □

3.3. *Suboptimality criterion*

In order to estimate the gap existing between the optimal value $F(x^*)$ and an other value $F(x)$ at an arbitrary SFS $\{x, J_B\}$, we replace in the increment formula (9) the vector $\bar{x}$ with x^* and find an upper bound for the expression. So

$$F(x^*) - F(x) = \frac{-\sum_{j \in J_N} \Delta_j(x)\left(x_j^* - x_j\right)}{Q(x^*)} = \frac{\sum_{\Delta_j(x)>0, j \in J_N} \Delta_j(x)\left(x_j - x_j^*\right) + \sum_{\Delta_j(x)<0, j \in J_N} \Delta_j(x)\left(x_j - x_j^*\right)}{Q(x^*)}.$$

Since $Q(x^*) \geq \alpha > 0$ and the optimal solution x^* verifies

$$l_j \leq x_j^* \leq u_j, \;\; j \in J,$$

we obtain

$$x_j - x_j^* \leq x_j - l_j \text{ and } x_j - x_j^* \geq x_j - u_j.$$

Hence

$$\Delta_j(x)\left(x_j - x_j^*\right) \leq \Delta_j(x)(x_j - l_j), \;\; \text{if} \;\; \Delta_j(x) > 0;$$

$$\Delta_j(x)\left(x_j - x_j^*\right) \leq \Delta_j(x)(x_j - u_j), \;\; \text{if} \;\; \Delta_j(x) < 0.$$

Therefore, we obtain:

$$F(x^*) - F(x) \leq \frac{\sum_{\Delta_j(x)>0, j \in J_N} \Delta_j(x)(x_j - l_j) + \sum_{\Delta_j(x)<0, j \in J_N} \Delta_j(x)(x_j - u_j)}{\alpha}. \tag{14}$$

We call the nonnegative quantity

$$\beta(x, J_B) = \frac{1}{\alpha}\left(\sum_{\Delta_j(x)>0, j \in J_N} \Delta_j(x)(x_j - l_j) + \sum_{\Delta_j(x)<0, j \in J_N} \Delta_j(x)(x_j - u_j)\right) \tag{15}$$

the suboptimality estimate.

So we have the following theorem [24, 26, 31]:

Theorem 3.2

(Suboptimality criterion)

Let $\{x, J_B\}$ be an SFS of the problem (1)-(2) and ϵ be a nonnegative arbitrary number. If

$$\beta(x, J_B) \leq \epsilon, \tag{16}$$

then the FS x is ϵ-optimal.

Proof
By using (14) and (15), we can write

$$F\left(x^{*}\right)-F\left(x\right)\leq\beta\left(x,J_{B}\right)\leq\epsilon\Rightarrow F\left(x^{*}\right)-F\left(x\right)\leq\epsilon.$$

The FS x is then ϵ-optimal. □

3.4. Algorithm of the primal support method

Let $\{x, J_B\}$ be an initial SFS and ϵ be an arbitrary nonnegative number. The scheme of the algorithm of the primal support method for solving LFPBV is presented in the following steps [24, 26, 31]:

Algorithm 1. (Primal Support Method for LFP)

1. Calculate $F\left(x\right)$ and

$$\pi_P^T=p_B^TA_B^{-1},\ \Delta_j'=\pi_P^Ta_j-p_j,\ j\in J_N,$$

$$\pi_Q^T=q_B^TA_B^{-1},\ \Delta_j''=\pi_Q^Ta_j-q_j,\ j\in J_N,$$

$$\Delta_j\left(x\right)=\ \Delta_j'-F\left(x\right)\Delta_j'',\ \ j\in J_N;$$

2. Calculate the suboptimality estimate with (15);

3. If $\beta\left(x,J_B\right)=0$, then the algorithm stops with the optimal SFS $\{x, J_B\}$;

4. If $\beta\left(x,J_B\right)\leq\epsilon$, then the algorithm stops with the ϵ-optimal SFS $\{x, J_B\}$;

5. Else, go to step **6**;

6. Determine the set of indices which do not verify the optimality relationships (10):

$$J_{NNO}=\{j\in J_N:[\Delta_j\left(x\right)>0\text{ and }x_j>l_j]\text{ or }[\Delta_j\left(x\right)<0\text{ and }x_j<u_j]\};$$

7. Choose the index j_0, such that $|\Delta_{j_0}\left(x\right)|=\max\limits_{j\in J_{NNO}}|\Delta_j\left(x\right)|$;

8. Calculate the ascent direction d with the relationships:

$$d_{j_0}=-\text{sign}\Delta_{j_0}\left(x\right),$$

$$d_j=0,\ j\neq j_0,\ \ j\in J_N,$$

$$d\left(J_B\right)=-A_B^{-1}A_Nd\left(J_N\right)=A_B^{-1}a_{j_0}\text{sign}\Delta_{j_0}\left(x\right);$$

9. Calculate $\theta_{j_1}=\min\limits_{j\in J_B}\theta_j$, where θ_j is determined with the formula:

$$\theta_j=\begin{cases}\frac{u_j-x_j}{d_j}, & \text{if } d_j>0;\\ \frac{l_j-x_j}{d_j}, & \text{if } d_j<0;\\ \infty, & \text{if } d_j=0;\end{cases}$$

10. Calculate θ_{j_0} using the formula:

$$\theta_{j_0}=\begin{cases}x_{j_0}-l_{j_0}, & \text{if } \Delta_{j_0}\left(x\right)>0;\\ u_{j_0}-x_{j_0}, & \text{if } \Delta_{j_0}\left(x\right)<0;\end{cases}$$

11. Calculate $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}$;

12. Calculate $\bar{x} = x + \theta^0 d$ and $F(\bar{x}) = F(x) + \frac{\theta^0 |\Delta_{j_0}(x)|}{Q(\bar{x})}$;

13. Calculate

$$\Delta_N(\bar{x}) = \Delta'_N - F(\bar{x})\Delta''_N;$$

and

$$\beta(\bar{x}, J_B) = \frac{1}{\alpha}\left(\sum_{\Delta_j(\bar{x})>0, j\in J_N} \Delta_j(\bar{x})(\bar{x}_j - l_j) + \sum_{\Delta_j(\bar{x})<0, j\in J_N} \Delta_j(\bar{x})(\bar{x}_j - u_j)\right);$$

14. If $\beta(\bar{x}, J_B) = 0$, then the algorithm stops with the optimal SFS $\{\bar{x}, J_B\}$;

15. If $\beta(\bar{x}, J_B) \leq \epsilon$, then the algorithm stops with the ϵ-optimal SFS $\{\bar{x}, J_B\}$;

16. Else, go to step **17**;

17. If $\theta^0 = \theta_{j_0}$, then we put $\bar{J}_B = J_B$;

18. If $\theta^0 = \theta_{j_1}$, then we put $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$;

19. Put $x := \bar{x}$, $J_B := \bar{J}_B$ and go to step **1**;

Remark 1. The primal support method for solving LFP problems with nonnegative variables and those with bounded variables [24, 26, 31] uses the simplex direction defined by relationships (12) in order to improve the current SFS. When we start with an initial basic feasible solution $\{x, J_B\}$, i.e., $x_j = l_j \vee u_j,\ j \in J_N$, the primal support method will pass from one extreme point to a better adjacent one and the path followed by the support method algorithm will be exactly the same one as that of the simplex algorithm.

4. The hybrid direction method

In [22], a new hybrid direction method is proposed for solving LP problems. In this section, we generalize this method in order to solve LFP problems with bounded variables.

Let $\{x, J_B\}$ be an SFS for problem (1)-(2) and $\eta > 0$. Define the following index sets:

$$\begin{cases} J_{NE}^{+} = \{j \in J_N : \ \ \Delta_j(x) > \eta(x_j - l_j) \ \text{ and } \ x_j > l_j\}, \\ J_{NE}^{-} = \{j \in J_N : \ \ \Delta_j(x) < \eta(x_j - u_j) \ \text{ and } \ x_j < u_j\}, \\ J_{NI}^{+} = \{j \in J_N : \ \ 0 < \Delta_j(x) \leq \eta(x_j - l_j)\}, \\ J_{NI}^{-} = \{j \in J_N : \ \ \eta(x_j - u_j) \leq \Delta_j(x) < 0\}, \\ J_{NR}^{+} = \{j \in J_N : \ \ \Delta_j(x) > 0 \ \text{ and } \ x_j = l_j\}, \\ J_{NR}^{-} = \{j \in J_N : \ \ \Delta_j(x) < 0 \ \text{ and } \ x_j = u_j\}, \\ J_N^{+} = \{j \in J_N : \Delta_j(x) > 0\},\ J_N^{-} = \{j \in J_N : \Delta_j(x) < 0\}, \\ J_{N0} = \{j \in J_N : \Delta_j(x) = 0\},\ J_{NI} = J_{NI}^{+} \cup J_{NI}^{-}, \\ J_{NE} = J_{NE}^{+} \cup J_{NE}^{-},\ J_{NR} = J_{N0} \cup J_{NR}^{+} \cup J_{NR}^{-}. \end{cases} \tag{17}$$

Then

$$J_N = J_{NE} \cup J_{NI} \cup J_{NR},\ J_N^{+} = J_{NE}^{+} \cup J_{NI}^{+} \cup J_{NR}^{+},\ J_N^{-} = J_{NE}^{-} \cup J_{NI}^{-} \cup J_{NR}^{-}.$$

Lemma 4.1

If $\{x, J_B\}$ is nonoptimal, then $J_{NI} \cup J_{NE} \neq \emptyset$.

Proof
We suppose that there exists an index $j_* \in J_N$ which does not verify the optimality conditions (10), so two cases can occur:
Case 1: If $\Delta_{j_*}(x) < 0$ and $x_{j_*} < u_{j_*}$, then $j_* \in J^-_{NI} \cup J^-_{NE}$. Hence $J_{NI} \cup J_{NE} \neq \emptyset$.
Case 2: If $\Delta_{j_*}(x) > 0$ and $x_{j_*} > l_{j_*}$, then $j_* \in J^+_{NI} \cup J^+_{NE}$. Hence $J_{NI} \cup J_{NE} \neq \emptyset$. □

For $\eta > 0$, we define the quantities γ and μ as follows:

$$\gamma = \frac{1}{\alpha}\left(\sum_{j\in J^+_{NI}} \Delta_j(x)(x_j - l_j) + \sum_{j\in J^-_{NI}} \Delta_j(x)(x_j - u_j) + \frac{1}{\eta}\sum_{j\in J^+_{NE}\cup J^-_{NE}} \Delta^2_j(x)\right), \tag{18}$$

$$\mu = \frac{1}{\alpha}\left(-\sum_{j\in J^+_{NE}} \Delta_j(x)(x_j - l_j) - \sum_{j\in J^-_{NE}} \Delta_j(x)(x_j - u_j) + \frac{1}{\eta}\sum_{j\in J^+_{NE}\cup J^-_{NE}} \Delta^2_j(x)\right). \tag{19}$$

We recall that the suboptimality estimate $\beta(x, J_B)$ is given by:

$$\beta = \beta(x, J_B) = \frac{1}{\alpha}\left(\sum_{j\in J^+_{N}} \Delta_j(x)(x_j - l_j) + \sum_{j\in J^-_{N}} \Delta_j(x)(x_j - u_j)\right). \tag{20}$$

Remark 2. In the adaptive method for LP, the suboptimality estimate is equal to the difference between the value of the objective function of the primal problem at the current primal solution and that of the dual problem at the current dual solution which is judiciously constructed. As a consequence of duality theory, this estimate decreases when we pass from x to $\bar{x}$ and from J_B to $\bar{J}_B$. However in this work, the passage from x to $\bar{x}$ ensures that $F(\bar{x}) \geq F(x)$, or $F(\bar{x}) > F(x)$ in the case of primal nondegeneracy, but the suboptimality estimate can increase when we move from x to $\bar{x}$. Hence, the suboptimality estimate is only used here as an upper bound for the difference between the maximum and the value of F at the current solution, which allows us to stop the proposed algorithm when a suboptimal or optimal solution is reached.

Lemma 4.2
For $\eta > 0$, the following inequalities hold:

$$\beta = \gamma - \mu \leq \gamma,\ \ \gamma \geq 0, \mu \geq 0.$$

Proof
First, remark that

$$\begin{aligned}\beta = \beta(x, J_B) &= \tfrac{1}{\alpha}\left(\sum_{j\in J^+_{N}} \Delta_j(x)(x_j - l_j) + \sum_{j\in J^-_{N}} \Delta_j(x)(x_j - u_j)\right)\\ &= \tfrac{1}{\alpha}\left(\sum_{j\in J^+_{NE}\cup J^+_{NI}} \Delta_j(x)(x_j - l_j) + \sum_{j\in J^-_{NE}\cup J^-_{NI}} \Delta_j(x)(x_j - u_j)\right).\end{aligned}$$

Thus,

$$\beta + \mu = \frac{1}{\alpha}\left(\sum_{j\in J^+_{NI}} \Delta_j(x)(x_j - l_j) + \sum_{j\in J^-_{NI}} \Delta_j(x)(x_j - u_j) + \frac{1}{\eta}\sum_{j\in J^+_{NE}\cup J^-_{NE}} \Delta^2_j(x)\right) = \gamma,$$

that implies $\beta = \gamma - \mu$.
Furthermore, for $\eta > 0$, we have

$$\begin{aligned}\beta=\beta(x,J_B) &= \tfrac{1}{\alpha}\Bigg(\sum_{j\in J_{NI}^+}\Delta_j(x)(x_j-l_j)+\sum_{j\in J_{NI}^-}\Delta_j(x)(x_j-u_j)\\ &\quad+\sum_{j\in J_{NE}^+}\Delta_j(x)(x_j-l_j)+\sum_{j\in J_{NE}^-}\Delta_j(x)(x_j-u_j)\Bigg)\\ &\leq \tfrac{1}{\alpha}\Bigg(\sum_{j\in J_{NI}^+}\Delta_j(x)(x_j-l_j)+\sum_{j\in J_{NI}^-}\Delta_j(x)(x_j-u_j)\\ &\quad+\tfrac{1}{\eta}\sum_{j\in J_{NE}^+}\Delta_j^2(x)+\tfrac{1}{\eta}\sum_{j\in J_{NE}^-}\Delta_j^2(x)\Bigg).\end{aligned}$$

Indeed,

for $j\in J_{NE}^+ : \eta(x_j-l_j)<\Delta_j(x)\Rightarrow x_j-l_j<\frac{\Delta_j(x)}{\eta}\Rightarrow \Delta_j(x)(x_j-l_j)<\frac{\Delta_j^2(x)}{\eta}$;

for $j\in J_{NE}^- : \Delta_j(x)<\eta(x_j-u_j)\Rightarrow x_j-u_j>\frac{\Delta_j(x)}{\eta}\Rightarrow \Delta_j(x)(x_j-u_j)<\frac{\Delta_j^2(x)}{\eta}$.

Hence,

$$\beta\leq\frac{1}{\alpha}\left(\sum_{j\in J_{NI}^+}\Delta_j(x)(x_j-l_j)+\sum_{j\in J_{NI}^-}\Delta_j(x)(x_j-u_j)+\frac{1}{\eta}\sum_{j\in J_{NE}^+\cup J_{NE}^-}\Delta_j^2(x)\right)=\gamma,$$

that implies $\gamma\geq\beta\geq 0$. Since $\beta=\gamma-\mu$, we conclude that $\mu=\gamma-\beta\geq 0$. □

4.1. An iteration of the algorithm

4.1.1. Change of the feasible solution Let $\{x,J_B\}$ be an SFS for problem (1)-(2) and $\eta>0$. We define the direction d as follows:

$$\begin{cases} d_j=l_j-x_j, & \text{if} \quad j\in J_{NI}^+;\\ d_j=u_j-x_j, & \text{if} \quad j\in J_{NI}^-;\\ d_j=\frac{-\Delta_j(x)}{\eta}, & \text{if} \quad j\in J_{NE}^+\cup J_{NE}^-;\\ d_j=0, & \text{if} \quad j\in J_{NR};\\ d_B=-A_B^{-1}A_Nd_N. \end{cases} \tag{21}$$

This direction is called a hybrid direction [22], and it is clear that $Ad=0$.
In order to improve the value of the objective function while remaining in the feasible region, we calculate the steplength θ^0 along the direction d. This steplength must satisfy the following inequality:

$$l\leq x+\theta^0d\leq u \Leftrightarrow \begin{cases} l_j-x_j\leq\theta^0d_j\leq u_j-x_j, & \text{for} \quad j\in J_B;\\ l_j-x_j\leq\theta^0d_j\leq u_j-x_j, & \text{for} \quad j\in J_N.\end{cases}$$

Therefore, we calculate the steplength θ^0 as follows:

$$\theta^0=\min\{\theta_{j_1},\theta_{j_2},1\},\ \ \theta_{j_1}=\min_{j\in J_B}\theta_j,\ \ \theta_{j_2}=\begin{cases}\min\limits_{j\in J_{NE}}\theta_j, & \text{if} \quad J_{NE}\neq\emptyset;\\ \infty, & \text{otherwise,}\end{cases} \tag{22}$$

where

$$\theta_j = \begin{cases} \frac{u_j - x_j}{d_j}, & \text{if } d_j > 0; \\ \frac{l_j - x_j}{d_j}, & \text{if } d_j < 0; \\ \infty, & \text{if } d_j = 0. \end{cases} \tag{23}$$

The value of the steplength is equal to 1 for the indices of J_{NI}.
Then the new improved feasible solution is given by

$$\bar{x} = x + \theta^0 d, \tag{24}$$

where d and θ^0 are defined by relationships (21)-(23).
Moreover, the increment of the objective function is:

$$\begin{aligned} F(\bar{x}) - F(x) &= \frac{-\theta^0 \sum\limits_{j \in J_N} \Delta_j(x) d_j}{Q(\bar{x})} \\ &= \frac{1}{Q(\bar{x})} \left(-\theta^0 \sum_{j \in J_{NI}^+} \Delta_j(x) d_j - \theta^0 \sum_{j \in J_{NI}^-} \Delta_j(x) d_j - \theta^0 \sum_{j \in J_{NE}^+ \cup J_{NE}^-} \Delta_j(x) d_j \right) \\ &= \frac{\theta^0}{Q(\bar{x})} \left(\sum_{j \in J_{NI}^+} \Delta_j(x)(x_j - l_j) + \sum_{j \in J_{NI}^-} \Delta_j(x)(x_j - u_j) + \sum_{j \in J_{NE}^+ \cup J_{NE}^-} \frac{\Delta_j^2(x)}{\eta} \right) \\ &= \frac{\alpha\theta^0}{Q(\bar{x})} \gamma = \frac{\alpha\theta^0}{Q(\bar{x})} (\beta + \mu) \geq 0. \end{aligned}$$

Let us define the following index sets:

$$\tilde{J}_N^+ = \{j \in J_N : \Delta_j(\bar{x}) > 0\}, \; \tilde{J}_N^- = \{j \in J_N : \Delta_j(\bar{x}) < 0\}.$$

The other corresponding sets are $\tilde{J}_{NI}^+$, $\tilde{J}_{NI}^-$, $\tilde{J}_{NE}^+$, $\tilde{J}_{NE}^-$ and $\tilde{J}_{NR}$.
We compute the suboptimality estimate corresponding to the new SFS $\{\bar{x}, J_B\}$:

$$\bar{\beta} = \beta(\bar{x}, J_B) = \frac{1}{\alpha} \left(\sum_{j \in \tilde{J}_N^+} \Delta_j(\bar{x})(\bar{x}_j - l_j) + \sum_{j \in \tilde{J}_N^-} \Delta_j(\bar{x})(\bar{x}_j - u_j) \right). \tag{25}$$

If $\bar{\beta} \leq \epsilon$, then the FS $\bar{x}$ is ϵ-optimal. If $\bar{\beta} > \epsilon$ and $\theta^0 = \theta_{j_2} \vee 1$, then we put $x := \bar{x}$ and we start a new iteration with the SFS $\{x, J_B\}$. If $\bar{\beta} > \epsilon$ and $\theta^0 = \theta_{j_1}$, then we change the support J_B.

4.1.2. Change of the support We define the n-vectors $\bar{d} = (\bar{d}_j, j \in J)$, κ and $t = (t_j, \; j \in J)$, as follows:

$$\begin{cases} \bar{d}_j = l_j - \bar{x}_j, & \text{if} \quad j \in \tilde{J}_{NI}^+; \\ \bar{d}_j = u_j - \bar{x}_j, & \text{if} \quad j \in \tilde{J}_{NI}^-; \\ \bar{d}_j = \frac{-\Delta_j(\bar{x})}{\eta}, & \text{if} \quad j \in \tilde{J}_{NE}^+ \cup \tilde{J}_{NE}^-; \\ \bar{d}_j = 0, & \text{if} \quad j \in \tilde{J}_{NR}; \\ \bar{d}_B = -A_B^{-1} A_N \bar{d}_N, \text{ with } \bar{d}_N = (\bar{d}_j, \; j \in J_N); \end{cases} \tag{26}$$

$$\kappa = \bar{x} + \bar{d}; \;\; t_{j_1} = -\text{sign } d_{j_1}, \; t_j = 0, \; j \neq j_1, \; j \in J_B; \; t_N^T = t_B^T A_B^{-1} A_N.$$

Remark 3. The vectors t and $\bar{d}$ are orthogonal. Indeed,

$$t^T \bar{d} = t_N^T \bar{d}_N + t_B^T \bar{d}_B = (t_B^T A_B^{-1} A_N)\bar{d}_N + t_B^T(-A_B^{-1} A_N \bar{d}_N) = 0.$$

The new support $\bar{J}_B$ and the new reduced costs vectors $\bar{\Delta}'$, $\bar{\Delta}''$ and $\bar{\Delta}(\bar{x})$ are computed as follows:

$$\begin{aligned} &\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}, \ \bar{J}_N = J \setminus \bar{J}_B, \\ &\bar{\Delta}' = \Delta' + \sigma_0' t, \ \bar{\Delta}'' = \Delta'' + \sigma_0'' t \ \text{ and } \ \bar{\Delta}(\bar{x}) = \Delta(\bar{x}) + \sigma^0 t, \end{aligned} \tag{27}$$

where

$$\sigma^0 = \sigma_{j_0} = \min_{j \in J_N} \{\sigma_j\}, \text{ with } \sigma_j = \begin{cases} \frac{-\Delta_j(\bar{x})}{t_j}, & \text{if } \ \Delta_j(\bar{x}) t_j < 0; \\ 0, & \text{if } \ \Delta_j(\bar{x}) = 0, \ t_j < 0, \ \kappa_j \neq u_j; \\ 0, & \text{if } \ \Delta_j(\bar{x}) = 0, \ t_j > 0, \ \kappa_j \neq l_j; \\ +\infty, & \text{otherwise}; \end{cases} \tag{28}$$

$$\sigma_0' = \begin{cases} \frac{-\Delta'_{j_0}}{t_{j_0}}, & \text{if } (\Delta_{j_0}(\bar{x}) t_{j_0} < 0) \text{ or } (\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} < 0, \ \kappa_{j_0} \neq u_{j_0}) \text{ or } (\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} > 0, \ \kappa_{j_0} \neq l_{j_0}); \\ +\infty, & \text{otherwise}; \end{cases} \tag{29}$$

$$\sigma_0'' = \begin{cases} \frac{-\Delta''_{j_0}}{t_{j_0}}, & \text{if } (\Delta_{j_0}(\bar{x}) t_{j_0} < 0) \text{ or } (\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} < 0, \ \kappa_{j_0} \neq u_{j_0}) \text{ or } (\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} > 0, \ \kappa_{j_0} \neq l_{j_0}); \\ s \times (+\infty), & \text{otherwise}; \end{cases} \tag{30}$$

with $s = -\text{sign} F(\bar{x})$.

Proposition 1

We have $\sigma_0', \sigma_0'' \in \mathbb{R}$, $\sigma^0 \geq 0$, with

$$\sigma^0 = \sigma_0' - F(\bar{x})\sigma_0'', \ \ \bar{\Delta}(\bar{x}) = \bar{\Delta}' - F(\bar{x})\bar{\Delta}'' \ \text{ and } \ \Delta_j(\bar{x})\bar{\Delta}_j(\bar{x}) \geq 0, \ j \in J.$$

Proof
First, we prove the equality $\sigma^0 = \sigma_0' - F(\bar{x})\sigma_0''$. Two cases can occur:
(i) $(\Delta_{j_0}(\bar{x}) t_{j_0} < 0)$ or $(\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} < 0, \ \kappa_{j_0} \neq u_{j_0})$ or $(\Delta_{j_0}(\bar{x}) = 0, \ t_{j_0} > 0, \ \kappa_{j_0} \neq l_{j_0})$.
In this case, $t_{j_0} \neq 0$, so we get

$$\sigma^0 = \frac{-\Delta_{j_0}(\bar{x})}{t_{j_0}} = \frac{-\Delta'_{j_0} + F(\bar{x})\Delta''_{j_0}}{t_{j_0}} = \frac{-\Delta'_{j_0}}{t_{j_0}} + F(\bar{x})\frac{\Delta''_{j_0}}{t_{j_0}} = \sigma_0' - F(\bar{x})\sigma_0''.$$

(ii) $\sigma^0 = +\infty$. In this case, we have

$$\sigma_0' = +\infty, \ \sigma_0'' = -\text{sign} F(\bar{x}) \times (+\infty) \Rightarrow \sigma_0' - F(\bar{x})\sigma_0'' = +\infty = \sigma^0.$$

Now, let us prove that $\bar{\Delta}(\bar{x}) = \bar{\Delta}' - F(\bar{x})\bar{\Delta}''$.
We have

$$\bar{\Delta}(\bar{x}) = \Delta(\bar{x}) + \sigma^0 t = \Delta' - F(\bar{x})\Delta'' + (\sigma_0' - F(\bar{x})\sigma_0'')t = (\Delta' + \sigma_0' t) - F(\bar{x})(\Delta'' + \sigma_0'' t),$$

that implies $\bar{\Delta}(\bar{x}) = \bar{\Delta}' - F(\bar{x})\bar{\Delta}''$. Finally, we have

$$\bar{\Delta}_j(\bar{x})\Delta_j(\bar{x}) = (\Delta_j(\bar{x}) + \sigma^0 t_j)\Delta_j(\bar{x}) = \Delta_j^2(\bar{x}) + \sigma^0 t_j \Delta_j(\bar{x}), \ j \in J. \tag{31}$$

- If $t_j \Delta_j(\bar{x}) > 0$, then $\bar{\Delta}_j(\bar{x})\Delta_j(\bar{x}) \geq 0$.

- If $t_j\Delta_j(\bar{x}) < 0$, then
$$\sigma^0 \le \frac{-\Delta_j(\bar{x})}{t_j} \Rightarrow \sigma^0 + \frac{\Delta_j(\bar{x})}{t_j} \le 0 \Rightarrow \sigma^0 t_j\Delta_j(\bar{x}) + \Delta_j^2(\bar{x}) \ge 0,$$
and from (31) we get $\bar{\Delta}_j(\bar{x})\Delta_j(\bar{x}) \ge 0$.
- If $t_j\Delta_j(\bar{x}) = 0$, then from (31) we deduce $\bar{\Delta}_j(\bar{x})\Delta_j(\bar{x}) = \Delta_j^2(\bar{x}) \ge 0$.

□

Remark 4.
Denote by
$$A_{\bar{B}} = A(I, \bar{J}_B),\ p_{\bar{B}} = (p_j,\ j \in \bar{J}_B),\ q_{\bar{B}} = (q_j,\ j \in \bar{J}_B),\ \bar{\pi}_P^T = p_{\bar{B}}^T A_{\bar{B}}^{-1},\ \bar{\pi}_Q^T = q_{\bar{B}}^T A_{\bar{B}}^{-1}.$$
Then, with the choice of the leaving index j_0 with (28), it is well known in linear programming (see [37]) that the new reduced costs vectors computed with the updating formulas (27) are equal to:
$$\bar{\Delta}' = A^T\bar{\pi}_P - p,\ \bar{\Delta}'' = A^T\bar{\pi}_Q - q.$$

Remark 5. In this work, we have supposed that an initial SFS is available for the problem (1)-(2) and that $\sigma^0 < \infty$. When the initial SFS is not known in advance, it can be computed with the initialization procedures described in [28, 5].

The suboptimality estimate corresponding to the new SFS $\{\bar{x}, \bar{J}_B\}$ is given by:
$$\bar{\bar{\beta}} = \beta(\bar{x}, \bar{J}_B) = \frac{1}{\alpha}\left(\sum_{j\in\bar{J}_N, \bar{\Delta}_j(\bar{x})>0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - l_j) + \sum_{j\in\bar{J}_N, \bar{\Delta}_j(\bar{x})<0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - u_j)\right). \tag{32}$$
Let us define the index sets $\tilde{J}_{N0}$, $\tilde{J}_{N0}^+$, $\tilde{J}_{N0}^-$ and the quantity V_0, as follows:
$$\tilde{J}_{N0} = \{j \in J_N : \Delta_j(\bar{x}) = 0\}, \tilde{J}_{N0}^+ = \{j \in \tilde{J}_{N0} : t_j > 0\}, \tilde{J}_{N0}^- = \{j \in \tilde{J}_{N0} : t_j < 0\}; \tag{33}$$
$$V_0 = \frac{1}{\alpha}\left(t_{j_1}\bar{d}_{j_1} + \sum_{j\in\tilde{J}_{N0}^+\cup\tilde{J}_{NE}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{N0}^-\cup\tilde{J}_{NE}^-} t_j(\kappa_j - u_j)\right). \tag{34}$$
Then, we have the following proposition.

Proposition 2
We can write the suboptimality estimate $\beta(\bar{x}, \bar{J}_B)$ as follows:
$$\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0 V_0. \tag{35}$$

Proof
We have $\bar{J}_N = (J_N \setminus \{j_0\}) \cup \{j_1\}$. If $\sigma^0 = 0$, then $j_0 \in \tilde{J}_{N0}^+$ or $j_0 \in \tilde{J}_{N0}^-$.
So $\Delta_{j_0}(\bar{x}) = 0 \Rightarrow \bar{\Delta}_{j_0}(\bar{x}) = \Delta_{j_0}(\bar{x}) + \sigma^0 t_{j_0} = 0$.
If $\sigma^0 > 0$, then
$$\bar{\Delta}_{j_0}(\bar{x}) = \Delta_{j_0}(\bar{x}) + \sigma^0 t_{j_0} = \Delta_{j_0}(\bar{x}) - \frac{\Delta_{j_0}(\bar{x})}{t_{j_0}} t_{j_0} = 0.$$
Therefore,
$$\bar{\bar{\beta}} = \begin{cases} \frac{1}{\alpha}\left(\sum\limits_{j\in J_N, \bar{\Delta}_j(\bar{x})>0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - l_j) + \sum\limits_{j\in J_N, \bar{\Delta}_j(\bar{x})<0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - u_j) + \bar{\Delta}_{j_1}(\bar{x})(\bar{x}_{j_1} - l_{j_1})\right), & \text{if } \bar{\Delta}_{j_1}(\bar{x}) > 0; \\ \frac{1}{\alpha}\left(\sum\limits_{j\in J_N, \bar{\Delta}_j(\bar{x})>0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - l_j) + \sum\limits_{j\in J_N, \bar{\Delta}_j(\bar{x})<0} \bar{\Delta}_j(\bar{x})(\bar{x}_j - u_j) + \bar{\Delta}_{j_1}(\bar{x})(\bar{x}_{j_1} - u_{j_1})\right), & \text{if } \bar{\Delta}_{j_1}(\bar{x}) < 0. \end{cases}$$

Since $\Delta_{j_1}(x)=0$, we have $\Delta'_{j_1}=\Delta''_{j_1}=0$ and $\Delta_{j_1}(\bar{x})=\Delta'_{j_1}-F(\bar{x})\Delta''_{j_1}=0$, thus

$$\bar{\Delta}_{j_1}(\bar{x})=\Delta_{j_1}(\bar{x})+\sigma^0 t_{j_1}=\sigma^0 t_{j_1}.$$

So two cases can occur:

- If $\bar{\Delta}_{j_1}(\bar{x})>0$, then we have $t_{j_1}=1 \Rightarrow d_{j_1}<0$. Hence

$$\bar{x}_{j_1}-l_{j_1}=x_{j_1}+\theta^0 d_{j_1}-l_{j_1}=x_{j_1}+\frac{l_{j_1}-x_{j_1}}{d_{j_1}}d_{j_1}-l_{j_1}=0.$$

- If $\bar{\Delta}_{j_1}(\bar{x})<0$, then we have $t_{j_1}=-1 \Rightarrow d_{j_1}>0$. Hence

$$\bar{x}_{j_1}-u_{j_1}=x_{j_1}+\theta^0 d_{j_1}-u_{j_1}=x_{j_1}+\frac{u_{j_1}-x_{j_1}}{d_{j_1}}d_{j_1}-u_{j_1}=0.$$

Furthermore, from $\Delta_j(\bar{x})\bar{\Delta}_j(\bar{x})\geq 0$, we deduce that

$$\bar{\Delta}_j(\bar{x})>0 \Leftrightarrow [(\Delta_j(\bar{x})>0) \text{ or } (\Delta_j(\bar{x})=0 \text{ and } t_j>0)]$$

and

$$\bar{\Delta}_j(\bar{x})<0 \Leftrightarrow [(\Delta_j(\bar{x})<0) \text{ or } (\Delta_j(\bar{x})=0 \text{ and } t_j<0)]\,.$$

Hence

$$\begin{aligned}
\bar{\bar{\beta}} &= \tfrac{1}{\alpha}\left(\sum_{j\in J_N,\bar{\Delta}_j(\bar{x})>0}\bar{\Delta}_j(\bar{x})(\bar{x}_j-l_j)+\sum_{j\in J_N,\bar{\Delta}_j(\bar{x})<0}\bar{\Delta}_j(\bar{x})(\bar{x}_j-u_j)\right)\\
&= \tfrac{1}{\alpha}\left(\sum_{j\in J_N,\Delta_j(\bar{x})>0}(\Delta_j(\bar{x})+\sigma^0 t_j)(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})<0}(\Delta_j(\bar{x})+\sigma^0 t_j)(\bar{x}_j-u_j)\right.\\
&\quad\left.+\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j>0}(\Delta_j(\bar{x})+\sigma^0 t_j)(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j<0}(\Delta_j(\bar{x})+\sigma^0 t_j)(\bar{x}_j-u_j)\right)\\
&= \tfrac{1}{\alpha}\left[\left(\sum_{j\in J_N,\Delta_j(\bar{x})>0}\Delta_j(\bar{x})(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})<0}\Delta_j(\bar{x})(\bar{x}_j-u_j)\right)\right.\\
&\quad+\sigma^0\left(\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j>0}t_j(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j<0}t_j(\bar{x}_j-u_j)\right)\\
&\quad\left.+\sigma^0\left(\sum_{j\in J_N,\Delta_j(\bar{x})>0}t_j(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})<0}t_j(\bar{x}_j-u_j)\right)\right]\\
&= \beta(\bar{x},J_B)+\sigma^0(\alpha_1+\alpha_2),
\end{aligned}$$

where

$$\alpha_1=\frac{1}{\alpha}\left(\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j>0}t_j(\bar{x}_j-l_j)+\sum_{j\in J_N,\Delta_j(\bar{x})=0,t_j<0}t_j(\bar{x}_j-u_j)\right)$$

and

$$\alpha_2 = \frac{1}{\alpha}\left(\sum_{j\in J_N,\Delta_j(\bar{x})>0} t_j(\bar{x}_j - l_j) + \sum_{j\in J_N,\Delta_j(\bar{x})<0} t_j(\bar{x}_j - u_j)\right).$$

So, we have:

$$\alpha_1 = \frac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{N0}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{N0}} t_j(\bar{x}_j - u_j)\right),\ \alpha_2 = \frac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{N}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{N}} t_j(\bar{x}_j - u_j)\right).$$

For $j \in \tilde{J}^+_{N0} \cup \tilde{J}^-_{N0} \subset \tilde{J}_{NR}$, $\bar{d}_j = 0 \Rightarrow \bar{x}_j = \kappa_j$. Then

$$\alpha_1 = \frac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{N0}} t_j(\kappa_j - l_j) + \sum_{j\in \tilde{J}^-_{N0}} t_j(\kappa_j - u_j)\right). \tag{36}$$

Now, we compute the quantity α_2:

$$\alpha_2 = \frac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{NE}\cup\tilde{J}^+_{NI}\cup\tilde{J}^+_{NR}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{NE}\cup\tilde{J}^-_{NI}\cup\tilde{J}^-_{NR}} t_j(\bar{x}_j - u_j)\right).$$

For $j \in \tilde{J}^+_{NR},\ \bar{x}_j = l_j$ and for $j \in \tilde{J}^-_{NR},\ \bar{x}_j = u_j$. Hence,

$$\begin{aligned}\alpha_2 &= \tfrac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{NE}\cup\tilde{J}^+_{NI}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{NE}\cup\tilde{J}^-_{NI}} t_j(\bar{x}_j - u_j)\right)\\ &= \tfrac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{NE}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{NE}} t_j(\bar{x}_j - u_j) + \sum_{\tilde{J}^+_{NI}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{NI}} t_j(\bar{x}_j - u_j)\right).\end{aligned}$$

For $j \in \tilde{J}^+_{NI},\ \bar{x}_j - l_j = -\bar{d}_j$, and for $j \in \tilde{J}^-_{NI},\ \bar{x}_j - u_j = -\bar{d}_j$. Thus, we obtain:

$$\alpha_2 = \frac{1}{\alpha}\left(\sum_{j\in \tilde{J}^+_{NE}} t_j(\bar{x}_j - l_j) + \sum_{j\in \tilde{J}^-_{NE}} t_j(\bar{x}_j - u_j) - \left(\sum_{j\in \tilde{J}^+_{NI}} t_j\bar{d}_j + \sum_{j\in \tilde{J}^-_{NI}} t_j\bar{d}_j\right)\right).$$

Since we have

$$\sum_{j\in J} t_j\bar{d}_j = \sum_{j\in J_N} t_j\bar{d}_j + \sum_{j\in J_B} t_j\bar{d}_j = 0;\ \bar{d}_j = 0,\ j \in \tilde{J}_{NR} \text{ and } \sum_{j\in J_B} t_j\bar{d}_j = t_{j_1}\bar{d}_{j_1},$$

we deduce

$$\sum_{j\in J_N} t_j\bar{d}_j = -\sum_{j\in J_B} t_j\bar{d}_j \Rightarrow \sum_{j\in \tilde{J}^+_{NI}} t_j\bar{d}_j + \sum_{j\in \tilde{J}^-_{NI}} t_j\bar{d}_j + \sum_{j\in \tilde{J}^+_{NE}\cup\tilde{J}^-_{NE}} t_j\bar{d}_j = -t_{j_1}\bar{d}_{j_1}.$$

Thus

$$\sum_{j\in \tilde{J}^+_{NI}} t_j\bar{d}_j + \sum_{j\in \tilde{J}^-_{NI}} t_j\bar{d}_j = -\sum_{j\in \tilde{J}^+_{NE}\cup\tilde{J}^-_{NE}} t_j\bar{d}_j - t_{j_1}\bar{d}_{j_1}.$$

Therefore, α_2 becomes:

$$\begin{aligned}\alpha_2 &= \tfrac{1}{\alpha}\left(\sum_{j\in\tilde{J}_{NE}^+} t_j(\bar{x}_j - l_j) + \sum_{j\in\tilde{J}_{NE}^-} t_j(\bar{x}_j - u_j) + \sum_{j\in\tilde{J}_{NE}^+\cup\tilde{J}_{NE}^-} t_j\bar{d}_j + t_{j_1}\bar{d}_{j_1}\right)\\ &= \tfrac{1}{\alpha}\left(t_{j_1}\bar{d}_{j_1} + \sum_{j\in\tilde{J}_{NE}^+} t_j(\bar{x}_j - l_j + \bar{d}_j) + \sum_{j\in\tilde{J}_{NE}^-} t_j(\bar{x}_j - u_j + \bar{d}_j)\right)\\ &= \tfrac{1}{\alpha}\left(t_{j_1}\bar{d}_{j_1} + \sum_{j\in\tilde{J}_{NE}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{NE}^-} t_j(\kappa_j - u_j)\right).\end{aligned}$$

Then

$$\alpha_1 + \alpha_2 = \frac{1}{\alpha}\left(t_{j_1}\bar{d}_{j_1} + \sum_{j\in\tilde{J}_{N0}^+\cup\tilde{J}_{NE}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{N0}^-\cup\tilde{J}_{NE}^-} t_j(\kappa_j - u_j)\right) = V_0$$

and

$$\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0 V_0.$$

□

Remark 6. In this remark, we will examine the conditions which allow us to get $\beta(\bar{x}, \bar{J}_B) \leq \beta(\bar{x}, J_B)$. We have $\forall j \in \tilde{J}_{N0}^+,\ t_j > 0,\ \kappa_j = \bar{x}_j \geq l_j$ and $\forall j \in \tilde{J}_{N0}^-,\ t_j < 0,\ \kappa_j = \bar{x}_j \leq u_j$. So

$$\alpha_1 = \frac{1}{\alpha}\left[\sum_{j\in\tilde{J}_{N0}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{N0}^-} t_j(\kappa_j - u_j)\right] \geq 0.$$

If there exists an index $j \in \tilde{J}_{N0}$, such that ($t_j > 0$ and $\kappa_j > l_j$) or ($t_j < 0$ and $\kappa_j < u_j$), then $\alpha_1 > 0$. However in this case, by using formulas (28), we get $\sigma^0 = 0$. If $\sigma^0 > 0$, then $\tilde{J}_{N0}^+ \cup \tilde{J}_{N0}^- = \emptyset$, $\alpha_1 = 0$. Thus in all cases we deduce that $\sigma^0\alpha_1 = 0$. Consequently, if $\tilde{J}_{NE}^+ \cup \tilde{J}_{NE}^- = \emptyset$, then $\alpha_2 = \frac{1}{\alpha} t_{j_1}\bar{d}_{j_1}$. So, three cases can occur:
If $\sigma^0 = 0$, then from (35), we get $\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B)$.
If $\sigma^0 > 0$ and $t_{j_1}\bar{d}_{j_1} < 0$, then $\alpha_2 < 0$ and $\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0\alpha_2 < \beta(\bar{x}, J_B)$.
If $\sigma^0 > 0$ and $t_{j_1}\bar{d}_{j_1} \geq 0$, then $\beta(\bar{x}, \bar{J}_B) \geq \beta(\bar{x}, J_B)$.
In the case where $\tilde{J}_{NE}^+ \cup \tilde{J}_{NE}^- \neq \emptyset$, α_2 may be positive and $\beta(\bar{x}, \bar{J}_B)$ may be greater than $\beta(\bar{x}, J_B)$.
Note that the negative effect in this later case will be reduced by the choice of the parameter η with Procedure 3 presented in Subsection 4.4.

Finally, we change the support with the short step rule or the long step rule [28, 37] slightly modified. These rules are described below in Procedures 1 and 2.

Procedure 1. (Short step rule).

1. Calculate $\sigma^0 = \sigma_{j_0} = \min_{j\in J_N} \sigma_j$, where σ_j are determined with (28);
2. Calculate σ_0' and σ_0'' with (29)-(30);
3. Calculate $\bar{\Delta}' = \Delta' + \sigma_0' t$, $\bar{\Delta}'' = \Delta'' + \sigma_0'' t$ and $\bar{\Delta}(\bar{x}) = \Delta(\bar{x}) + \sigma^0 t$;
4. Put $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$;
5. Calculate $\beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0 V_0$;

Procedure 2. (Long step rule)

1. Calculate $\{\sigma_j,\ j \in J_N\}$, where σ_j are determined with relationships (28);
2. Sort the indices $\{i \in J_N:\ \sigma_i \neq \infty\}$:
$$\sigma_{i_1} \leq \sigma_{i_2} \leq \cdots \leq \sigma_{i_p},\ i_k \in J_N,\ \sigma_{i_k} \neq \infty,\ k = 1, \ldots, p;$$
3. If $V_0 \geq 0$ or $p = 1$, then put $j_0 = i_1$ and go to step **7**;
4. For all $i_k,\ k = 1, \ldots, p$, calculate $\Delta V_{i_k} = \frac{1}{\alpha}|t_{i_k}|\,(u_{i_k} - l_{i_k})$;
5. Calculate $V_{i_k},\ k = 0, \ldots, p$, where
$$\begin{cases} V_{i_0} = V_0, \\ V_{i_k} = V_0 + \sum_{s=1}^{k} \Delta V_{i_s} = V_{i_{k-1}} + \Delta V_{i_k},\ k = 1, \ldots, p; \end{cases}$$
6. Choose the index $j_0 = i_q$, such that $V_{i_{q-1}} < 0$ and $V_{i_q} \geq 0$;
7. Calculate σ_0' and σ_0'' with formulas (29) and (30); put $\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}$, $\sigma^0 = \sigma_{j_0} = \sigma_{i_q}$, $\bar{\Delta}' = \Delta' + \sigma_0' t$, $\bar{\Delta}'' = \Delta'' + \sigma_0'' t$ and $\bar{\Delta}(\bar{x}) = \Delta(\bar{x}) + \sigma^0 t$;
8. Calculate $\beta(\bar{x}, \bar{J}_B)$ with (32);

4.2. Algorithm of the hybrid direction method

Let $\{x, J_B\}$ be an initial SFS for problem (1)-(2) and $\epsilon \geq 0$, $\eta > 0$. The algorithm of the Hybrid Direction Method (HDM) is described in the following steps:

Algorithm 2. (HDM)

1. Calculate $F(x)$ and
$$\pi_P^T = p_B^T A_B^{-1},\ \Delta_j' = \pi_P^T a_j - p_j,\ j \in J_N,\ \pi_Q^T = q_B^T A_B^{-1},$$
$$\Delta_j'' = \pi_Q^T a_j - q_j,\ j \in J_N,\ \Delta_j(x) = \Delta_j' - F(x)\,\Delta_j'',\ \ j \in J_N;$$
2. Calculate the suboptimality estimate β with (20);
3. If $\beta = 0$, then the algorithm stops with the optimal SFS $\{x, J_B\}$;
4. If $\beta \leq \epsilon$, then the algorithm stops with the ϵ-optimal SFS $\{x, J_B\}$;
5. Calculate $\eta(x - l)$ and $\eta(x - u)$; the index sets J_{NE}^+ and J_{NE}^- with (17), and μ with (19);
6. Calculate the index sets J_{NI}^+, J_{NI}^- and J_{NR} with (17);
7. Calculate the direction d using relationships (21);
8. Calculate θ_{j_1} , θ_{j_2} and the steplength θ^0 with (22);
9. Calculate $\bar{x} = x + \theta d$ and $F(\bar{x}) = F(x) + \frac{\alpha\theta^0(\beta+\mu)}{Q(\bar{x})}$;
10. Calculate
$$\Delta(\bar{x}) = \Delta' - F(\bar{x})\Delta'',\ \bar{\beta} = \frac{1}{\alpha}\left(\sum_{j \in \tilde{J}_N^+} \Delta_j(\bar{x})(\bar{x}_j - l_j) + \sum_{j \in \tilde{J}_N^-} \Delta_j(\bar{x})(\bar{x}_j - u_j)\right);$$

11. If $\bar{\beta} = 0$, then the algorithm stops with the optimal SFS $\{\bar{x}, J_B\}$;

12. If $\bar{\beta} \leq \epsilon$, then the algorithm stops with the ϵ-optimal SFS $\{\bar{x}, J_B\}$;

13. If $\theta^0 = \theta_{j_2} \vee 1$, then put $x := \bar{x}$, $F(x) := F(\bar{x})$, $\beta := \bar{\beta}$ and go to step **5**;

14. Change of the support J_B by $\bar{J}_B$:

14.1 Calculate the new vectors $\eta(\bar{x} - l)$, $\eta(\bar{x} - u)$ and the new index sets $\tilde{J}^+_{NE}$, $\tilde{J}^-_{NE}$;

14.2 Calculate the new index sets $\tilde{J}^+_{NI}$, $\tilde{J}^-_{NI}$ and $\tilde{J}_{NR}$;

14.3 Calculate the vector $\bar{d}$ with (26), the vectors $\kappa = \bar{x} + \bar{d}$ and the vector $t = (t_j,\ j \in J)$:

$$t_{j_1} = -\text{sign}\, d_{j_1};\ t_j = 0,\ j \neq j_1,\ j \in J_B;\ t_N^T = t_B^T A_B^{-1} A_N;$$

14.4 Calculate V_0 with (34);

14.5 Calculate the new support $\bar{J}_B$, the new reduced costs vectors $\bar{\Delta}'$, $\bar{\Delta}''$, $\bar{\Delta}(\bar{x})$ and the new suboptimality estimate $\bar{\bar{\beta}} = \beta(\bar{x}, \bar{J}_B)$ with the short step rule (Procedure 1) or the long step rule (Procedure 2) and go to step **15**;

15. Put $x := \bar{x}$, $F(x) := F(\bar{x})$, $J_B := \bar{J}_B$, $\Delta' := \bar{\Delta}'$, $\Delta'' := \bar{\Delta}''$, $\Delta(x) := \bar{\Delta}(\bar{x})$, $\beta := \bar{\bar{\beta}}$ and go to step **3**;

Remark 7. If $\theta^0 > 0$ and $\sigma^0 > 0$ in each iteration of HDM, then we will have a new different SFS $\{\bar{x}, \bar{J}_B\}$, with $\bar{J}_B \neq J_B$ and $F(\bar{x}) > F(x)$. Since the number of the supports of problem (1)-(2) is finite (less or equal to C_n^m), so the algorithm will stop in a finite number of iterations.

Remark 8. When $\eta \to +\infty$, we will have $J^+_{NE} = J^-_{NE} = \emptyset$. Hence

$$J_{NE} = \emptyset,\ J^+_{NI} = J^+_N = \{j \in J_N :\ \Delta_j(x) > 0\},\ J^-_{NI} = J^-_N = \{j \in J_N :\ \Delta_j(x) < 0\}$$

and

$$\lim_{\eta \to +\infty} \gamma(\eta, x, J_B) = \beta(x, J_B) \text{ and } \lim_{\eta \to +\infty} \mu(\eta, x, J_B) = 0.$$

Thus, the hybrid direction will be equal to the standard direction of the adaptive method:

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

When η is set to $+\infty$ and $Q(x) = 1$ in HDM for LFP, then we will find exactly the adaptive method of linear programming [28].

4.3. Numerical example

Let us solve the following problem with HDM using the long step rule:

$$\max\ F(x) = \frac{P(x)}{Q(x)} = \frac{5x_1 + x_2 + 10}{4x_1 + 2x_2 + 12},$$

subject to

$$\begin{array}{lllllll} 5x_1 & +x_2 & +x_3 & & = & 20, \\ 4x_1 & & -x_3 & +x_4 & = & 14, \end{array}$$

$$\begin{array}{ccccc} 2 & \leq & x_1 & \leq & 5, \\ 4 & \leq & x_2 & \leq & 12, \\ 0 & \leq & x_3 & \leq & 25, \\ 0 & \leq & x_4 & \leq & 18. \end{array}$$

Let $\{x, J_B\}$, with $x = (2, 10, 0, 6)^T$ and $J_B = \{1, 2\}$, an initial support feasible solution and $F(x) = \frac{3}{4} = 0.75$ the value of the objective function at this solution. We set $\eta = 1$ and $\epsilon = 0$.
The minimum of the function Q over the feasible set S is $\alpha = \min_{x \in S} Q(x) = 28$.
First iteration:
We have

$$J_B = \{1, 2\}\,,\ J_N = \{3, 4\}\,,\ p_B^T = (5, 1),\ p_N^T = (0, 0),\ q_B^T = (4, 2),\ q_N^T = (0, 0),$$

$$A_B = \begin{pmatrix} 5 & 1 \\ 4 & 0 \end{pmatrix},\ A_B^{-1} = \begin{pmatrix} 0 & 1/4 \\ 1 & -5/4 \end{pmatrix},\ A_N = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}.$$

The multipliers and the reduced costs vectors are:

$$\pi_P^T = p_B^T A_B^{-1} = (1, 0),\ \Delta'_N = A_N^T \pi_P - p_N = (1, 0)^T,\ \Delta' = (0, 0, 1, 0)^T,$$

$$\pi_Q^T = q_B^T A_B^{-1} = (2, -3/2),\ \Delta''_N = A_N^T \pi_Q - q_N = (7/2, -3/2)^T,$$

$$\Delta'' = (0, 0, 7/2, -3/2)^T,\ \Delta_N(x) = \Delta'_N - F(x)\Delta''_N = (-13/8, 9/8)^T,$$

$$\Delta(x) = (0, 0, -13/8, 9/8)^T.$$

We have $J_N^+ = \{4\},\ J_N^- = \{3\}$, so the suboptimality estimate is:

$$\beta(x, J_B) = \frac{\Delta_4(x)(x_4 - l_4) + \Delta_3(x)(x_3 - u_3)}{\alpha} = \frac{379}{224} \simeq 1.6920 > \epsilon.$$

The vectors $\eta(x - u)$ and $\eta(x - l)$ are:

$$\eta(x - u) = (-3, -2, -25, -12)^T,\ \ \eta(x - l) = (0, 6, 0, 6)^T.$$

The index sets are:

$$J_{NI}^+ = \{4\},\ J_{NI}^- = \{3\},\ J_{NE}^+ = J_{NE}^- = \emptyset \Rightarrow \mu = 0.$$

The direction d is computed as follows:
We have $3 \in J_{NI}^- \Rightarrow d_3 = u_3 - x_3 = 25$; $4 \in J_{NI}^+ \Rightarrow d_4 = l_4 - x_4 = -6$. Hence

$$d_N = \begin{pmatrix} 25 \\ -6 \end{pmatrix},\ d_B = -A_B^{-1} A_N d_N = \begin{pmatrix} 31/4 \\ -255/4 \end{pmatrix},\ d = \begin{pmatrix} 31/4 \\ -255/4 \\ 25 \\ -6 \end{pmatrix}.$$

The steplength θ^0 along the direction d is computed as follows:

$$\theta_{j_1} = \min\{\theta_1, \theta_2\} = \min\left\{\frac{u_1 - x_1}{d_1}, \frac{l_2 - x_2}{d_2}\right\} = \min\left\{\frac{12}{31}, \frac{8}{85}\right\} = \frac{8}{85} = \theta_2,$$

$$\theta_{j_2} = \infty,\ \theta^0 = \min\{\theta_{j_1}, \theta_{j_2}, 1\} = \min\{8/85, \infty, 1\} = 8/85 = \theta_2 = \theta_{j_1} \Rightarrow j_1 = 2.$$

The new feasible solution $\bar{x}$ is:

$$\bar{x} = x + \theta^0 d = \begin{pmatrix} 2 \\ 10 \\ 0 \\ 6 \end{pmatrix} + \frac{8}{85}\begin{pmatrix} 31/4 \\ -255/4 \\ 25 \\ -6 \end{pmatrix} = \begin{pmatrix} 232/85 \\ 4 \\ 40/17 \\ 462/85 \end{pmatrix}.$$

The new value of the objective function is:

$$Q(\bar{x}) = \frac{2628}{85},\ \ F(\bar{x}) = F(x) + \frac{\alpha\theta^0(\beta + \mu)}{Q(\bar{x})} = \frac{1175}{1314} \simeq 0.8942 > F(x) = 0.75.$$

The new reduced costs vector $\Delta(\bar{x})$ and the new suboptimality estimate $\bar{\beta}$:

$$\Delta(\bar{x}) = \Delta' - F(\bar{x})\Delta'' = (0, 0, -1231/578, 1175/876)^T,\ \tilde{J}_N^+ = \{4\},\ \tilde{J}_N^- = \{3\},$$

$$\bar{\beta} = \frac{\Delta_4(\bar{x})(\bar{x}_4 - l_4) + \Delta_3(\bar{x})(\bar{x}_3 - u_3)}{\alpha} = \frac{1281}{646} \simeq 1.9830 > \epsilon.$$

Change of the support J_B by $\bar{J}_B$:
We have:

$$\tilde{J}_{NE}^+ = \tilde{J}_{NE}^- = \emptyset,\ \tilde{J}_{NI}^+ = \{4\},\ \tilde{J}_{NI}^- = \{3\}.$$

The vector $\bar{d}$ is:
$\bar{d}_3 = u_3 - \bar{x}_3 = 25 - 40/17 = 385/17$; $\bar{d}_4 = l_4 - \bar{x}_4 = 0 - 462/85 = -462/85$. Hence

$$\bar{d}_N = \begin{pmatrix} 385/17 \\ -462/85 \end{pmatrix},\ \bar{d}_B = -A_B^{-1}A_N\bar{d}_N = \begin{pmatrix} 2387/340 \\ -231/4 \end{pmatrix},\ \bar{d} = \begin{pmatrix} 2387/340 \\ -231/4 \\ 385/17 \\ -462/85 \end{pmatrix}.$$

The vector κ is: $\kappa = \bar{x} + \bar{d} = (39/4, -215/4, 25, 0)^T$.
The vector t is:

$$\begin{aligned} t_B^T &= (t_1, t_2) = (0, -\text{sign } d_2) = (0, 1); \\ t_N^T &= (t_3, t_4) = t_B^T A_B^{-1} A_N = (9/4, -5/4); \\ t &= (0, 1, 9/4, -5/4)^T. \end{aligned}$$

Hence $\tilde{J}_{N0}^+ = \tilde{J}_{N0}^- = \emptyset$ and

$$\alpha_1 = \frac{1}{\alpha}\left(\sum_{j\in\tilde{J}_{N0}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{N0}^-} t_j(\kappa_j - u_j)\right) = 0.$$

$$\alpha_2 = \frac{1}{\alpha}\left(t_{j_1}\bar{d}_{j_1} + \sum_{j\in\tilde{J}_{NE}^+} t_j(\kappa_j - l_j) + \sum_{j\in\tilde{J}_{NE}^-} t_j(\kappa_j - u_j)\right) = \frac{\bar{d}_2}{\alpha} = -33/16.$$

Thus, $V_0 = -33/16 < 0$. The steplengths $\sigma_j,\ j \in J_N$ are:

$$\Delta_3(\bar{x}) < 0,\ t_3 = 9/4 > 0 \Rightarrow \sigma_3 = -\Delta_3(\bar{x})/t_3 = 1169/1235;$$

$$\Delta_4(\bar{x}) > 0,\ t_4 = -5/4 < 0 \Rightarrow \sigma_4 = -\Delta_4(\bar{x})/t_4 = 235/219.$$

We have $\sigma_3 < \sigma_4 \Rightarrow i_1 = 3,\ i_2 = 4$, and $p = 2$. Then

$$\begin{cases} \Delta V_{i_1} = \Delta V_3 = \frac{1}{\alpha}\left(|t_3|\,(u_3 - l_3)\right) = 225/112, \\ \Delta V_{i_2} = \Delta V_4 = \frac{1}{\alpha}\left(|t_4|\,(u_4 - l_4)\right) = 45/56. \end{cases}$$

Hence

$$\begin{cases} V_{i_0} = V_0 = -33/16 < 0, \\ V_{i_1} = V_3 = V_{i_0} + \Delta V_{i_1} = -3/56 < 0, \\ V_{i_2} = V_4 = V_{i_1} + \Delta V_{i_2} = 3/4 \geq 0. \end{cases}$$

We have $V_{i_1} = -3/56 < 0$ and $V_{i_2} = 3/4 > 0$. So

$$j_0 = i_q = i_2 = 4,\ \sigma^0 = \sigma_{j_0} = \sigma_4 = 235/219.$$

Therefore, the new support is:

$$\bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\} = \{1, 4\}\,,\ \bar{J}_N = \{2, 3\}\,.$$

The new reduced costs vectors are:

$$\sigma_0' = -\frac{\Delta_{j_0}'}{t_{j_0}} = 0, \sigma_0'' = -\frac{\Delta_{j_0}''}{t_{j_0}} = -6/5;$$

$$\bar{\Delta}' = \Delta' + \sigma_0' t = (0, 0, 1, 0)^T,\ \bar{\Delta}'' = \Delta'' + \sigma_0'' t = (0, -6/5, 4/5, 0)^T;$$

$$\bar{\Delta}(\bar{x}) = \Delta(\bar{x}) + \sigma^0 t = (0, 235/219, 187/657, 0)^T.$$

The new suboptimality estimate is:

$$\beta(\bar{x}, \bar{J}_B) = \frac{\bar{\Delta}_2(\bar{x})(\bar{x}_2 - l_2) + \bar{\Delta}_3(\bar{x})(\bar{x}_3 - l_3)}{\alpha} = \frac{110}{4599} \simeq 0.0239 > \epsilon.$$

Remark 9. If we use the relationships given in Remark 4 for computing the multipliers and the reduced costs vectors corresponding to the new SFS $\{\bar{x}, \bar{J}_B\}$, we obtain results which are equal to the ones found with the updating formulas (27). Indeed, we have

$$\bar{x} = (232/85, 4, 40/17, 462/85)^T,\ F(\bar{x}) = \frac{1175}{1314},\ \ \bar{J}_B = \{1, 4\}\,,\ \bar{J}_N = \{2, 3\}\,,$$

$$p_{\bar{B}}^T = (5, 0),\ \ p_{\bar{N}}^T = (1, 0),\ \ q_{\bar{B}}^T = (4, 0),\ \ q_{\bar{N}}^T = (2, 0),$$

$$A_{\bar{B}} = \begin{pmatrix} 5 & 0 \\ 4 & 1 \end{pmatrix},\ \ A_{\bar{B}}^{-1} = \begin{pmatrix} 1/5 & 0 \\ -4/5 & 1 \end{pmatrix},\ \ A_{\bar{N}} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix},$$

$$\bar{\pi}_P^T = p_{\bar{B}}^T A_{\bar{B}}^{-1} = (1, 0),\ \ \bar{\Delta}_N' = A_{\bar{N}}^T \bar{\pi}_P - p_{\bar{N}} = (0, 1)^T,\ \bar{\Delta}' = (0, 0, 1, 0)^T,$$

$$\bar{\pi}_Q^T = q_{\bar{B}}^T A_{\bar{B}}^{-1} = (4/5, 0),\ \ \bar{\Delta}_N'' = A_{\bar{N}}^T \bar{\pi}_Q - q_{\bar{N}} = (-6/5, 4/5)^T,\ \bar{\Delta}'' = (0, -6/5, 4/5, 0)^T,$$

$$\bar{\Delta}_N(\bar{x}) = \bar{\Delta}_N' - F(\bar{x})\bar{\Delta}_N'' = (235/219, 187/657)^T\,,\ \bar{\Delta}(\bar{x}) = (0, 235/219, 187/657, 0)^T.$$

Second iteration:
We have

$$x = (232/85, 4, 40/17, 462/85)^T,\ F(x) = \frac{1175}{1314},\ J_B = \{1, 4\}\,,\ J_N = \{2, 3\}\,,\ \beta = \frac{110}{4599},$$

$$\Delta' = (0, 0, 1, 0)^T,\ \Delta'' = (0, -6/5, 4/5, 0)^T,\ \Delta(x) = (0, 235/219, 187/657, 0)^T.$$

The vectors $\eta(x - u)$ and $\eta(x - l)$:

$$\eta(x - u) = (-193/85, -8, -385/17, -1068/85)^T,\ \ \eta(x - l) = (62/85, 0, 40/17, 462/85)^T.$$

The index sets are:

$$J_{NI}^+ = \{3\},\ J_{NI}^- = \emptyset,\ J_{NE}^+ = J_{NE}^- = \emptyset \Rightarrow \mu = 0.$$

The ascent direction d is:

Since $2 \in J_{NR}^{+}$ and $3 \in J_{NI}^{+}$, then $d_2 = 0$ and $d_3 = l_3 - x_3 = -40/17$.

$$d_N = \begin{pmatrix} 0 \\ -40/17 \end{pmatrix}, \quad d_B = -A_B^{-1} A_N d_N = \begin{pmatrix} 8/17 \\ -72/17 \end{pmatrix}, \quad d = \begin{pmatrix} 8/17 \\ 0 \\ -40/17 \\ -72/17 \end{pmatrix}.$$

The steplength θ^0 is:

$$\theta_{j_1} = \min\{\theta_1, \theta_4\} = \min\left\{\frac{u_1 - x_1}{d_1}, \frac{l_4 - x_4}{d_4}\right\} = \min\left\{\frac{193}{40}, \frac{77}{60}\right\} = \frac{77}{60} = \theta_4,$$

$$\theta_{j_2} = \infty, \ (J_{NE} = \emptyset), \ \theta^0 = \min\{\theta_{j_1}, \theta_{j_2}, 1\} = \min\{77/60, \infty, 1\} = 1.$$

The feasible solution $\bar{x}$ is:

$$\bar{x} = x + \theta^0 d = \begin{pmatrix} 232/85 \\ 4 \\ 40/17 \\ 462/85 \end{pmatrix} + 1 \begin{pmatrix} 8/17 \\ 0 \\ -40/17 \\ -72/17 \end{pmatrix} = \begin{pmatrix} 16/5 \\ 4 \\ 0 \\ 6/5 \end{pmatrix}.$$

The new value of the objective function is:

$$Q(\bar{x}) = \frac{164}{5}, \quad F(\bar{x}) = F(x) + \frac{\alpha\theta^0(\beta + \mu)}{Q(\bar{x})} = \frac{75}{82} \simeq 0.9146 > F(x) = 0.8942.$$

The new reduced costs vector and the new suboptimality estimate are:

$$\Delta(\bar{x}) = (0, 45/41, 11/41, 0)^T, \ \bar{\beta} = \frac{\Delta_2(\bar{x})(\bar{x}_2 - l_2) + \Delta_3(\bar{x})(\bar{x}_3 - l_3)}{\alpha} = 0.$$

Therefore, the vector $x^* = \left(\frac{16}{5}, 4, 0, \frac{6}{5}\right)^T$, with $F(x^*) = \frac{75}{82}$, is optimal for the considered LFP problem.

4.4. Choice of the parameter η

The experimental study carried out in this work has shown that it is judicious to choose the parameter η so as to force the sets J_{NE}^{+}, J_{NE}^{-}, $\tilde{J}_{NE}^{+}$ and $\tilde{J}_{NE}^{-}$ to be empty at each iteration. Thus, in order to improve the efficiency of Algorithm 2, we first start by setting η to an initial positive fixed value, for example $\eta = 1$, and if the SFS $\{x, J_B\}$ is not optimal or ϵ-optimal, then we calculate the index sets J_{NE}^{+} and J_{NE}^{-}. If $J_{NE}^{+} \cup J_{NE}^{-} \neq \emptyset$, then we update the value of η with Procedure 3. Furthermore, we compute the index sets $\tilde{J}_{NE}^{+}$ and $\tilde{J}_{NE}^{-}$; if they are not empty, then we set $x := \bar{x}$ and update an other time η with Procedure 3.

Procedure 3.

(1) If $J_{NE}^{+} \neq \emptyset$, then put $\eta_0 = \max\limits_{j \in J_{NE}^{+}} \frac{\Delta_j(x)}{x_j - l_j}$, else put $\eta_0 = 0$;

(2) If $J_{NE}^{-} \neq \emptyset$, then put $\eta_1 = \max\limits_{j \in J_{NE}^{-}} \frac{\Delta_j(x)}{x_j - u_j}$, else put $\eta_1 = 0$;

(3) Put $\eta := max\{\eta_0, \eta_1\} + 1$.

Thus, for the new value of η computed with Procedure 3 as explained above, we will have in each iteration $J_{NE}^{+} \cup J_{NE}^{-} = \emptyset$ and $\tilde{J}_{NE}^{+} \cup \tilde{J}_{NE}^{-} = \emptyset$, $\mu = 0$, $\gamma = \beta$, $\theta_{j_2} = \infty$ and $\theta^0 = \min\{\theta_{j_1}, 1\}$, etc. This will allow to simplify calculus and reduce computational effort in each iteration. Moreover, by using Procedure 3, we may increase the probability to get a better suboptimality estimate $\beta(\bar{x}, \bar{J}_B)$ by eliminating the quantity depending on the sets $\tilde{J}_{NE}^{+}$ and $\tilde{J}_{NE}^{-}$, which can be positive. So, we modify Algorithm 2 (HDM), as follows: after the step 5, we

introduce the step 5a and after the step 14.1, we add the step 14.1a:
(5a) If $J_{NE}^{+} \cup J_{NE}^{-} \neq \emptyset$, then update η with Procedure 3 and go to step 5, else go to step 6.
(14.1a) If $\tilde{J}_{NE}^{+} \cup \tilde{J}_{NE}^{-} \neq \emptyset$, then set $x := \bar{x}$, update η with Procedure 3 and go to step 14.1, else go to step 14.2.
Let us call Algorithm 2, HDM with a Fixed parameter (HDMF) and HDM with the steps 5a and 14.1a, HDM with a Variable parameter (HDMV).
The practical advantages of the modified version of HDM (HDMV) will be discussed in the experimental study presented in the following section.

5. Numerical experiments

In order to perform a numerical comparison between the hybrid direction method (HDM), the Primal Simplex Method (PSM) and the Interior-Points Method (IPM) implemented in Matlab, we have developed an implementation of HDM and PSM with the MATLABR2018a programming language. In the implementation of the two algorithms HDM and PSM, the basis matrix inverse is updated in each iteration using the product form of the inverse [20, 44], and in order to maintain numerical stability we reinvert the basis matrix from scratch every 100 iterations. For the IPM algorithm, we have set the parameters of the Matlab "fmincon" function as follows: the initial point "$x_0 = (lb + ub)/2$", "Algorithm=interior-point", "MaxIter=100" and "MaxFunEvals=1e6". Then we have run the different solvers on a machine with 8 GB of RAM and a microprocessor Intel(R) Core(TM) i5-8250U 1.60GHz working under the Windows 10 operating system.

The numerical study is carried out on 120 randomly generated LFP test problems with bounded variables. These test problems are written in the following form:

$$\max F(x) = \frac{P(x)}{Q(x)} = \frac{p^T x + p_0}{q^T x + q_0},$$

$$\text{subject to } Ax + x^e = b,\ l \leq x \leq u,\ l^e \leq x^e \leq u^e,$$

where p, q, x, b and x^e are vectors in $\mathbb{R}^n$; A a square matrix of order n, p_0 and q_0 are two real numbers.

The components of the vectors p, q, l and the entries of the matrix A are randomly generated following the uniform distribution: $-100 \leq p_j \leq 100$, $0 \leq q_j \leq 100$, $0 \leq a_{ij} \leq 100$, $0 \leq l_j \leq 100$. In order to have $Q(x) > 0$, we have generated $q \geq 0$ and $q_0 > 0$. The vectors b, u, l^e and u^e are constructed as follows: $b = Al + r_1$, $u = l + r_2$, $l^e = b - Al - r_3$, $u^e = b - Al + r_4$, where r_k, $k = 1, 2, 3, 4$, are n-vectors randomly generated following the uniform distribution, such that $1 \leq r_{kj} \leq 100$, $j = 1, 2, \dots, n$, and r_{kj} represents the jth component of the vector r_k.

We have initialized PSM and HDM with the basic feasible solution $\{x, J_B\}$, where $x = (l, b - Al)$ and $J_B = \{n+1, n+2, \dots, 2n\}$, and we have set the initial value of η to 1 and $\epsilon = 10^{-10}$ for HDM. In this study, we have generated problems with a constraint matrix having a density $d = 10\%$ (the number of non zero entries$\times 100/n^2$) and we have considered two variants of the hybrid direction method: HDM with a Variable parameter and the Short step rule (HDMVS) and HDM with a Variable parameter and the Long step rule (HDMVL).

For each algorithm and for each problem size n, we report in Table 1 the average number of iterations (NIT) and the average CPU time (CPU) necessary for solving the 10 generated problems with the same size. ERR designates the average absolute value of the gap between the optimal value found by the simplex algorithm and the maximum found by the considered algorithm. For each algorithm, we plot the average number of iterations and the average CPU time in terms of the problem size n. The obtained graphs are shown in Figure 1.

From Table 1 and Figure 1, we remark that HDMVL is more efficient than HDMVS and it outperforms the primal simplex algorithm and the interior-points method implemented in Matlab. The superiority of HDMVL over PSM and IPM becomes more clear with the increase of the problem dimension. However, HDM is less efficient than the primal simplex method when we use the short step rule for changing the current support. Notice also that the experimental study has shown that the choice of the parameter η with Procedure 3 presents the following practical advantages:

n	PSM		HDML			HDMS			IPM		
	CPU	NIT	CPU	NIT	ERR	CPU	NIT	ERR	CPU	NIT	ERR
100	0.02	37.6	0.04	44.4	8.4E-07	0.05	75.7	8.4E-07	1.36	100	1.0E-03
200	0.05	90.7	0.07	88.9	6.7E-07	0.14	183.1	6.7E-07	5.64	100	4.5E-04
300	0.20	167.0	0.19	119.6	7.8E-08	0.42	272.5	7.8E-08	15.02	100	2.5E-04
400	0.75	231.9	0.68	161.3	2.3E-07	1.62	405.1	2.3E-07	31.90	100	1.8E-04
500	1.73	333.2	1.27	202.8	1.8E-08	3.59	562.3	1.8E-08	57.19	100	1.3E-04
600	3.67	437.6	2.32	230.4	1.4E-07	6.89	687.7	1.4E-07	91.43	100	9.4E-05
700	7.34	589.6	3.87	266.5	3.8E-09	12.47	796.2	3.8E-09	134.26	100	6.5E-05
800	12.21	633.0	6.61	297.6	9.0E-08	20.00	896.6	9.0E-08	192.10	100	8.5E-05
900	17.52	692.9	9.14	306.0	6.1E-10	30.97	1035.9	6.1E-10	263.28	100	7.3E-05
1000	30.78	877.1	13.96	345.0	8.9E-10	48.13	1193.1	8.9E-10	346.69	100	4.9E-05
1200	54.38	977.1	24.71	398.5	7.6E-17	94.56	1513.7	5.3E-17	554.85	100	5.6E-05
1400	117.27	1364.1	41.95	450.7	9.2E-17	164.28	1782.4	9.9E-17	855.62	100	4.6E-05

Table 1. The average number of iterations and the average CPU time in terms of n

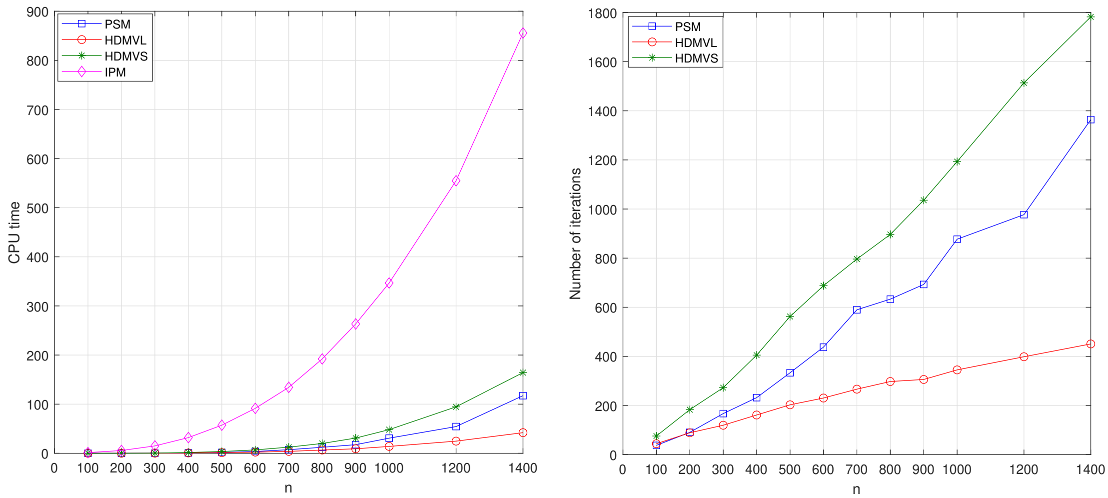

Figure 1. Graphs of CPU and NIT in terms of n

1. The degeneracy is highly reduced and the number of iterations is reduced too. Indeed, in all the iterations and for all the test problems, we have obtained a primal steplength $\theta^0 > 0$ and a strict increase of the objective function: $F(\bar{x}) > F(x)$.

2. The obtained quantity V_0 at each iteration of HDMVL for all the considered test problems is negative and $\beta(\bar{x}, \bar{J}_B) < \beta(\bar{x}, J_B)$. This leads to a better suboptimality estimate when we pass from the support J_B to $\bar{J}_B$.

3. We have also noticed that for all the iterations of HDMVL and for all the considered test problems the new suboptimality estimate $\beta(\bar{x}, \bar{J}_B)$ is better than (less than) $\beta(x, J_B)$. In the left side of Figure 2, we represent graphically the variation of the suboptimality estimate in terms of the iterations number for the third test problem of size 100.

In the other hand, the variant HDMFL (HDM with a fixed parameter η and the long step rule, i.e., Procedure 3 is not used to change η in each iteration) suffers from degeneracy. Indeed, for the majority of test problems, we have a big number of iterations with $\theta^0 = 0,\ \bar{x} = x$ and $F(\bar{x}) = F(x)$; this prevents the objective function to progress,

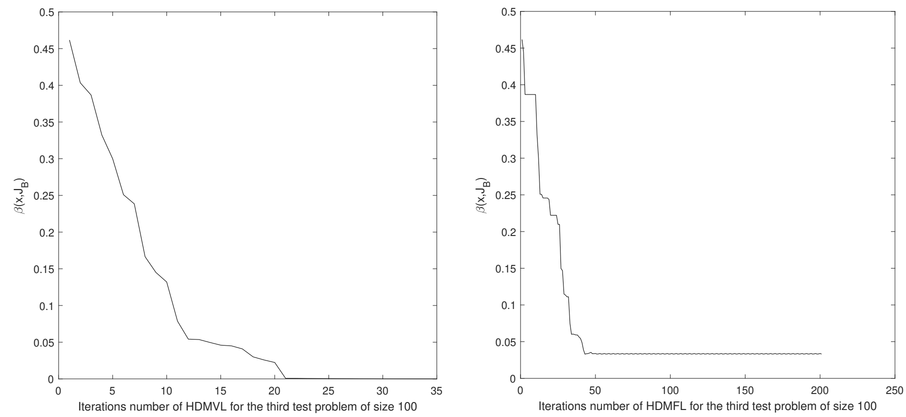

Figure 2. Variation of $\beta(x, J_B)$ in terms of the iterations number of HDMVL and HDMFL

although an optimal solution or an ϵ-optimal is not yet achieved. Furthermore for some iterations, the quantity V_0 is positive and we have $\beta(\bar{x}, \bar{J}_B) \geq \beta(\bar{x}, J_B)$.

For example, when we apply HDMFL for the first and the second test problems of size $n = 100$, we find the optimal values $F(x^*) = -0.1190$ (in 69 iterations) and $F(x^*) = -0.0308$ (in 85 iterations), respectively. However for the third test problem of size 100 (see the graph in the right side of Figure 2), after 51 iterations, the objective function stays without further progress in the value 0.18191775 until we interrupt the execution in iteration 200, with $\beta(x, J_B) = 0.0331$. However, the optimal values of the first, second and the third test problems are found by HDMVL without degeneracy in 56, 48 and 35 iterations, respectively.

6. Conclusion

In this article, we have proposed a new method for solving linear fractional programming problems with bounded variables. This method is a generalization of the hybrid direction method developed recently in [22]. The experimental study carried out on solving 120 randomly generated test problems has shown the superiority of the proposed algorithm with the long step rule over the primal simplex algorithm and the interior-points method implemented in MATLABR2018a. In future works, we will study the efficiency of HDM on solving practical test problems which arise in different real-world applications of linear fractional programming. Furthermore, we intend to generalize the proposed algorithm for the resolution of quadratic fractional programming problems.

Acknowledgement

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

REFERENCES

1. A. Andjouh, and M. O. Bibi, *Adaptive global algorithm for solving box-constrained non-convex quadratic minimization problems*, J Optim Theory Appl, vol. 192, no. 1, pp. 360–378, 2022.
2. M. Azi, and M. O. Bibi, *Optimal control of a dynamical system with intermediate phase constraints and applications in cash management*, Numerical Algebra, Control and Optimization, vol. 12, no. 2, pp. 279–291, 2022.
3. E. B. Bajalinov, *On the economic sense of dual Variables in linear-Fractional Programming*, Ekonomika i matematicheskie metody, vol. 24, pp. 558–561, 1988 (in Russian).

4. J. Awerjcewicz, *Linear-Fractional Programming: Theory, Methods, Applications and Software*, Applied optimization, Springer, University of Florida, USA, 2003.
5. M. Bentobache, and M. O. Bibi, *A two-phase support method for solving linear programs: Numerical experiments*, Mathematical Problems in Engineering, 28 pages, Article ID 482193, doi:10.1155/2012/482193, 2012.
6. M. Bentobache, *On mathematical methods of linear and quadratic probgramming*, PhD Dissertation, University of Bejaia, Bejaia, Algeria, 2013 (in French).
7. M. Bentobache, and M. O. Bibi, *A hybrid direction algorithm with long step rule for linear programming: Numerical experiments*, In: Le Thi, H., Pham Dinh, T., Nguyen, N. (eds) Modelling, Computation and Optimization in Information Systems and Management Sciences. Advances in Intelligent Systems and Computing, Springer, Cham, vol. 359, pp. 333–344, 2015.
8. M. Bentobache, M. Telli, and A. Mokhtari *New LP-based local and global algorithms for continuous and mixed-integer nonconvex quadratic programming*, Journal of Global Optimization, vol. 82, no. 4, pp. 659–689, 2022.
9. M. Bentobache, and M. O. Bibi, *Méthodes Numériques de la Programmation Linéaire et Quadratique: Théorie et Algorithmes*, Presses Académiques Francophones, Allemagne, 2016.
10. M. O. Bibi, *Optimization of a linear dynamic system with double terminal constraints on the trajectories*, Optimization, vol. 30, no. 4, pp. 359–366, 1994.
11. M. O. Bibi, *Support method for solving a linear-quadratic problem with polyhedral constraints on control*, Optimization, vol. 37, no. 2, pp. 139–147, 1996.
12. M. O. Bibi, and M. Bentobache, *The adaptive method with hybrid direction for solving linear programming problems with bounded variables*, Proceedings of Colloque sur l'Optimisation et les Systèmes d'Information, COSI'2011, pp. 80–91, University of Guelma, Algeria, 24-28 April 2011.
13. M. O. Bibi, and M. Bentobache, *A hybrid direction algorithm for solving linear programs*, International Journal of Computer Mathematics, vol. 92, no. 2, pp. 200–216, 2015.
14. M. O. Bibi, N. Ikheneche, and M. Bentobache, *A hybrid direction algorithm for solving a convex quadratic problem*, International Journal of Mathematics in Operational Research, vol. 16, no. 2, pp. 159–178, 2020.
15. M. O. Bibi, H. Boussouira, and A. Laouar, *Solution of an integer linear programming problem via a primal dual method combined with a heuristic*, International Journal of Mathematical Modelling and Numerical Optimisation, vol. 12, no. 1, pp. 69–87, 2022.
16. H. Boussouira, and M. O. Bibi, *Solving mixed integer linear programming problems with bounded variables by the adaptive method*, To appear in International Journal of Mathematics in Operational Research, doi: 10.1504/IJMOR.2024.10063628, 2025.
17. B. Brahmi, and M. O. Bibi, *Dual support method for solving convex quadratic programs*, Optimization, vol. 59, no. 6, pp. 851–872, 2010.
18. A. Charnes, and W. W. Cooper, *Programming with Linear Fractional Functionals*, Naval Res. Logistics Quart, vol. 9, no. 3-4, pp. l81–186, 1962.
19. A. Charnes, W. W. Cooper, and R. M. Thrall, *A structure for classifying and characterizing efficiency and inefficiency in data envelopment analysis*, The Journal of Productivity Analysis, vol. 2, pp. 197–237, 1991.
20. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.
21. Q. F. Dar, T. R. Padi, and A. M. Tali, *Mixed input and output orientations of Data Envelopment Analysis with Linear Fractional Programming and Least Distance Measures*, Stat., Optim. Inf. Comput., vol. 4, pp 326–341, 2016.
22. K. Djeloud, M. Bentobache, and M. O. Bibi, *A new method with hybrid direction for linear programming*, Concurrency and Computation, Practice and Experience, vol. 33, no. 1, pp. doi: 10.1002/cpe.5836, 2021.
23. W. Dinkelbach, *Die maximierung eines quotienten zweier linearer funklionen unter linearen nebenbedingungen*, Wahrscheinlichkeitstheorie, vol. 1, no. 2, pp. 141–145, 1962.
24. L. F. Dezhurko, and F. T. Long, *On the methods of the resolution of the general problem of LFP*, Doklady AN BSSR,vol. 27, pp. 595–598, 1983 (in Russian).
25. R. Gabasov, and F. M. Kirillova, *Methods of linear programming, Vol. 1, 2 and 3*, Edition of the Minsk University, 1977, 1978, 1980 (in Russian).
26. R. Gabasov, and L. F. Dezhurko, *An adaptive method of solution of the general fractional-linear programming problem*, Doklady AN BSSR, vol. 29, no. 8, pp. 685–687, 1985 (in Russian).
27. R. Gabasov, F. M. Kirillova, and O. I. Kostyukova, *Solution of linear quadratic extremal problems*, Soviet Mathematics Doklady, vol. 31, pp. 99–103, 1985.
28. R. Gabasov, F. M. Kirillova, and S. V. Prischepova, *Optimal Feedback Control*, Springer-Verlag, London, 1995.
29. F. Ghellab, and M. O. Bibi, *Optimality and suboptimality criteria in a quadratic problem of optimal control with a piecewise linear entry*, International Journal of Mathematics in Operational Research, vol. 19, no. 1, pp. 1–18, 2021.
30. R. Guerbane, and M. O. Bibi, *Primal-dual method for a linear program with hybrid direction*, International Journal of Mathematics in Operational Research, vol. 23, no. 3, pp. 316–343, 2022.
31. M. A. Hakmi, *Méthode de support pour la résolution des problèmes d'optimisation fractionnaire linéaire*, Mémoire de Magistère en Mathématiques, Ecole Normale Supérieure de Laghouat, 2017.
32. K. Hassaini, and M. O. Bibi, *An Algorithm for Solving Quadratic Programming Problems with an M-matrix*, Statistics, Optimization & Information Computing, vol. 12, no. 2, pp. 405–417, 2024.
33. T. Illés, A. Szirmai, and T. Terlaky, *The finite criss-cross method for hyperbolic programming*, European Journal of Operational Research, vol. 114, no. 1, pp. 198–214, 1999.
34. N. Khimoum, and M. O. Bibi, *Primal-dual method for solving a linear-quadratic multi-input optimal control problem*, Optimization Letters, vol. 14, no. 3, pp. 653–669, 2020.
35. N. Khimoum, T. Mernache, and M. O. Bibi, *A constructive method for solving a multi-objective linear program with bounded variables*, International Journal of Multicriteria Decision Making, vol. 9, no. 2, pp. 154–173, 2022.
36. E. A. Kostina, and O. I. Kostyukova, *An algorithm for solving quadratic programming problems with linear equality and inequality constraints*, Computational Mathematics and Mathematical Physics, vol. 41, no. 7, pp. 960–973, 2001.

37. E.A. Kostina, *The long step rule in the bounded-variable dual simplex method: Numerical experiments*, Mathematical Methods of Operations Research, vol. 55, no. 3, pp. 413–429, 2002.
38. A. Laouar, and M. O. Bibi, *An adaptive hybrid direction algorithm for convex box-QP problems with enhanced pre-solving*, To appear in International Journal of Mathematics in Operational Research, doi: 10.1504/IJMOR.2023.10062725, 2025.
39. B. Martos, *Hyperbolic programming*, Publ. Math. Inst., Hungarian Academy of Sciences, vol. 5, pp. 386–406, 1960.
40. B. Martos, *Hyperbolic programming*, Naval Research Logistics Quarterly, vol. 11, pp. 135–155, 1964.
41. A. Nagih, and G. Plateau, *Problèmes fractionnaires: Tour d'horizon sur les applications et méthodes de résolution*, RAIRO Oper. Res., vol. 33, no. 4, pp. 383–419, 1999.
42. A.S. Nemirovski, *The long-step method of analytical centers for fractional problems*, Mathematical Programming, vol. 77, no. 2, pp. l91–224, 1997.
43. P. Pandian, and M. Jayalakshmi, *On solving linear fractional programming problems*, Modern Applied Science, vol. 7, no. 6, pp. 90–100, 2013.
44. N. Ploskas, and N. Samaras, *Linear programming using MATLAB*, Springer Optimization and Its Applications 127. Springer, Cham, 2017.
45. S. Radjef, and M. O. Bibi, *An effective generalization of the direct support method in quadratic convex programming*, Applied Mathematical Sciences, vol. 6, no. 31, pp. 1525–1540, 2012.
46. S. Radjef, and M. O. Bibi, *A new algorithm for linear multiobjective programming problems with bounded variables*, Arabian Journal of Mathematics, vol. 3, no. 1, pp. 79–92, 2014.
47. A. Rezzag, M. O. Bibi, and A. Laouar, *Reduction of bounded variables in integer linear programming problems*, To appear in International Journal of Mathematics in Operational Research, doi: 10.1504/IJMOR.2023.10059652, 2025.
48. S. Saha, M. Hossain, M. Uddin, and R. Mondal, *A new approach of solving linear fractional programming problem (LFP) by using computer algorithm*, Open Journal of Optimization, vol. 4, pp. 74–86, 2015.
49. S. Schaible, *Fractional programming : applications and algorithms*, European Journal of Operational Research, vol. 7, no. 2, pp. 111–120, 1981.
50. I. M. Stancu-Minasian, *Fractional Programming : Theory, Methods and Applications*, Kluwer Academic Publishers, 1997.
51. S. F. Tantawy, *A new procedure for solving linear fractional programming problems*, Mathematical and Computer Modelling, vol. 48, no. 5-6, pp. 969–973, 2008.
52. T. Terlaky, *Interior Point Methods in Mathematical Programming*, Kluwer academic publichers, Boston-London-Dordrecht, Delft, The Netherlands, 1996.
53. P. Wolfe, *The simplex method for quadratic programming*, Econometrica, vol. 27, pp. 382–398, 1959.
54. M. A. Zaitri, M. O. Bibi, and M. Bentobache, *A hybrid direction algorithm for solving optimal control problems*, Cogent Mathematics & Statistics, vol. 6, no. 1, Article ID 1612614, doi: 10.1080/25742558.2019.1612614, 2019.