Proceedings of the International MultiConference of Engineers and Computer Scientists 2015 Vol II,
IMECS 2015, March 18 - 20, 2015, Hong Kong

Algorithms for Optimization of Convex Quadratic Functions

Amel Belabbaci, and Bachir Djebbar

***Abstract*—This paper deals with optimization of convex quadratic problems. For these problems we propose two algorithms. The first algorithm is similar to the simplex one. The second is based on the calculation of distances between an external point (critical point) and vertices of convex. The optimal solution is obtained at the farthest vertex from the critical point.**

***Index Terms*—convex quadratic program, vertex, critical point, separation.**

I. Introduction

LET $f(x) = \sum_i^n \alpha_i x_i + \beta_i x_i^2$ with any real numbers α_i and positive β_i as coefficients. We consider the following quadratic problem:

$$\max f(x)$$

subject to the constraints defined by the inequalities $Ax \leqslant b$ with a $m \times n$ real matrix A, and vector b of $\mathbb{R}_+^m$. We denote by Ω the convex formed by the constraints inequalities. Then Ω is a closed bounded convex set of $\mathbb{R}^n$. It is well known that the solution is reached at a vertex of Ω. Several methods can be used for searching the optimal vertex. We cite the extreme points ranking method: Starting from one of the vertices the nearbouring vertices are ranked with the calculation of objective function value at every vertex. This provides a new vertex to move to. The process will continue until no adjacent vertex can be found with an increasing objective function value. At each step, a linear program is solved. This method is computationally infeasible that is why others approaches can be used (see [4]–[6]).

In this paper, we give two algorithms for searching the optimal solution. The first algorithm is similar to the simplex algorithm. It allows the passage from an extreme point to another one, and permits to obtain the form of the objective function after each passage.

The second algorithm is an algorithm of separation and elimination. It permits to search the farthest vertex from an external point (called critical point) $x^* = -\frac{\alpha_i}{2\beta_i}$. The proposed algorithm is the adaptation of the other one proposed in [2] and [7] for the concave case.

The paper is organized as follows: In Section 2, we give a description of the first algorithm. The second algorithm is given in Section 3. In Section 4 a comparison between these two algorithms is given.

Manuscript received January 08, 2015; revised January 20, 2015. The work is part of the preparation of doctorate in computer science.

A. Belabbaci is with the Department of Mathematics and Computer Science, University of Laghouat, Algeria. e-mail: (amelbelabbaci@gmail.com, amel.belabbaci@univ-usto.dz).

B. Djebbar is with the Department of Mathematics and Computer Science, University of Science and Technology, Oran, Algeria. e-mail: (badj2001@yahoo.fr).

Fig. 1. Algotithm 1 for optimization of a convex quadratic function under linear constraints

```
Require: • The initial vertex $S_0 = (0, 0, \cdots, 0)$,
    • $C$ the set of neighbouring vertices of the vertex $S_0$,
    • $\Delta_0$ the initial value of the objective function,
    • $\gamma_{ij}$ the coefficients of double product,
    • the coefficients $\alpha$ and $\beta$,
    • the matrix $A$ and the vector $b$.
Ensure: The optimal solution
  Begin
  $\gamma_{ij} = 0$ for all $i,\ j$
  $\Delta_0 = 0$
  if ($a_{ki} < 0,\ \forall k, i$) then
    The quadratic program has not an optimum.
  else
    repeat
      Calculate $\theta_{i_0} = \min_{\substack{a_{ik}>0 \\ k}} \frac{b_k}{a_{kj}}$ for each $k = 1 \cdots m$ and
      $i = 1 \cdots n$. $x_{i_0}$ is then the condidate leaving variable;
      Choose an entering variable $x_j$;
      Calculate the new coefficients for $A,\ b,\ \alpha,\ \beta$ and $\gamma$;
      if (the resultant vertex $S_j$ increases the value of the
      objective function) then
        Move to $S_j$;
        Reset $C$;
      else
        $C = C - S_j$;
      end if
    until ($C = \varnothing$)
  end if
  End.
```

II. The First Algorithm

The first algorithm permits the passage from an extreme point to another one better as in the simplex method: We begin by converting the program into its standard form by adding non-negative slack variables. A complete study of the algorithm is given in [1]. We give here a short description. For choosing the entering variable which must increase the objective function we choose the vector that gives the greatest marginal gain that is, for each entering variable, we calculate $\theta_j = \min_i \left(\frac{a_{ij}}{b_j}\right)$, the marginal gain $\Delta_j = \alpha_j\theta_j + \beta_j\theta_j^2$ for each θ_j, and we choose $\Delta_{j_0} = \max_j \left\{\alpha_j\theta_j + \beta_j\theta_j^2\right\} = \alpha_{j_0}\theta_{j_0} + \beta_{j_0}\theta_{j_0}^2$. The vector x_{j_0} is then the entering vector which replaces the leaving vector. Note that others choices are possible provided for increasing the objective function.

The matrix of the constraints A and the vector b will change in the same way as in the simplex method.

After the first passage, the form of the objective function

TABLE I
THE FIRST TABLE (EXAMPLE1)

3	-2	0	0	0	α
1	1	0	0	0	β
2	2				θ
10	0				Δ
1	1	1	0	0	$x_3 = 2$
2	1	0	1	0	$x_4 = 4$
-3	2	0	0	1	$x_5 = 6$

becomes:

$$f(x) = \Delta_0 + \sum_{j \neq j_0} \alpha_j^{'} x_j + \sum_{j \neq j_0} \beta_j^{'} x_j^2 + \sum_j \sum_i \gamma_{ji} x_j x_i$$

where Δ_0 is the cost of the objective function and the coefficients can be calculated with the following formulas:

$$\acute{\alpha_j} = \alpha_j - (\alpha_{j_0} + 2\beta_{j_0}\theta_0) \frac{a_{i_0 j}}{a_{i_0 j_0}} + \frac{b_{i_0}}{a_{i_0 j_0}} (\gamma_{j_0 j} + \gamma_{j j_0})$$
$$\text{if } j \neq j_0 \text{ and } \acute{\alpha_{j_0}} = 0$$

$$\acute{\beta_j} = \beta_j + \frac{\beta_{j_0}}{(a_{i_0 j_0})^2} (a_{i_0 j})^2 - \frac{1}{a_{i_0 j_0}} a_{i_0 j} (\gamma_{j_0 j} + \gamma_{j j_0})$$
$$\text{if } j \neq j_0 \text{ and } \acute{\beta_{j_0}} = 0$$

$$\acute{\gamma_{ji}} = \gamma_{ji}^{*} - \frac{a_{i_0 j}}{a_{i_0 j_0}} (\gamma_{j_0 i} + \gamma_{i j_0})$$
$$\text{for } j \neq i, j \neq j_0, i \neq j_0$$
$$\text{and } \gamma_{j_0 i} = \gamma_{i j_0} = \gamma_{ii} = 0 \ \forall i = 1, 2, \cdots$$

where $\gamma_{ji}^{*} = \beta_{j_0} \left(\frac{\theta_0}{b_{i_0}}\right)^2 a_{i_0 j} a_{i_0 i}$.
Note that the terms due to the double products γ_{ij} have no effect for the choice of the entering and the leaving variables. However, they are used for computing the new coefficients α_i and β_i and the new value of the objective function.

A. Examples

To illustrate the algorithm, we consider the following examples solved in [1]. We represent only the coefficients α_i and β_i of the objective function at each step. We have used the greatest marginal gain.

Example1: We consider the following program:

$$\begin{cases} x \geq 0 \\ x_1 - x_2 + x_3 = 2 \\ 2x_1 + x_2 + x_4 = 4 \\ -3x_1 + 2x_2 + x_5 = 6 \\ \max f(x) = 3x_1 - 2x_2 + x_1^2 + x_2^2 \end{cases}$$

From the first table (table I), the entering variable is x_1 and the leaving one is x_3. This yields table II The objective function is written as follows:
$f(x) = -9x_2 - 7x_3 + 2x_2^2 + x_3^2 + 2x_2x_3 + 10$ and the cost of f equals 10.
From table II we see that no entering variable can be chosen, and thus it is impossible to obtain a better solution. The optimal solution is then reached at the vertex $(2, 0, 0)$.

TABLE II
RESULT AFTER THE FIRST ITERATION (EXAMPLE1)

0	-9	-7	0	0	α
0	2	1	0	0	β
	2	2			θ
	-10	-10			Δ
1	1	1	0	0	$x_1 = 2$
0	-1	-2	1	0	$x_4 = 0$
0	5	3	0	1	$x_5 = 12$

TABLE III
THE FIRST TABLE (EXAMPLE2)

1	2	0	0	0	α
1	3	0	0	0	β
6	3				θ
42	33				Δ
-1	1	1	0	0	$x_3 = 3$
1	-1	0	1	0	$x_4 = 6$
1	2	0	0	1	$x_5 = 12$

TABLE IV
RESULT AFTER THE FIRST ITERATION (EXAMPLE2)

0	15	0	-13	0	α
0	4	0	1	0	β
	2		6		θ
	46		-42		Δ
0	0	1	1	0	$x_3 = 9$
1	-1	0	1	0	$x_1 = 6$
0	3	0	-1	1	$x_5 = 6$

TABLE V
RESULT AFTER THE SECOND ITERATION (EXAMPLE2)

0	0	0	$\frac{-20}{4}$	$\frac{-31}{3}$	α
0	0	0	$\frac{7}{9}$	$\frac{4}{9}$	β
			9	6	θ
			3	-46	Δ
0	0	1	1	0	$x_3 = 9$
1	0	0	$\frac{2}{3}$	$\frac{1}{3}$	$x_1 = 8$
0	1	0	$\frac{-1}{3}$	$\frac{1}{3}$	$x_2 = 2$

Example2: For the following program:

$$\begin{cases} x \geq 0 \\ -x_1 + x_2 \leqslant 3 \\ x_1 - x_2 \leqslant 6 \\ x_1 + 2x_2 \leqslant 12 \\ \max f(x) = x_1 + 2x_2 + x_1^2 + 3x_2^2 \end{cases}$$

We obtain:

From table III the entering variable is x_1, and the leaving one is x_4. This yields table IV.The objective function is written as follows:
$f(x) = 15x_2 - 13x_4 + 4x_2^2 + x_4^2 - 2x_2x_4 + 42$ and the cost of f equals 42. From this table (table IV) the entering variable is x_2, and the leaving one is x_5, which yields table V.

The objective function is written as follows:
$f(x) = \frac{-20}{4}x_4 - \frac{31}{3}x_5 + \frac{7}{9}x_4^2 + \frac{4}{9}x_5^2 - \frac{2}{9}x_4x_5 + 88$ and the cost of f equals 88.
From this table (table V) the entering variable is x_4, and the leaving one is x_3. This yields the last table (table VI).

TABLE VI
RESULT AFTER THE THIRD ITERATION (EXAMPLE2)

0	0	$\frac{-22}{3}$	0	$\frac{-37}{3}$	α
0	0	$\frac{7}{9}$	0	$\frac{4}{9}$	β
		9		6	θ
		-3		-58	Δ
0	0	1	1	0	$x_4 = 9$
1	0	$\frac{-2}{3}$	0	$\frac{1}{3}$	$x_1 = 2$
0	1	$\frac{1}{3}$	0	$\frac{1}{3}$	$x_2 = 5$

The objective function is written as follows:
$f(x) = \frac{-22}{3}x_3 - \frac{37}{3}x_5 + \frac{7}{9}x_3^2 + \frac{4}{9}x_5^2 + \frac{2}{9}x_3x_5 + 91$ and the cost of f equals 91. Now, no entering variable can be chosen and thus it is impossible to obtain a better value of f than 91. The optimal solution is then reached at the vertex $x = (2, 5, 0)$.

III. ALGORITHM OF SEPARATION AND ELIMINATION

In this section we give a second algorithm for searching the optimal solution of a quadratic convex program.

Let $x^* = \frac{-\alpha_i}{2\beta_i}$ be the critical point of the function f . For a given vertex x of Ω, we look for another neighbouring vertex $x^{'}$ which is the farthest from x^* than x (if there exist more than one vertex we chose arbitrary one of them). All the other vertices are eliminated. We continue looking for the neighbouring of the current vertex and moving to the farthest one from x^* until there is no vertex to move to. The optimal solution is then reached.

Fig. 2. Algorithm for finding the farthest vertex to x^*.

```
Require: Critical point $x^*$, an initial vertex $s_0$.
Ensure: The optimal solution.
  Begin
  stop:= false;
  $D_1 = \|s_0 - x^*\|^2$;
  while (stop=false) do
    $D_2 = \max\|s_i - x^*\|^2$, where $s_i$ are the neighbours of $s_0$
    if ($D_1 > D_2$) then
      stop:=true;
    else
      let $s_1$ be the farthest vertex with the distance $D_2$;
      $s_0 := s_1$;
      move to $s_0$;
    end if
  end while
  End.
```

A. Example

To illustrate this second algorithm, the following example is solved.
We consider the following program:

$$\begin{cases} x \geq 0 \\ x_1 + x_2 \leqslant 2 \\ 2x_1 + x_2 \leqslant 4 \\ -3x_1 + 2x_2 \leqslant 6 \\ \max\ P(x) = 3x_1 - 2x_2 + x_1^2 + x_2^2 \end{cases}$$

The convex Ω is the set: $\{x = (x_1, x_2) \in \mathbb{R}^3 : x_1 + x_2 \leqslant 2, 2x_1 + x_2 \leqslant 4 \text{ and } -3x_1 + 2x_2 \leqslant 6\}$. The set of

TABLE VII
THE FIRST TABLE

	x_1	x_2	x_3	x_4	x_5	
x_3	1	1	1	0	0	2
x_4	2	1	0	1	0	4
x_5	-3	2	0	0	1	6

TABLE VIII
RESULT AFTER THE FIRST ITERATION

	x_1	x_2	x_3	x_4	x_5	
x_1	1	1	1	0	0	2
x_4	0	-1	-2	1	0	0
x_5	0	5	3	0	1	12

vertices of this convex is $\{(2,0),(0,0),(0,3),(2.85,3.43)\}$. Coefficients $\beta_i = 1, i = 1,2$; the critical point $x^* = (\frac{-3}{2}, 1)$. We start with the initial vertex $s_0 = (0,0)$, the vertices $s_1 = (2,0), s_3 = (0,3)$ are the neighbouring of s_0.
Next, we have $\|x^* - s_0\|^2 = 1.80$, $\|x^* - s_1\|^2 = 3.64$, and $\|x^* - s_3\|^2 = 2.5$. Then, we pass from s_0 to s_1. We obtain the second table (table VIII) from which we see that the neighbouring vertices of s_1 are $\{s_2 = (0,2), s_0 = (0,0)\}$.

Next $\|x^* - s_2\|^2 = 1.8$ and so s_1 is farther from x^* than s_2. This table is the last one to be considered and the optimal vertex is reached at s_1.

IV. COMPARISON AND DISCUSSION

We give here only a comparison in terms of specificities between the two algorithms which appears schematically in table IX. Both algorithms are simple and do not require a lot of calculus to reach the optimal solution. However, there are some differences.
The algorithm of separation is applied only for strictly convex problems whereas the first algorithm treats convex problems in general, and gives the form of the objective function after each passage from a vertex to another one. The first algorithm requires no transformation except the transformation from canonical form to standard one and adding non-negative slack variables. The algorithm of separation requires that all the coefficients β_i are equal to one otherwise the convex must be transformed (see [7]).
Although the first algorithm has the advantage of simplicity, there is no specific rule to select an entering variable: The choice of an entering variable must guarantee the increase of the objective function. A lot of tests applied on vertices is then needed before choosing an entering variable . This disadvantage will have an impact on the time complexity of the algorithm because time is spent to avoid choosing a vertex that decreases the value of the objective function.

The algorithm of separation has a remarkable advantage compared to the first algorithm. It is well adapted when the initial basic solution is unknown: no need to look for an initial base solution to start searching the optimal vertex.

V. CONCLUSION

In this paper we have presented two algorithms for searching the optimal solution of a convex quadratic program. Both are simple and easy to use and each has its relative

TABLE IX
SPECIFICAL COMPARISON BETWEEN THE TWO ALGORITHMS

Algorithm / Criteria	first algorithm	separation
convexity	general convex program	strictly convex program
Stopping condition	no growth for objective function	the farthest point from the critical point
Objective function	the form after each passage	no form is given
Transformations	from canonical to standard form	requires transformation of the convex, in general

advantages. To compare the algorithms in a set of instances and to compare these algorithms with others known in the literature needs more time and further work. This will be done later.

REFERENCES

[1] A. Belabbaci, “ La programmation quadratique et son application à la programmation séparable,” *mémoire du magister, 2007*, Université de Laghouat, Algérie

[2] A. Belabbaci, B. Djebbar, and A. Mokhtari, “Optimization of a strictly concave quadratic form, ” *Lecture Notes in Management Science*, Vol. 4, pp. 16–20, Jul. 2012.

[3] A. Chikhaoui, B. Djebbar, A. Mokhtari, and A. Belabbaci, “Nouvelle Méthode de Programmation Quadratique, France, 2011 ”, *12e congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision ROADEF, Saint-Étienne.* uma.ensta-paristech.fr/files/.../ roadef2011_submission_102.Pdf.

[4] S. J.Chang and K. G.Murty, “Polynomial bounded ellipsoid algorithm for convex quadratic programming,” *Nonlinear programming, 1981*, Vol. 4, no. 0, pp. 63-66.

[5] S. Mehotra and j. Sun, “An algorithm for convex quadratic programming that requires O(n 3.5 L) arithmetic operations,” *Mathematics of operations research, 1990*, Vol. 15, no, 2, pp. 342-363.

[6] M. Ben-Daya ,K. S.A1-Sultan, “A new penalty function algorithm for convex quadratic programming,” *European Journal of Operational Research, 1997*, Vol. 101, no. 1, pp. 155-163.

[7] A. Chikhaoui,B. Djebbar,A. Belabbaci,A. Mokhtari, “ Optimization of a quadratic function under its canonical form, ” *Asian journal of applied sciences, 2009*, Vol. 2, no. 6, pp.499-510.