A Logarithmic Barrier Method Based On a New Majorant Function for Convex Quadratic Programming

Soraya Chaghoub and Djamel Benterki

***Abstract*—In this paper, we present a logarithmic barrier method based on a new majorant function for solving a convex quadratic program. The proposed majorant function allows the computation of the displacement step easily and in a short time, unlike the line search method which is expensive in terms of computational volume and necessitates much time. To compare the proposed majorant function's performance against that of line search, we conducted numerical experiments on numerous collections of test problems. The computational results indicate the efficiency and the accuracy of our new majorant function.**

***Index Terms*—quadratic programming, interior point methods, line search, majorant function**

I. Introduction

QUADRATIC programming is a particular type of nonlinear programming, and it appears in many areas of applications, such as in finance, agriculture, economics, optimal control and geometric problems. For solving such problems, various methods and techniques have been proposed and developed. The Frank-Wolfe method [14], which is one of the first powerful algorithms used to solve nonlinear optimization problems. Another method of Frank-Wolfe was used in [24]; this method consists of transforming the problem to a linear one to apply then the simplex method. However, this method needs to add an essential number of constraints and artificial variables. In 2004, Dozzi [12] has treated an example with two variables, using the simplex method and manipulated ten variables. Many other methods exist (see [5]), and all these methods give approximate solutions. Gärtner and Schönherr [15] developed a method that provides an exact solution. This method is considered as a generalization of the simplex method for quadratic programming. Still, this method does not apply to any problem because it must be dense and have a few variables or constraints. Yunong and Jun [25] proposed a neural network called dual neural network for convex quadratic programming subject to linear equality and inequality constraints. In 2009, Chikhaoui et al. [9] proposed an algorithm to resolve a quadratic function under its canonical form. Elias and Santosh [13] introduced a new heuristic for convex quadratic programming. In 2012, Belabbaci et al. [3] proposed a new algorithm for finding the exact optimal solution without introducing any other variables. Besides, in 2017, Belabbaci and Djebbar [4] proposed a method of separation and elimination based on the principle of concentric spheres or ellipsoids.

However, during the past two decades, and based on the famous Karmarkar's method [23], the most dramatic progress in computational optimization has been achieved in the implementation of interior point methods ([20],[22]). Roumili and Boudjellal [21] provided an infeasible interior point method for convex quadratic problems. In particular, feasible primal-dual path-following methods are considered the most popular among interior point methods. The derived algorithms of these methods are efficient and have polynomial complexity. These algorithms trace approximately the so-called central path. This central path is a curve that lies in the feasible region of the problem at hand, and then they reach an optimal solution to this problem. In 2006, Achache [1] derived an algorithm based on new techniques for finding a new class of search directions. The primal-dual interior point methods based on the kernel function technique are extensively studied, such as linear optimization (LO) Bai et al. [2]. They provided a large class of eligible kernel functions. In 2020, Boudjellal et al. [7] have also proposed a primal-dual interior point algorithm for the convex quadratic problem (CQP). They also provide numerical tests to show the efficiency of their proposed approach. Roumili and Boudjellal [21] also provided an infeasible interior point method for convex quadratic problems.

In parallel and independently of previous works, different logarithmic barrier interior point methods based on majorant or minorant function techniques were considered. Crouzeix and Merikhi [10], are the first to introduce a logarithmic barrier algorithm based on majorant functions for semidefinite programming. Inspiring from [10], Menniche and Benterki [19] proposed a barrier method based on new majorant functions for linear programming. Bachir Cherif and Merikhi [8] extended this idea to provide a mojorant function for nonlinear programming. On the other hand, Leulmi et al. [18], and [17] proposed minorant functions for semidefinite programming and linear programming, respectively.

Inspiring previous works, we are interested in optimizing a convex quadratic function under linear inequality constraints. Our aim consists of elaborating an efficient and straightforward logarithmic barrier method based on a new majorant function. This majorant function's object is to compute the displacement step in the algorithm quickly and simply and reduce the computation time required by the line search method.

Manuscript received December 31, 2020; revised May 20, 2021.

Soraya Chaghoub is a PhD candidate of School of Mathematical Science & Institute of Mathematics, Nanjing Normal University, Nanjing 210023, China (corresponding author e-mail: chaghoubsoraya@yahoo.fr).

Djamel Benterki is a professor of Laboratory of Fundamental and Numerical Mathematics, Setif-1 Ferhat Abbas University, Setif 19000, Algeria (e-mail: djbenterki@univ-setif.dz).

The paper is organized as follows. In section 2, we introduce the problem at hand and its associated perturbed problem; then, we prove the later problem's convergence into the initial one. In section 3, we are interested in resolving the perturbed problem. We present our main result by introducing a new majorant function to compute the displacement step of the obtained logarithmic barrier algorithm. In section 4, we describe our algorithm briefly and provide a computational study. The conclusion is summarized in the last section.

II. Position of the problem

Consider the following convex quadratic problem:

$$\begin{cases} \min q(x) = \frac{1}{2}x^tQx + c^tx \\ x \in D, \end{cases} \qquad (PQ)$$

where Q is a $\mathbb{R}^{n\times n}$ symmetric semidefinite matrix, $c \in \mathbb{R}^n$ and $D = \{x \in \mathbb{R}^n : Ax \geq b\}$, such that $b \in \mathbb{R}^m$ and A is a $\mathbb{R}^{m\times n}$ matrix.

1) The matrix A has full row rank ($rank(A) = m < n$).
2) (PQ) satisfies the interior-point condition, i.e., there exist $x^0 \in \mathbb{R}^n$ such that:
$$Ax^0 > b. \qquad (1)$$
3) The set of optimal solutions of the problem (PQ) is nonempty and bounded.

Recall that the scalar product of $x, y \in \mathbb{R}^n$ is given by:

$$< x, y >= x^ty = \sum_{i=1}^{n} x_iy_i$$

and the Euclidean norm of y is

$$\|y\| = \sqrt{< y, y >} = \sqrt{\sum_{i=1}^{n} y_i^2}$$

A. The perturbed problem of (PQ)

Let (PQr) be the unconstrained perturbed problem associated with (PQ). This problem takes the form.

$$\begin{cases} \min q_r(x) \\ x \in \mathbb{R}^n, \end{cases} \qquad (PQr)$$

where $q_r : \mathbb{R}^n \to (-\infty, +\infty]$ is a barrier function defined by:

$$q_r(x) = \begin{cases} q(x) - r\sum\limits_{i=1}^{m} \ln < e_i, Ax - b > & \text{if } Ax - b > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Where $(e_1, e_2, ..., e_m)$ is the canonical base in $\mathbb{R}^m$.

We know that (PQ) is convex and by assumption (3) its solutions set is nonempty and bounded, then according to Bachir Cherif and Merikhi [8], the strictly convex problem (PQr) admits a unique optimal solution x_r^*, for each $r > 0$.

We aim to solve the problem (PQr) since the resolution of the problem (PQ) is equivalent to the resolution of (PQr) when r tends to 0.

Firstly, we need to study the convergence of (PQr) to (PQ).

B. Convergence of the perturbed problem (PQr)

Let $r > 0$, for all $x \in D$, we define $\phi(x, r) = q_r(x)$.

Lemma 1: Let $r > 0$. If x_r is an optimal solution of the problem (PQr), such that $\lim\limits_{r\to 0} x_r = x^*$, then x^* is an optimal solution of the problem (PQ).

Proof: It follows from the necessary and sufficient conditions and the differentiability of the function ϕ at the point (x_r, r) that:

$$\nabla_x\phi(x_r, r) = \nabla q_r(x_r) = 0$$

and for all x checking $Ax - b > 0$, we have:

$$\begin{aligned} q(x) &= \phi(x, 0) \geq \phi(x_r, r) + < x - x_r, \nabla_x\phi(x_r, r) > \\ &+ (0 - r)\phi_r'(x_r, r) \end{aligned}$$

then,

$$\begin{aligned} q(x) &\geq q(x_r) - r\sum_{i=1}^{m} \ln < e_i, Ax - b > \\ &+ r\sum_{i=1}^{m} \ln < e_i, Ax - b >= q(x_r) \end{aligned}$$

hence

$$\min_{x\in D} q(x) \geq q(x_r), \forall r > 0,$$

on the other hand we have

$$\min_{x\in D} q(x) \leq q(x_r), \forall r > 0,$$

therefore,

$$\min_{x\in D} q(x) = \lim_{r\to 0} q(x_r) = q(x^*)$$

which implies the claimed result. ∎

III. Resolution of the perturbed problem

Based on the necessary and sufficient optimality conditions of the convex problem (PQr), x_r is an optimal solution of (PQr) if and only if it satisfies the nonlinear system:

$$\nabla q_r(x_r) = 0. \qquad (2)$$

To solve this system, we propose a logarithmic interior point method based on Newton's approach, which consists of constructing a sequence $((x_r)_k = x_k)$ of interior points, such that this sequence converges into the optimal solution of (PQ). The iteration of Newton is defined by $x_{k+1} = x_k + d_k$, where d_k is the descent direction solution of the linear system

$$\nabla^2 q_r(x_r)d_k = -\nabla q_r(x_r) \qquad (3)$$

Note this approach does not ensure the feasibility of the interior points x_k generated in each iteration of the algorithm, i.e., there is no guarantee that $A(x_k + d_k) > b$. We introduce a displacement step α_k to remedy this difficulty. The iteration becomes: $x_{k+1} = x_k + \alpha_kd_k$.

There are two main techniques used for computing the displacement step α_k:

1) **Line search methods**: such as Wolfe method, Goldstein-Armijo method, Fibonacci method, etc. These methods are based on the minimization of the unidimensional function

$$\varphi(\alpha) = \min_{\alpha>0} q_r(x_r + \alpha d)$$

Unfortunately, they are very delicate and time-consuming.

2) **Majorant function**: the technique of the majorant function was first proposed by Crouzeix and Merikhi [10] for the positive semidefinite programming. This technique relies on approximating the function

$$\theta(\alpha) = \frac{1}{r}(q_r(x_r + \alpha d) - q_r(x_r))$$

by another function whose minimum can easily be computed, which permits the computation of the displacement step at each iteration in a relatively short time and with a smaller number of instructions in contrast to line search technique.

We start with the following lemma, and in the rest of the paper we consider x instead of x_r.

Lemma 2: For all $\alpha \in [0, \tilde{\alpha}[$, such that $\tilde{\alpha} = \min_{i\in I_-}\{\frac{-1}{y_i}\}$ and $I_- = \{i : y_i < 0\}$, the function θ can be written as follows:

$$\begin{aligned}\theta(\alpha) &= \alpha\left(\sum_{i=1}^{m} y_i - \|y\|^2\right) - \sum_{i=1}^{m}\ln(1+\alpha y_i)\\ &+ \frac{1}{r}(\frac{1}{2}\alpha^2 d^tQd - \alpha d^tQd).\end{aligned}$$

Where $y_i = \frac{<e_i,Ad>}{<e_i,Ax-b>}, i \in \{1, ..., m\}$.

Proof:

$$\begin{aligned}\theta(\alpha) &= \frac{1}{r}(q_r(x+\alpha d) - q_r(x))\\ &= \frac{1}{r}\left(\frac{1}{2}(x+\alpha d)^tQ(x+\alpha d) + c^t(x+\alpha d)\right)\\ &- \frac{1}{2r}x^tQx - \frac{1}{r}c^tx\\ &- \sum_{i=1}^{m}\ln\left(1+\alpha\frac{< e_i, Ad >}{< e_i, Ax-b >}\right)\\ &= \frac{1}{r}\left(\frac{1}{2}\alpha d^tQx + \frac{1}{2}\alpha x^tQd + \frac{1}{2}\alpha^2 d^tQd\right)\\ &- \sum_{i=1}^{m}\ln\left(1+\alpha\frac{< e_i, Ad >}{< e_i, Ax-b >}\right) + \frac{1}{r}\alpha c^td\end{aligned}$$

As Q is symmetric then $x^tQd = d^tQx$, then

$$\begin{aligned}\theta(\alpha) &= \frac{1}{r}\left(\alpha d^tQx + \frac{1}{2}\alpha^2 d^tQd + \alpha c^td\right)\\ &- \sum_{i=1}^{m}\ln\left(1+\alpha\frac{< e_i, Ad >}{< e_i, Ax-b >}\right)\end{aligned}$$

we have also

$$\nabla q_r(x) = Qx + c - r\sum_{i=1}^{m}\frac{A^te_i}{< e_i, Ax-b >} \quad (4)$$

and

$$\nabla^2 q_r(x) = Q + r\sum_{i=1}^{m}\frac{A^te_i(A^te_i)^t}{(< e_i, Ax-b >)^2}$$

and from (3)

$$d^t\nabla q_r(x) = -d^t\nabla^2 q_r(x)d$$

we obtain

$$d^tQx + d^tc - rd^t\sum_{i=1}^{m}\frac{A^te_i}{<e_i,Ax-b>} = -d^tQd - r\sum_{i=1}^{m}\frac{<e_i,Ad>^2}{<e_i,Ax-b>^2}$$

then

$$d^tQx + d^tc = -d^tQd - r\sum_{i=1}^{m}\frac{< e_i, Ad >^2}{< e_i, Ax-b >^2} + rd^t\sum_{i=1}^{m}\frac{A^te_i}{< e_i, Ax-b >}$$

which gives

$$\begin{aligned}\theta(\alpha) &= \frac{1}{r}\Big(\frac{1}{2}\alpha^2 d^tQd - \alpha d^tQd\\ &- r\alpha\sum_{i=1}^{m}\frac{< e_i, Ad >^2}{< e_i, Ax-b >^2}\\ &+ r\alpha d^t\sum_{i=1}^{m}\frac{A^te_i}{< e_i, Ax-b >}\Big)\\ &- \sum_{i=1}^{m}\ln\left(1+\alpha\frac{< e_i, Ad >}{< e_i, Ax-b >}\right)\\ &= \frac{1}{r}\left(\frac{1}{2}\alpha^2 d^tQd - \alpha d^tQd\right) + \alpha\left(\sum_{i=1}^{m} y_i - \|y\|^2\right)\\ &- \sum_{i=1}^{m}\ln(1+\alpha y_i).\end{aligned}$$

which implies the claimed result. ■

Now, we give the main result of the paper.

A. New majorant function

To introduce our new majorant function, we use the following well known inequality [19]:

$$\alpha\sum_{i=1}^{m} y_i + \alpha\|y\| + \ln(1-\alpha\|y\|) - \sum_{i=1}^{m}\ln(1+\alpha y_i) \le 0 \quad (5)$$

Lemma 3: For $\alpha \in I_\alpha = [0, \hat{\alpha}[\cap[0, \tilde{\alpha}[$, we have:

$$\theta(\alpha) \le \theta_C(\alpha).$$

Where θ_C is a majorant function of θ defined on $[0, \hat{\alpha}[$ with $0 < \hat{\alpha} < \frac{1}{\|y\|}$ by:

$$\theta_C(\alpha) = -\alpha(\|y\| + \|y\|^2) - \ln(1-\alpha\|y\|) + \frac{1}{2r}\hat{\alpha}^2 d^tQd.$$

Proof: For $\alpha \in I_\alpha$ and from inequality (5), we have

$$\begin{aligned}\alpha\sum_{i=1}^{m} y_i - \sum_{i=1}^{m}\ln(1+\alpha y_i) - \alpha\|y\|^2 &\le -\alpha\|y\| - \alpha\|y\|^2\\ &- \ln(1-\alpha\|y\|)\end{aligned}$$

So we have

$$\frac{1}{2r}\alpha^2 d^t Qd < \frac{1}{2r}\hat{\alpha}^2 d^t Qd, \forall \alpha \in I_\alpha,$$

and

$$-\alpha d^t Qd \leq 0, \forall \alpha \in I_\alpha,$$

this yields

$$\begin{aligned}\theta(\alpha) &= \alpha\sum_{i=1}^{m} y_i - \sum_{i=1}^{m}\ln(1+\alpha y_i) - \alpha\|y\|^2 \\ &+ \frac{1}{r}(\frac{1}{2}\alpha^2 d^t Qd - \alpha d^t Qd) \\ &\leq -\alpha(\|y\|+\|y\|^2) - \ln(1-\alpha\|y\|) \\ &+ \frac{1}{2r}\hat{\alpha}^2 d^t Qd = \theta_C(\alpha)\end{aligned}$$

Hence

$$\forall \alpha \in I_\alpha, \theta(\alpha) \leq \theta_C(\alpha).$$

which implies the claimed result. ■

Remark 1: We note that $\theta_C^{''}(\alpha) = \frac{\|y\|^2}{(1-\alpha\|y\|)^2} \geq 0,\ \forall \alpha \in [0, \hat{\alpha}[$, hence θ_C is convex and if it admits a minimum, this minimum is global.

1) Minimization of the majorant function: θ_C is defined and convex on $[0, \hat{\alpha}[$, then its global minimum is reached when $\theta_C^{'}(\alpha) = 0$. Therefore, finding the minimum of the function θ_C is equivalent to solving the equation $\theta_C^{'}(\alpha) = 0$. The solution of this later is the root of the equation:

$$\alpha(\|y\|^2+\|y\|^3) - \|y\|^2 = 0 \tag{6}$$

The root of the equation (6) is

$$\alpha^* = \frac{1}{1+\|y\|} \in I_\alpha,$$

which is the global minimum of the function θ_C.

The logarithmic barrier algorithm is described as follows.

IV. Description of the algorithm

In the following, we give a brief description of the logarithmic barrier algorithm.

Algorithm 1 Logarithmic barrier algorithm (LB)

Data: A quadratic program (PQ) and its system of inequalities $Ax \geq b$, a strictly feasible solution x_0 of (PQ), $r > 0$, ε a given precision, $k = 0$.
Result: Optimal solution x^* of (PQ).
while $\|\nabla q_r(x_k)\| > \varepsilon$ **do**
  -Solve the linear system $\nabla^2 q_r(x_k)d_k = -\nabla q_r(x_k)$.
  -Compute the displacement step α_k.
  -Set $x_{k+1} = x_k + \alpha_k d_k$ and $k = k + 1$.
  -Take $r = \rho r, 0 < \rho < 1$.
end while
Return x^*.

The following Lemma indicates that the interior point x_{k+1} generated in each iteration k of the algorithm (LB) ensures the decrease of the function q_r.

Lemma 4: The function q_r significantly decrease from iteration k to iteration $k+1$, that is, if x_k and x_{k+1} are two feasible solutions obtained at iteration k and $k+1$ respectively, then

$$q_r(x_{k+1}) < q_r(x_k).$$

Proof: Let x_k and x_{k+1} be two feasible solutions obtained at iteration k and $k+1$ respectively, we have

$$q_r(x_{k+1}) \simeq q_r(x_k) + < \nabla q_r(x_k), x_{k+1} - x_k >$$

and

$$x_{k+1} = x_k + \alpha_k d_k$$

then

$$\begin{aligned}q_r(x_{k+1}) - q_r(x_k) &\simeq < \nabla q_r(x_k), \alpha_k d_k > \\ &\simeq -\alpha_k < \nabla^2 q_r(x_k)d_k, d_k > < 0\end{aligned}$$

Hence,

$$q_r(x_{k+1}) < q_r(x_k).$$

which implies the claimed result. ■

A. Computational study

To measure the performance of the proposed method, we present a comparison of the results obtained by the proposed algorithm (LB) using our new majorant function to compute the displacement step and those obtained by using the line search Wolfe's method. We use examples with fixed and variable sizes to carry out the numerical tests.

1) Example with fixed size: The examples with fixed size considered in TABLE I are randomly generated by MATLAB.

2) Example with variable size: The following examples of variable size are taken from the literature.

Example 4.1: $n = 2m, A[i,j] = 0$ if $i \neq j$ or $(i+1) \neq j$, $A[i,i] = A[i,i+m] = 1,\ c[i] = -1$, $b[i] = 2,\ c[i+m] = 0$, for $i = 1,..,m.\ j = 1,..,n.$

$$Q[i,j] = \begin{cases} 2j-1 & i > j \\ 2i-1 & i < j \\ i(i+1)-1 & i = j,\ i,j = 1,..,n. \end{cases}$$

Example 4.2: $n = 2m,\ A[i,j] = 0$ if $i \neq j$ or $(i+1) \neq j$, $A[i,i] = A[i,i+m] = 1,\ c[j] = j$, $b[i] = \frac{i+1}{2}$, for $i = 1,..,m.\ j = 1,..,n.$

$$\begin{cases} Q[1,1] = 1 \\ Q[i,i] = i^2+1 \\ Q[i,i-1] = Q[i-1,i] = i & i = 2,..,n \end{cases}$$

Example 4.3: $n = 2m,\ A[i,j] = 0$ if $i \neq j$ or $(i+1) \neq j,\ A[i,i] = A[i,i+m] = 1,\ c[j] = \frac{j+1}{2}$, $b[i] = 4$ for $i = 1,..,m.\ j = 1,..,n.$

$$\begin{cases} Q[1,1] = Q[n,n] = 1 & \\ Q[i,i] = 4 & i = 2,..,n-1 \\ Q[i,i-1] = A[i-1,i] = 1 & i = 2,..,n \end{cases}$$

Example 4.4: $n = 2m,\ A[i,j] = 0$ if $i \neq j$ or $(i+1) \neq j$, $A[i,i] = A[i,i+m] = 1, c[j] = 2j$, $b[i] = i^2$, for $i = 1,...,m.\ j = 1,..,n.$

$$Q[i,j] = \frac{1}{i+j} \text{ for } i,j = 1,..,n.$$

TABLE I: Example with fixed size

	$method$	LS	$method$	MF
$ex\,(m,n)$	Itr	$time\,(s)$	Itr	$time\,(s)$
$ex1\,(2,4)$	7	0.087533	3	0.031245
$ex2\,(3,5)$	13	0.089912	9	0.041121
$ex3\,(3,6)$	19	0.103447	13	0.072488
$ex4\,(5,7)$	28	0.173312	17	0.080330
$ex5\,(6,10)$	29	0.222309	19	0.090031

TABLE II: Example with variable size (example 4.1)

	$method$	LS	$method$	MF
$ex\,(m,n)$	Itr	$time\,(s)$	Itr	$time\,(s)$
$(300,600)$	26	22.524009	11	6.922453
$(400,800)$	35	97.103447	14	41.842410
$(600,1200)$	48	224.321200	23	89.412109
$(1000,2000)$	51	497.011652	33	113.310561
$(1500,3000)$	78	1321.032778	40	305.021560

TABLE III: Example with variable size (example 4.2)

	$method$	LS	$method$	MF
$ex\,(m,n)$	Itr	$time\,(s)$	Itr	$time\,(s)$
$(300,600)$	38	22.524009	20	8.912566
$(400,800)$	45	88.112347	34	49.003558
$(600,1200)$	66	148.621033	37	66.655410
$(1000,2000)$	70	2004.112568	46	125.142130
$(1500,3000)$	101	2453.923121	77	344.614112

TABLE IV: Example with variable size (example 4.3)

	$method$	LS	$method$	MF
$ex\,(m,n)$	Itr	$time\,(s)$	Itr	$time\,(s)$
$(300,600)$	21	34.322156	14	7.752230
$(400,800)$	31	172.325501	22	71.402198
$(600,1200)$	53	503.443160	33	111.312557
$(1000,2000)$	67	2033.500621	41	168.099711
$(1500,3000)$	132	3121.113031	59	386.741103

TABLE V: Example with variable size (example 4.4)

	$method$	LS	$method$	MF
$ex\,(m,n)$	Itr	$time\,(s)$	Itr	$time\,(s)$
$(300,600)$	16	29.123309	7	6.433098
$(400,800)$	27	84.150012	13	29.712501
$(600,1200)$	55	153.922103	34	77.001290
$(1000,2000)$	68	2213.114310	41	191.511230
$(1500,3000)$	124	3121.113031	54	375.265401

In the above tables, we reported the results obtained by implementing the algorithm (LB) in MATLAB.
We denote by:
-$ex(m,n)$: the example of m constraints and n variables.
-LS: the strategy that uses line search of Wolfe.
-MF: the strategy that uses majorant function.
-Itr: the number of iterations needed to find an optimal solution.
-$time$: runtime in seconds.

3) Comments: From the above tables, we conclude the proposed method is more effective than the line search, and it can improve the results obtained by the line search method. This is especially true when the instances get larger. Besides, the improvement in time is significant, since we can easily see the time needed to get the optimal solution by the line search method is at least twice the time required by the proposed method.

V. Conclusion

In this paper, we addressed a convex quadratic problem with inequality constraints. We used a logarithmic barrier method and proposed a new majorant function to compute the displacement step, and we showed the the technique of the majorant function is more effective than the line search technique.

Our future exciting work is to improve the computational time further of the logarithmic barrier algorithm by proposing a more efficient majorant function. But extensions would be envisaged to the nonlinear, not necessarily to the quadratic optimization problem.

References

[1] Achache, M., A new primal-dual path-following method for convex quadratic programming. Computational and Applied Mathematics, V25, N1, (2006)

[2] Bai, YQ., El Ghami, M., Roos, C., A comparative study of kernel functions for primal-dual interior point algorithms in linear optimization. SIAM J. Optim., V15, N1, pp. 101-128, (2004)

[3] Belabbaci, A., Djebbar, B., Mokhtari, A., Optimization of a strictly concave quadratic form. International Conference on Applied Operational Research, Bangkok, Thailand, pp. 16-20, (2012)

[4] Belabbaci, A., Djebbar, B., Algorithm and separating method for the optimization of quadratic functions. Int. J. Computing Science and Mathematics, V8, N2, pp. 183-192, (2017)

[5] Ben-Daya, M., Al-Sultan, K.S., A new penalty function algorithm for convex quadratic programming. European Journal of Operational Research, V101, N1, pp. 155-163, (1997)

[6] Benterki, D., Crouzeix, J.P., Merikhi, B., A numerical feasible interior point method for linear semidefinite programs. RAIRO-Operations Research, V41, pp. 49-59, (2007)

[7] Boudjellal, N., Roumili, H., Benterki, D., A primal-dual interior point algorithm for convex quadratic programming based on a new parametric kernel function. Optimization, Taylor and Francis, DOI: 10.1080/02331934.2020.1751156

[8] Cherif, L.B., Merikhi, B., A penality method for nonlinear programming. RAIRO-Operations Research, V53, pp. 29-38, (2019)

[9] Chikhaoui, A., Djebbar, B., Belabbaci, A., Mokhtari, A., Optimization of a quadratic function under its canonical form. Asian Journal of Applied Sciences, Malaysia, V2, N6, pp. 499-510, (2009)

[10] Crouzeix, J.P., Merikhi, B., A logarithm barrier method for semidefinite programming. RAIRO-Operations Research, V42, pp. 123-139, (2008)

[11] Dikin, I.I., Iterative solution of problems of linear and quadratic programming. Doklady Akademiia Nauk SSSR, V174, pp. 747-748, (1967)

[12] Dozzi, M., Méthodes non linéaires et stochastiques de la recherche opérationnelle. http://iecl.univ-lorraine.fr Marco.Dozzi/Recherche Operationnelle3/prognonlin.pdf, (2004)

[13] Elias, M., Santosh, K., A new heuristic for the convex quadratic programming problem. American Journal of Operations Research, V5, pp. 373-383, (2015)

[14] Frank, M., Wolfe, B., An algorithm for quadratic programming. Naval Research Logistics Quarterly, V3, N(1-2), pp. 95-110, (1956)

[15] Gärtner, B., Schönherr, S., Smallest enclosing circles – an exact and generic implementation in C++ [online] ftp://ftp.inf.fu-berlin.de/pub/reports/tr-b-98-05.ps.gz (accessed 23 September 2016)

[16] Karmarkar, N.K., A new polynomial-time algorithm for linear programming. Proceedings of the 16^{th} Annual ACM Symposium on Theory of Computing, V4, pp. 373-395, (1984)

[17] Leulmi, A., Leulmi, S., Logarithmic barrier method via minorant function for linear programming. Journal of Siberian Federal University, Mathematics and Physics, V12, N2, pp. 191-201, (2019)

[18] Leulmi, A., Merikhi, B., Benterki, D., Study of a logarithmic barrier approach for linear semidefinite programming. Journal of Siberian Federal University, Mathematics and Physics, V11, N3, pp. 300-312, (2018)

[19] Menniche, L., Benterki, D., A logarithmic barrier approach for linear programming. Journal of Computational and Applied Mathematics, Elsevier, V312, pp. 267-275, (2017)

[20] Roos, C., Terlaky, T., Vial J.P., Theory and algorithms for linear optimization: an interior point approach. Chichester: Wiley, (1997)

[21] Roumili, H., Boudjellal, N., An infeasble interior point method for convex quadratic problems. Journal of Numerical Analysis and Approximation Theory, V47, N2, pp. 177-186, (2018)

[22] Wright, SJ., Primal-dual interior point methods. Society for Industrial and Applied Mathematics, (1997).

[23] Ye, Y., Tse, E. An extension of Karmarkar's projective algorithm for convex quadratic programming. Mathematical Programming, V44, pp. 157-179, (1989)

[24] Wolfe, P., The simplex method for quadratic programming. Econometrica: Journal of the Econometric Society, V27, N3, pp. 382-398, (1959)

[25] Yunong, Z., Jun, W., A dual neural network for convex quadratic programming subject to linear equality and inequality constraints. Physics Letters a, V298, pp. 271-278, (2002)