Journal of
Applied & Computational Mathematics

Choufi, J Appl Computat Math 2017, 6:2
DOI: 10.4172/2168-9679.1000352

Research Article Open Access

Development of a Procedure for Finding Active Points of Linear Constraints

Said Choufi*

Department of Mathematics, University of Batna, Route de Biskra, Batna, Algeria

Abstract

In this paper, we present an iterative method to determine active point of linear constraints. It is based on two basic operations which are addition and permutation of constraints. This procedure generates a finite sequence of points that basis in a new lemma and a new formula direction, the laspoint of sequence constitutes an active point, and this procedure gives also two matrices. The first one is constituted by the active constraints which are linearly independent and the second one is a matrix whose columns are the basis vectors of the kernel of the first matrix.

Keywords: Active point; Interior point; Kernel of matrix; Optimisation continuous

Introduction

Currently, the domain of optimization is attracting considerable interest from the academic and industrial communities, see, for instances [1-5]. The various existing techniques for solving a given problem and the efficient algorithmic implementations open up many perspectives and diverse applications in different areas [6-8].

There are different methods of optimization exist in the literature, among other, we cite, the simplex [1], the interior point [1,2], the exterior point methods, and evidently with their improved versions [9-13].

In most optimization problems, initialization points are necessary and required in the resolution algorithms for performing numerical implementations [6-8,11,12]. However, the choice of the initialization points is not general, and the values of these points depend strongly on the adopted technique. Furthermore, these points are considered as active or feasible in the applied method.

In this study, we are interested by the optimization problem of the CSLP type (constraint satisfaction linear problems), where the set of constraints are linear and it is defined by determined the active point x_{act} of a set E such as:

$E = \{x \in IR^n, subject\ to\ Ax \leq b\}$ (CSLP), where

A is an mxn data matrix, not necessarily full rank and b is a given as vector IR^m

The problem to solve is the determination of the active points satisfying all the aforementioned constraints.

If the values of the matrix A and the vector b components are integer numbers, the above problem is discrete and can be solved by using the ellipse method [4]. However, in the case of optimization continuous, this is not studied in literature.

This past has motivated this investigation in the purpose of giving a theoretical and practical method of resolution of this problem.

The method that we propose is based on the construction of an iterative algorithm, such that, from any initial point (feasible or not) one produces another better point x, then one associates to it two matrices A and Z. The lines of A are constituted by the active constraints, which are linearly independent. The columns of matrix Z are constituted of the kernel of matrix A, and are also linearly independent.

This process allows to generate a sequence of points $(x_k)_{k\in IN}$ which converge to the point that one search (feasible or active). It is important to mention that our method can be applied without knowing whether this domain of constraints is empty or not.

In the numerical implementation, we used the scientific environment FOTRAN F90 under windows, and the obtained results were very satisfactory.

The rest of this paper is organized as follows. In Sec. 2, we give some definitions and propositions that are used in this article. In Sec. 3, we present the construction of a die kernel. In Sec. 4, we give description of the active method and its implementation in Sec.5, we form algorithm for the active method. At last, we summarize our results in the last section.

We note that:

I={1, 2, 3,..., m} the set of constraint indices.

$X_{k:}$ The iteration point k.

$I_k = \{i \in I : a_i^t x_k > b_i\}$ the set of constraints indices containing the point x_k externally.

$I^-_k = \{i \in I : a_i^t x_k > b_i\}$ the set of constraints indices containing the point x_k internally.

$\perp$: Orthogonal.

nt: The total number of iterations.

x_{act}: The active point.

x_{fe}: The feasible point.

Δ_K: The set of active constraints in x_k point.

A_k: The matrix formed by the active constraints linearly independent at the point x_k.

***Corresponding author:** Said Choufi, Department of Mathematics, University of Batna, Algeria, Tel: +213 33 31 91 34; E-mail: choufi_said @yahoo.fr

Received March 03, 2016; **Accepted** April 28, 2017; **Published** May 06, 2017

Citation: Choufi S (2017) Development of a Procedure for Finding Active Points of Linear Constraints. J Appl Computat Math 6: 352. doi: 10.4172/2168-9679.1000352

Z_k: The kernel of the matrix A_k.

np: The total number of permutations resulting by the choice x_0.

zel_i: The columns of index i deleted on the matrix Z_1.

Definitions and Propositions

Definition

The constraint $a_i^t x \le b_i$ is called active in x_k if $a_i^t x = b_i$ [8].

This definition leads to the fact that, for any vector v, we can introduce all

$$I_v = \{i \in I : a_i^t v = b_i\} \quad (3.1)$$

We then say that the vector v is a regular point of all eligible

$\Delta = \{v \in IR^n : a_i^t v \le b_i$, for each i ∈ I

(or just a regular point of the constraints) if and only if it is a regular point of $D = \{x \in IR^n; a_i^t(x) = b_i, i = 1,....,m\}$.

Definition

Let D be a domain of constraints in IR^n, defined by [5]

$$D = \{x \in IR^n; a_i^t(x) = b_i, i = 1,....,m\} \quad (3.2)$$

We call all candidates constraints, any set of constraints, among the a_i^t (x) ≤ b_i considered as active constraint of the solution that we search for.

Proposition

A direction d is tangent to x∈X, if and only if there exists a sequence (d_n) of limit d, and a sequence (μ_n) of positive real zero limit, such that $x+\mu_n d_n \in X$. [5].

Remarks

Most of the algorithms fail when they have to solve a problem whose constraints are not qualified in the solution. Therefore it is preferable to change the description of the set of constraints before solving the problem.

Proposition (CS contraints satisfaction)

The constraints of D domain are qualified at the point x ∈ D, if the gradients in x of the active constraints [5].

$\{\nabla h_i(x) i = 1,...., p\} \cup \{\nabla g_j(x) j = 1,....,q\}$ are linearly independent, where

$h_i(x) = b_i for\ i = 1,...,p\ and\ g_j(x) < b_j\ for\ j = 1,...,q$ such that p +q=m.

Construction of Kernel Matrix

This section contains the most important results for the kernel numerical calculation of any matrix, especially in the case where we have a lot of matrices of large size, and to avoid repetition of the calculations.

The calculation of a matrices and vector kernel obeys certain rules of compliance.

These results are given in the following:

Lemma

Let v be a vector v ∈R^n, which defines a set of constraints as follows:

$\Delta = \{x \in IR^n : v^t x - a = 0\}$ Where α is a real number (4.1)

then $v \perp \Delta$ i.e. v is orthogonal to Δ.

Proof

Let $x_1, x_2 \in \Delta$, *then* $v^t x_1 - \alpha = 0$ (1)

$v^t x_2 - \alpha = 0$ (2)

By subtraction (1) of (2) it comes: $v^t x_2 - \alpha - v^t x_1 + \alpha = 0 \Rightarrow v^t(x_2 - x_1) = 0$

This gives that $v \perp x_1 x_2$ for each $x_1, x_2 \in \Delta$. *so* $v \perp \Delta$.

From this lemma, we can construct sets Δ_+ and Δ_- that help us to lead the constraint equations in the following two corollaries:

Corollary

Let Δ be a set of constraints of the form: $\Delta = \{x \in | R^n : v^t x - \alpha \le 0\}$

Where v is a vector of $IR^n, \alpha \in R$. Then $\Delta_+ = \{a \in R^n : x, x' R^n$ such that $a = x - x'$ with $-\alpha < 0$, $v^t x' - \alpha = 0$, $v \in \Delta_+$ (4.2)

Corollary

Consider the same data of corollary 4.2, then $\Delta = \{a \in | R^n : \exists x, \in | x' | R^n$ such that $a = x' - x$ with $\}$ ….(4.3) where $-v \in \Delta$.

Lemma

Let v be a vector in $|R^{n^*}$, then its kernel is formed by the following basis $\{z_1, z_2, z_3, \ldots, z_{n-1}\}$ where $v^t z_i = 0$, for each i=1,2,…,n-1 (4.4)

Proof

It suffices to show that the rows of the matrix $(v\ z_1\ z_2 ... z_{n-2}\ z_{n-1})^t$ are linearly independent. Consider the scalars $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{n-1}, \lambda$ satisfying $\lambda v + \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3 + \ldots + \lambda_{n-1} z_{n-1} = 0$ Multiplying by v^t, it follows: $\lambda v^t v + \lambda_1 v^t z_1 + \lambda_2 v^t z_2 + \lambda_3 v^t z_3 + \ldots . \lambda_{n-1} v^t z_{n-1} = 0$ such as $v^t z_i = v^t z_j\, 0\, i = 1,2,\ldots,n-1$ and $v^t v = \sum_{i=1}^{n} v_i^2$ then λ=0. And (1), we have: $\lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3 + \ldots + \lambda_{n-1} z_{n-1} = 0$ $\lambda_i = 0$ for each i=1,…,n-1, from where the result.

Lemma

Let v_1 and v_2 be a two vectors in R^n, the set $\{z_1, z_2, z_3, ..., z_{n-1}\}$ is a basis of the kernel vector v_1, where the i_m index {1,2,...., n-1} satisfies

$$v_2^t z_{im} \neq \underset{i=1}{\overset{n-1}{Max}} \left| v_2^t z_i \right| \quad (4\text{-}5)$$

i=1

Then the matrix $(v_1\ v_2\ z_1 ..... z_{im-1}\ z_{im+1} ... z_{n-1})^t$ is invertible if and only if $v_2^t z_{im} \neq 0$.

Proof

The same proof of Lemma (4.4), in the rows of the matrix $(v_1\ v_2\ z_1 \ldots\ldots z_{im-1} \ldots .. z_{n-1})^t$ which are linearly independent vectors.

Corollary

Keeping the same data of Lemma (4-4), but here $v_2^t z_{im} \neq 0$. Then the matrix $(v_2\ z_1\ z_2\ \ldots\ldots z_{n-2}\ z_{n-1})^t$ is full rank.

Proof

The vectors v_1, v_2 are linearly dependent.

Lemma

Let $v_1$and v_2 be a two vectors in R^n, admitting $\{z_1, z_2, z_3, \ldots, z_{n-1}\}$

as the kernel of a basis vector v_1,$\{ i_1, i_2, i_3,\ldots, i_k\}$ the index set of {1, 2, 3,…, n-1} such that $v_2^t z_j \neq 0$ for each $j \notin \{ i_1, i_2, i_3,\ldots, i_k\}$.

And $v_2^t z_j = 0$ for each $j \notin \{ i_1, i_2, i_3,\ldots, i \}$ (4.6)

Then the set $\{ z'_{i1}, z'_{i2}, \ldots, z'_{ik}\} \cup \{ z_i , i \notin \{ i_1, i_2, i_3,\ldots, i_k\} \}$ form a common basis of the kernel vectors v_1 and v_2, and verifying $z'_j = z_j + \beta_j z_{im}$ with $\beta_j = -v^{T_2} z_j / v^{T_2} z_{im}$

where $v_2^t z_{im} = Max\left|v_2^t z_j\right| = \underset{j\in I/\{i_1,i_2,\ldots,i_k\}}{Max}\left|v_2^t z_j\right|$

$j \in \{1,2,\ldots\ldots,n-1\}$

Proof

Since $\{ z_1, z_2, z_3,\ldots, z_{n-1}\}$ is a basis of the kernel vector v_1.

We will show that $\{z'_{i1}, z'_{i2},\ldots\ldots z'_{i(k-1)}, z'_{ik}\} \cup \{z_i : i \in \{i \notin \{i_1, i_2,\ldots\ldots,i_k\}$ (4.7)

from a common basis vectors v_1 and v_2, knowing $\{z'_{i1}, z'_{i2},\ldots\ldots z'_{i(k-1)}, z'_{ik}\}$ resulting kernel v_1 such that

$z'_i = z_i + \beta_i z_{im}$ and $v_1^t z'i = v_1^t z_i + \beta_i = v_1^t z_{im}\ 0 + \beta_i 0 = 0$

When $i \in \{ i_1, i_2,\ldots,i_k\}$. For vector v_2, we have: if $i \in \{ i_1, i_2,\ldots,i_k\}$, it comes

$\Rightarrow z'_i \in$ Ker (v_2), for each $v_2^t z'_i = v_2^t z'_i - \frac{v_2^t z'_i}{v_2^t z_{im}} = v_2^t z'_i - v_2^t z'_i = 0$ $i \in \{ i_1, i_2,\ldots,i_k\}$.

And if $i \in \{1,2,\ldots,n-1\} / \{ i_1, i_2,\ldots,i_k\}$, it comes $v_2^t z_i = 0$ because $\{i_1, i_2,\ldots,i_k\}$is the largest subset of {1,2,...,n-1} satisfies $v_2^t z_j \neq 0$, when i $\in\{ i_1, i_2,\ldots,i_k \}$.

It remains to show that set (4.7) is linearly independent.

$\sum_{l=1}^{k} \lambda'_{il} z'_{il} + \sum_{i\in\{1,2,\ldots\ldots\ldots,n-1\}/\{i_1,i_2,\ldots\ldots i_k\}} \lambda_i z_i = 0$ impose

$\sum_{l=1}^{k} \lambda'_{il} (z'_{il} + \beta_{il} z_{im}) + \sum_{i\in\{1,2,\ldots\ldots\ldots,n-1\}/\{i_1,i_2,\ldots\ldots i_k\}} \lambda_i z_i = 0$

$\sum_{i=1}^{k} \lambda'_{il} z_{il} + \sum_{i=1}^{k} \lambda'_{il} \beta_{il} z_{im} + \sum_{i\in\{1,2,\ldots n-1\}/\{i_1,i_2,\ldots\ldots i_k\}} \lambda_i z_i = 0$

$\Rightarrow \lambda'_{il} = 0$ l=1,…,k with $i_l \neq i_m$ et $\lambda_i = 0$ if i $\in$ {1,2,...,n-1}/{ $i_1, i_2,\ldots,i_k$ }.

Because { $z_1, z_2, z_3,\ldots, z_{n-1}$} are linearly independent, from where the result.

Description of the Active Method

We focus in this section on a so-called active point approach

We construct the iterated x_{k+1} by the formula $x_{k+1} = x_k + \alpha_k d_k$ where α_k is the displacement step in the d_k direction.

The choice of α_k and d_k ensures that x_{k+1} approaches the border of the constraints better than x_k, and $A_{k+1} = (\frac{A_k}{a_{k+1}^t})$ .. (5.1) such as A_k is the matrix of active constraints at point x_{k+1}

This process is repeated until the stopping test is satisfied.

Initialization

Location of the starting point x_0: Let E be a set of constraints in a general form (equalities, inequalities, and mixed) be an arbitrary point and x_0 be a point of departure in IR^n.

We can distinguish the situation from the point x_0 with respect to E, in one of the following three cases:

Case 1: Point x_0 is located within E.

Case 2: The point x_0 is located outside of E.

Case 3: The point x_0 is located in the boundary of E.

Geometric representation at point x_0

Adding and permutation of Constraints

Adding a constraint: Let A_k be the matrix of active constraints at x_k point of iteration k, stitch-forming iteration k +1, we add in the matrix A_k the constraint a_{k+1}^t resulting from the following two equations:

$a_{k+1}^t x_k - b_{k+1} = \underset{i\in I\_(x_k)}{Max}{}^{(a_i^t x_k - b_i)}$ if x_k is the result of the first or third case cited in sub -section (5.1.1)

Because, When the point x_k is situated in the interior domain $E(\xi = 1)$, We seek the constraint that nears to this xk point. in fact, If $i \in I\_(x_k)$ it gives all $(a_i^t x_k - b_i) < 0$. Then we choose the constraint $(a_{k+1}^t x_k - b_{k+1})$ that have a negate f max-value $(a_{k+1}^t x_k - b_{k+1}) < 0$. Note that $a_{k+1}^t x_k - b_{k+1} = \underset{i\in I\_(x_k)}{Max}(a_i^t x_k - b_i)$ (5.2)

This result is obtained by remplacing the x_k point in all constraint of domain E (Figure 1).

Else in other part if x_k is the result of second case cited in sub-section 5.1.1. i.e.

The point x_k is cited in exterior of E, we seek the constraint which is far to this point x_k in fact.

If $i \in I_+(x_k)$, it gives all $(a_i^t x_k - b_i) > 0$. Then we choose the constraint $(a_i^t x_k - b_i) > 0$ that have a positif max-value: $(a_i^t x_k - b_i) > 0$.

Note that $a_{k+1}^t x_k - b_{k+1} = \underset{i\in I_+(x_k)}{Max}(a_i^t x_k - b_i)$ (5.3)

This result is obtained by remplacing the x_k point in all constraint of domain E.

Permutation of constraints: Let x_k be the point in iteration k, in which two matrices are associated A_k, Z_k, and i_k is the index on constraints that can be added to A_k to obtain A_{k+1}, z_k is the column that can be eliminated from the matrix Z_{k-1} to reach Z_k.

Permute the constraint of index i_k by another constraint of index i_0 that result of equality: $\left|a_{i0}^t zel_{i0}\right| = \underset{i\in\{1,2,\ldots,k\}}{Max}\left|a_i^t zel_i\right|$

If and only if where the algorithm is moved from iteration k to iteration k+1 we meet the condition $a_{ik}^t Z_k = 0_{nk}^t$

Where n_k is the number of columns of matrix Z_k.

and $\{zel_i, i = 1,\ldots,k\}$ is a set of columns eliminated on the matrix Z_1.

Remarks: (1) The rows of the matrix A_{k+1} are linearly independent, they are also active at the point x_{k+1}. (2) From the kernel of the matrix A_k, we can easily determine the Z_{k+1} matrix whose columns form a basis of the kernel of A_{k+1}.

Direction of displacement

We consider the matrix A_k composed of active constraints linearly independent at the point x_k and the columns of the matrix Z_k form a basis of the kernel of A_k.

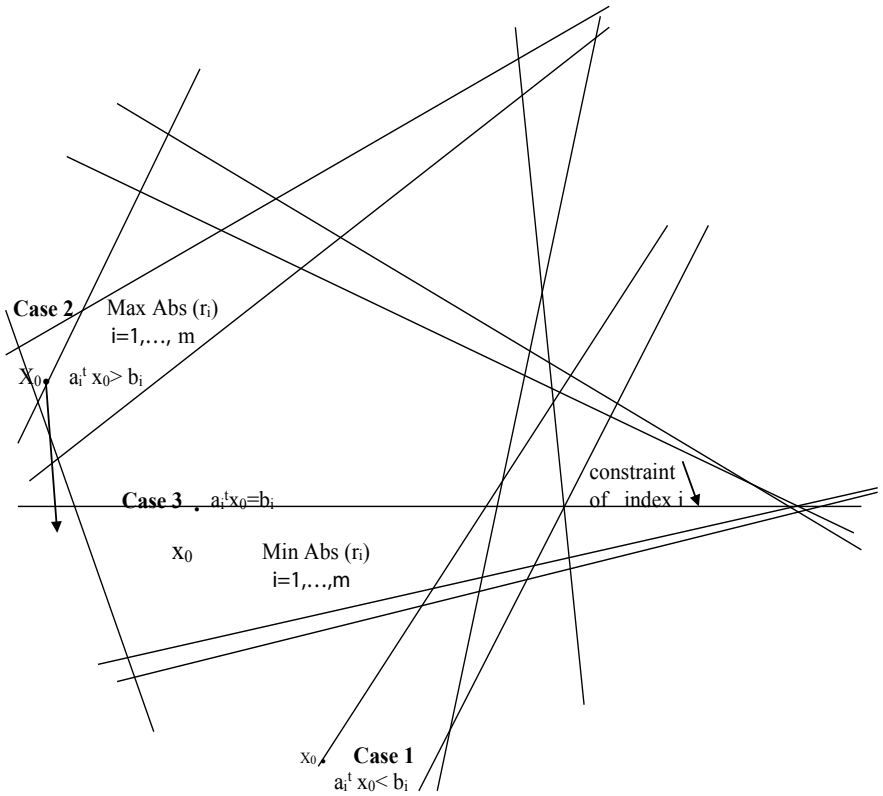

Figure 1: Geometric representation at point x_0.

a_{k+1}^t the constraint that may be added to the matrix A_k.

To determine the direction d_k, we distinguish two alternatives:

If k=0, we pose $d_k = \xi.\nabla a^T{}_{ik} x_k - b_{ik})$…. (5.4)

Where, ik is the index of the constraint to added to A_k.

ξ is indicative of the position x_k, and

if x_k is the result of the second case. (§5.1.1)

ξ=1 If x_k is the result of another case. (§5.1.1)

If k ≠ 0, here, we find also two other alternative:

If $a_{ik}^t Z_k \neq 0_{nk}$ the direction d_k is resulted by solution of the following linear system:

$$\begin{pmatrix} A_k \\ a_{ik}^t \\ Z_k^- \end{pmatrix} d_k = \begin{pmatrix} 0 \\ \xi \\ 0 \end{pmatrix} \quad (5.5)$$

$$Where\, Z_k = \begin{pmatrix} Z_k^- & z_k \end{pmatrix}, \xi = \begin{cases} -1 if\ x_0 \in case2(\S 4-1-1) \\ 1 if\ x_0 \in E \end{cases}$$

and z_k is the column that can be eliminated from the matrix Z_{k-1} to obtain Z_k.

If $a_{ik}^t Z_k = 0_{nk}$, the direction d_k is resulted by the solve of the following linear system:

$$\begin{pmatrix} A_k^- \\ a_{ik}^t \\ Z_k \end{pmatrix} d_k = \begin{pmatrix} 0 \\ \xi \\ 0 \end{pmatrix} \quad (5.6)$$

Where $A_k = \begin{pmatrix} A_k^- \\ a_{i0}^t \end{pmatrix}, \xi = \begin{cases} -1 if\ x_0 \in case2(\S 4-1-1) \\ 1 if\ x_0 \in E \end{cases}$ and i_o is the index of constraint satisfying $\left|a_{i0}^t zel_{i0}\right| = \underset{i\in\{1,2,\ldots,k\}}{Max} \left|a_i^t zel_i\right|$ that concerned by the permutation. (By application of Lemma 4.7).

Step of displacement

Let x_k is the point of iteration k, and d_k the direction of displacement at the point x_k.

After finding the associated constraint of iteration (k+1) which is active at the point xk+1, then $a_{k+1}^t x_k - b_{k+1} = -\propto_k . a_{k+1}^t d_k$, from the determination of the direction dk, it comes that $a_{k+1}^t d_k = \xi$, Which gives $a_{k+1}^t x_k - b_{k+1} = -\propto_k .\xi$ as $\propto_k > 0$ and $|\xi| = 1$.

We conclude that $\alpha_k = Abs(a_{k+1}^t x_k - b_{k+1})$. So, the step in the direction of displacement dk denoted by αk is given by the following expression

$$\propto_k = Abs\left(b_{ik} - a_{ik}^t x_k\right) \quad (5.7)$$

Where ik is the index of constraint to added in the matrix A_k.

Remarks: The active constraint at x_k, is also active at the point x_{k+1}.

Theorem of convergence

Let $(x_k)_k \in_{IN}$ be an iterative sequence defined by $x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$, where α_{k-1} is the displacement step along the direction d_{k-1}, and x_0 is a finit starting point of IR^n.

Then the sequence $(x_k)_k \in_{IN}$ formed by a set of the directions

$(d_k)_k \in_{IN}$ for the two cases (§5-1-1) Is convergent after a finite number of iterations.

Proof

It sufficiently to show that the set of direction $(dk)_{k\in IN}$ is linearly independent

By recurrence we can write that the scalar product

$\sum_{i=1}^{m+1} \lambda_i . d_i = 0$ such that $\lambda_i \in IR$ *and* d_i *the set of direction.*

First we consider $\lambda_1 d_1 + \lambda_2 d_{2=} 0$ (*) and we proof $\lambda_1 = \lambda_2 = 0$

With application of our new defined direction, then

$\lambda_1 . a_{i1}^t d_1 + \lambda_2 . a_{i1}^t d_2 = 0$, we have $a_{i1}^t d_1 = \xi \neq 0$ and $a_{i1}^t d_2 = 0$ then

$\lambda_1 \xi + \lambda_2 d_2 = 0$ and $\lambda_1 . \xi = 0$ *so* $\lambda_1 = 0$

we replace in (*), we obtain $\lambda_2 . d_2 = 0$

as we know d_2 is a non-null direction, then it result that $\lambda_{2=} 0$

now we have $\lambda_1 = \lambda_2 = 0$

we suppose that are true for all step m, and we proof it for step m+1.

$\lambda_1 d_1 + \lambda_2 d_2 + \ldots + \lambda_m d_m = 0 \, d_i \neq 0, \; \lambda_i = 0 \; for \; i = 1, \ldots, m$

$\lambda_1 d_1 + \lambda_2 d_2 + \ldots + \lambda_m d_m + \lambda_{m+1} d_{m+1} = 0$

$\lambda_m d_m + \lambda_{m+1} d_{m+1} = x_{m+1} - x_m + x_{m+2} - x_{m+1}$

$= -x_m + x_{m+2}$

$= -x_m + x_m + \lambda_m d_m + \lambda_{m+1} d_{m+1}$

$= -x_m + x_m + \lambda_m d_m + \lambda_{m+1} d_{m+1}$

$= \lambda_{m+1} d_{m+1}$

We have $\lambda_m = 0$, and we know, the direction d_{m+1} is non-null, it result $\lambda_{m+1} = 0$.

Finally the set of direction $(d_k)_{k\in IN}$ is linearly independent.

Algorithm for the Active Method

Data: The matrix A, the vector b and the departure x_0.

Output: The point to find is active exact point.

1-Choose an arbitrary starting point, x_0 in IR^n, set k=0.

2- As long as stopping criterion is defined.

a) Computation of a searched direction, calculate d_k.

b) Determine the step α_k, and the new point $x_{k+1}=x_k+\alpha_k d_k$ and add the active constraint a_{ik}^t in x_{k+1}.

c) Test: if $a_{ik}^t Z_k = 0_{nk}^t$, we call the permute procedure.

d) Construct the active matrix $A_{k+1} = \begin{pmatrix} A_k \\ a_{ik}^t \end{pmatrix}$

The same way, we calculate the basis of $KerA_{k+1}$.

e) K=k+1 and return to a).

Remarks:

i) This method determines the active points of a problem (E), without any constraints condition i.e., it does not require to make the linearly independent constraints.

ii) This method can be applied to any set E, defined by linear constraints, and even if it is empty.

Numerical Tests

From a practical point of view, our method has remarkable advantages.

This will be shown by numerical application of this method in different cases that may exist: the number of constraints, the number of variables, and the size of the matrix to be taken.

The obtained results are listed in the following tables:

Case 1: Standard form (Small size, Large size)

a - Let m=11 and n=5

$$A = \begin{pmatrix} 1 & 1 & 1 & -1 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ 2 & 1 & 3 & 0 & -1 \\ -2 & -1 & -3 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \text{ and } b = \begin{pmatrix} 8 \\ -8 \\ 20 \\ -20 \\ 5 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Table 1 shows the Standard form of small size and large size [9].

b - Let m=26 and n=12

$X_{11} + x_{12} + x_{13} + x_{14} = 50$

$X_{21} + x_{22} + x_{23} + x_{24} = 30$

$X_{31} + x_{32} + x_{33} + x_{34} = 70$

$X_{11} + x_{21} + x_{31} = 30 \; x_{ij} \geq 0 \; i = 1,2,3 \; and \; j = 1, \ldots, 4$

$X_{12} + x_{22} + x_{32} = 60 \quad X_{13} + x_{23} + x_{33} = 20 \quad X_{14} + x_{24} + x_{34} = 40$

Table 2 shows the inequality form in Large size.

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} A_2 = \begin{pmatrix} -1 & -1 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

x_0	r_0	a_{i1}	r_1	a_{i2}		ai_3	r_3	xsol	Z_{xsol}
8	-20	2	-7	1	-3.999	-1	-1.826×10^{-7}	1,833	$-1-1{,}49\times10^{-8}$
8		1		1		-1		3,166	1 0
8		3		0		-1 1		7,5	0,333 0,333
8		0		0		0		4,5	0,333 0,333
8		-1		0				9,333	0 1

Table 1: Standard form (Small size, Large size).

x_{01}	x_{02}	r_0	x_{sol1}	x_{sol2}	nt	A_x	Z_x	r_6
8	8	-38	10	1,666	6	A_1A_2	z_{1x}	$-2,885\times10^{-7}$
8	8		20	8,333			z_{2x}	
8	8		6,66	14,99				
8	8		13,33	25				
8	8		4,99	11,66				
8	8		14,99	18,33				

Table 2: Inequality form (Large size).

x_{01}	x_{02}	r_0	x_{sol1}	x_{sol2}	nt	A_x	Z_{1x}	Z_{2x}
8	8	-700.131	1,64	1,258	9		$4,49\times10^{-4}$	$-3,95\times10^{-10}$
8	8		1,258	36,878			0,013	-0,349
8	8		-64,23	36,878			0	-0,349
8	8		0,777	49,999			1,56x10-5	0
8	8		1,639	28,258			4,49x10-4	1

Table 3: Mixed form (large size).

$$Z_1^t=\begin{pmatrix}0&0&0&0&-1&0&0&1&1&0\\1&-1&0&0&0&0&0&0&-1&1\\1&0&-1&0&-1&0&1&0&0&0\\-1&0&0&1&0&0&0&0&1&0\\1&0&-1&0&0&0&0&0&-1&0\\1&-1&0&0&-1&1&0&0&0&0\end{pmatrix}\quad Z_2^t=\begin{pmatrix}0&-1\\0&0\\0&0\\0&-1\\1&0\\0&0\end{pmatrix}$$

Case 2: Inequality form (Large size)

Let m=39 and n=10

$$A=\begin{pmatrix}A_1\\A_2\\A_3\\A_4\end{pmatrix}\ and\ \mathrm{b}=\begin{pmatrix}\mathrm{b}_1\\\mathrm{b}_2\\\mathrm{b}_3\\\mathrm{b}_4\end{pmatrix}$$

$$A_1=\begin{pmatrix}9.119&61.555&0.012&0&0&0&0&-a&-a&-a\\-9.119&-61.555&-0.012&0&0&0&0&a&a&a\\-9.119&0&0&0&0&0&0&a&0&0\\0&-61.555&0&0&0&0&0&0&a&0\\0&0&-0.012&0&0&0&0&0&0&a\\-3.475&0&0&100&0&0&0&0&0&0\\3.475&0&0&-100&0&0&0&0&0&0\\0&-3.475&0&0&100&0&0&0&0&0\\7.407&0&0&0&-57.407&0&1&-1&0&0\\-7.407&50.0017&0&0&50&50&0&1.908&4.681&0\end{pmatrix}$$

a=1.231059

$$b_1^t=\begin{pmatrix}-50&250&150&150&150&72,068&-72,068&159.538&434.365&500\end{pmatrix}$$

$$A_2=\begin{pmatrix}7.407&-50.0017&0&0&-50&-50&0&-1.908&-4.681&0\\0&50&-0.01&0&0&-50&0&0&-8.999&-8.999\\1&-1&0&0&0&0&0&0&0&0\\-1&0&0&0&1&0&0&0&0&0\\1&0&0&0&0&0&0&0&0&0\\-1&0&0&0&0&0&0&0&0&0\\0&0&-3.475&0&0&100&0&0&0&0\\0&0&0&0&0&-1&0&0&0&0\\0&0&0&0&0&1&0&0&0&0\\0&-1&0&0&0&0&0&0&0&0\end{pmatrix}$$

$$b_2^t=\begin{pmatrix}-500&234.076&0&0&8.037&-1&349.031&-1&9&-4.5\end{pmatrix}$$

$$A_3=\begin{pmatrix}0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&-1&0&0&0\\0&0&0&0&0&0&1&0&0&0\\0&0&-1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&-1&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&-1&0\\0&0&0&0&0&0&0&0&1&0\\0&o&0&-1&0&0&0&0&0&0\end{pmatrix}\quad b_3=\begin{pmatrix}9.718\\-0.01\\100\\-9\\10\\-0.01\\54.918\\-50\\100\\-0.001\end{pmatrix}$$

$$A_4=\begin{pmatrix}0&0&0&1&0&0&0&0&0&0\\0&0&0&0&-1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&-1\\0&0&0&0&0&0&0&0&0&1\\0&-1&0&0&0&1&0&0&0&0\\0&1&-1&0&0&0&0&0&0&0\\0&0&0&0&0&0&-1&1&0&0\\0&0&0&0&0&0&1&-1&0&0\end{pmatrix}\quad b_4=\begin{pmatrix}10\\-1\\4.6838\\-0.01\\50\\0\\0\\0\\50\end{pmatrix}$$

Table 3 shows the mixed form in large size.

$$A_x=\begin{pmatrix}-3,475&0&0&100&0&0&0&0&0&0\\-7,407&50,0017&0&0&50&50&0&1,908&4,681&0\\9,119&61,555&0,012&0&0&0&0&-a&-a&-a\\0&-3,475&0&0&100&0&0&0&0&0\\0&0&-3,475&0&0&100&0&0&0&0\\0&0&0&0&0&0&0&0&-1&0\\0&-1&0&0&0&1&0&0&0&0\\-1&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&-1&0&0&0\end{pmatrix}$$

Case 3 Mixed form (Large size)

a - Let m=23 and n=7

$$A=\begin{pmatrix}A_1\\A_2\\A_3\end{pmatrix}\ and\ b=\begin{pmatrix}b_1\\b_2\\b_3\end{pmatrix}$$

such that m_i=8, i=1,2 and m_3=7

$$A_1=\begin{pmatrix}0&0&-0.325&1&-1.098&0&0\\0&0&0&0&-0.1316&0&1\\-12.2&10&0&0&0&0&0\\12.2&-10&0&0&0&0&0\\-127.01542&10&0&0&-200&0&0\\127.01542&-10&0&0&200&0&0\\65.346&0&0&20.346&200&0&0\\-65.346&0&0&-20.346&-200&0&0\end{pmatrix}$$

$$A_2=\begin{pmatrix}-1&0&0&0&0&0&2,1568\\1&0&0&0&0&0&-2.1568\\0&0&1.0833&0&0&0&0\\-1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\0&-1&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&-1&0&0&0&0\end{pmatrix},b_1=\begin{pmatrix}-0.1\\200\\-0.1\\-1600\\0\\0\\0\\0\end{pmatrix}$$

x_0	r_0	ai1	x_1	ai2	x_2	ai3	x_3	ai4	n_p	r_4	xsol
8	-2536.12	127.015	2.271	12.2	-62.967	65.346	3.8	-12.2	1	-1600	Is not found
8		-100	8.45	-10	83.179	0	164.64	10			
8		0	8	0	8	0	8	0			
8		0	8	0	8	20.346	-69.388	0			
8		200	-1.02	0	44.14	200	5.815	0			
8		0	8	0	8	0	8	0			
8		0	8	0	8	0	8	0			

Table 4: Standard form (large size).

x_{01}	x_{02}	r_0	x_{sol1}	x_{sol2}	nt	A_x	z_x	r_4
-8	-8	-30	-8	-9.18	4	A1 A2		-1.62×10^{-6}
-8	- 8		-9.4	-6,32				
- 8	-10		-8	-10				
-8	-10		-2,87	-7.88				
-8	-10		-9,38	-6.17				
-8			-9,41					
-8			-13,27					
-8			-6.25					
-8			-6.81					
-8			-10.82					

Table 5: Mixed form (large size).

1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	-0,15	0	-1,1	-0,15	1,35	-1,5	1,35	-0,3	-0,4	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0
0	-0,25	0	-1,5	-0,25	-0,75	0,5	2,25	-0,5	0	0
0	0	0	0	0	0	0	0	0	0	1
0	0,05	0	0,7	0,05	-0,45	1,5	-0,45	0.1	0,8	0,4
0	0,65	0	2,1	0,65	0,149	0,5	-2,85		0	0

Table 6: Where, the remaining matrices are given by: Zx=obseve in the above table.

$$A_3=\begin{pmatrix}0&0&1&0&0&0&0\\0&0&0&-1&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&-1&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&-1\end{pmatrix},b_2=\begin{pmatrix}120\\-0.01\\32.786\\885\\0\\20\\-85\\95\end{pmatrix}\ and\ b_3=\begin{pmatrix}-92.666\\66\\95\\-3\\12\\1.412\\-0.819\end{pmatrix}$$

Table 5 shows the mixed form in large size.

Where, the remaining matrices are given by:

Where, the remaining matrices are given by: Zx=obseve in Table 6

$$A_1=\begin{pmatrix}0&0&0&-2&0&0&3&-2&0&0\\0&1&0&-1&0&1&0&0&0&2\\0&0&0&-1&1&0&0&3&0&0\\0&0&0&1&2&0&0&0&-3&0\end{pmatrix},\text{and } A_2=\begin{pmatrix}0&1&0&-1&0\\0&-3&0&1&-3\\2&0&0&-3&0\\0&2&0&0&1\end{pmatrix}$$

Discussion

The numerical results obtained in the previous examples, show the efficiency of the proposed a new method to solve any initialization problem of optimization with linear constraints.

In the first example, with $x_0=(8,\ 8,\ 8,\ 8,\ 8)$, we remark that the constraint in the first iteration, appears also in other iterations until the third iteration. The last iteration gives the active point, $x_{act}=(1.833,3.166,7.5,4.5,9.333)$

In addition, our method allows defining two matrices A_3 and Z_3 satisfying:

$$A_3=\begin{pmatrix}2&1&3&0&-1\\1&1&0&0&0\\-1&-1&-1&1&0\end{pmatrix}$$

by constraints of the domain of optimization. It satisfied $A_x.X_{act}=b_x$,

And its lines are linearly independent.

$$Z_3^t=\begin{pmatrix}-1&1&0.333&0.333&0\\-1.49x10^{-8}&0&0.333&0.333&1\end{pmatrix}$$

Satisfies $A_x.Z_x=0_{\ mxxn}$

We observe that x_{act} is feasible with respect to the constraints which do not appear in A_x.

From a numerical point of view, it is difficult to take the best starting point in IR^n, which helps us to obtain easily the active point that we search.

In a numerical application, it is competent to verify whether the domain of optimization is empty or not. This problem is very easy to solve it by our method.

We easily can know the state of the domain. This has been illustrated in the numerical test-case 3 (Table 4).

All the above results show the efficiency of this method in the problem of optimization, where the domains are consecutives, and with small size.

For large size, the problem is substantially the same; one has only to do a large amount of calculations.

So, the discussion is similar to that of domains of small sizes.

These results are showed in examples of three cases.

Conclusion

After a long scientific research, we have not found any thing on the method that discusses to solve this type of continues problem optimization, and then we have suggested this method with a new formula direction d_k.

In this work, we studied theoretically and algorithmically an active method, which determines the extremes of a set defined by linear constraints. This set is in the form of equalities, inequalities or both of them. These m constraints are linear and function of n variables, our results can be givens in the following points:

- Starting from any initial point, it generates points belonging to the set E.
- It is possible to construct from the m constraints two matrices, where the lines of the first are linearly independents and actives, and the columns of the second form a basis of the kernel of the first matrix.
- Our method can be applied to matrices of large sizes.
- The active point is determined in at most n+ np iterations.
- The other advantage is the simplification of the computation, because each used constraint appears at most only once.
- Our method can be used in other algorithms of resolution of optimization problems to simplify their initializations and to improve their results.

References

1. Adler L, Karmarkar N, Resende MGC, Veiga G (1989) An implementation of armarkar algorithm for linear programming. Math Program 44: 297-335.
2. Gill PE, Murray W, Saunders MA, Wright MH (1991) Inertia controlling method of the quadratic programming. SIAM Rev Soc Ind Appl Math 33: 1-36.
3. Mansour MS, Kadri EH, Kenai S, Ghrici M, Bennaceur R (2011) Influence of calcined kaolin on mortar properties.Construction and Building Materials 25: 2275-2282.
4. Altman A, Gondzio J (1998) Symmetric indefinite systems in interior point methods for linear and quadratic optimization. Optimization methods and Software, pp 275-302.
5. Gerad SG (2001) Introduction aux méthodes de point intérieurs.février.
6. Gondzio J, Sarkissian R (2003) Interior point solver for structured linear programs. Math Program 96: 561-584.
7. Adil MB, Zhang J (2003) Comparative analysis of the cutting angle and similated annealing methods in global optimization.Journal of Mathematical Programming Operation Resarch 52: 363-378.
8. Chinchuluun A, Pardalos PM, Enkhbat R (2005) Global minimisation Algorithms for concave quadratic programming problems. Journal of Mathematical Programming Operation Resarch 54: 627-639.
9. Farouk A (2006) Programmation linéaire.
10. Azevedo AT, Oliveira LRA, Soares S (2008) Interior point method for long-term generation scheduling of large-scale hydrothermal systems. Ann Oper Res 169: 55.
11. Morales JL, Nocedal J, Wu Y (2012) A sequential quadratic algorithm with an additional equality constraint phase.IMA Journal of Numerical analysis 32: 553-579.
12. Dussqult JP (2011) Programmation non linéaire.
13. Culioli J C (2012) Introduction à l'optimisation, pp 384.