Gulf Journal of Mathematics
Vol 21, Issue 2 (2025) 275-287

ISSN: 2309-4966
https://doi.org/10.56947/gjom.v21i2.3623

ANALYSIS AND NUMERICAL ESTIMATION OF THE STEP SIZE IN OPTIMIZATION PROBLEMS

FARID LEULMI[1], SOUMYA LEULMI[2*] and AMAR MOUMEN BEKKOUCHE[3]

Abstract. Several studies have focused on determining the optimal step size for a wide range of optimization problems. In the present study, we propose a novel minorant function within the barrier approach framework to address convex quadratic problems with inequality constraints. The paper presents a comparative numerical analysis between traditional line search methods and the proposed minorant function for step-size determination in the logarithmic barrier method. The findings of our numerical experiments indicate that minorant function outperforms line search methods, making it a more effective and reliable approach.

Keywords. Optimization method, minorant function, convex quadratic programming, optimization problems
2020 Mathematics Subject Classification. Primary 90C25; Secondary 90C20

1. Introduction and preliminaries

Optimization methods play a crucial role in addressing a wide range of real-world problems, where the goal is to minimize or maximize an objective function subject to specific constraints. Among the many classes of optimization problems, convex quadratic programming (CQP) holds a special place due to its wide applicability in engineering, economics, operations research, and machine learning. In CQP, the function to be minimized (or maximized) is quadratic, and the constraints are typically linear, making the problem both tractable and relevant for practical applications.

Many studies have examined techniques aimed at improving constrained optimization procedures, including those applied to linear optimization (LO). For example, Manickam [14] explored related optimization frameworks.

A key challenge in solving CQP problems efficiently lies in the determination of step size during iterative algorithms. The choice of step size directly affects the convergence rate and computational efficiency of the solution process. Traditionally, line search techniques have been widely employed to compute appropriate

Date: Received: Sep 29, 2025; Revised: Oct 27, 2025 Accepted: Nov 10, 2025.
* Corresponding author
© The Author(s) 2025. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of the licence, visit https://creativecommons.org/licenses/by-nc-nd/4.0/.

step sizes. However, these methods can be computationally expensive and may not always guarantee the most efficient progress toward the solution.

To address these limitations, recent studies have explored the use of minorant functions as an alternative for step-size selection. A minorant function provides a lower approximation of the barrier function or merit function used in the algorithm, allowing for a more systematic and often more efficient computation of the step size. By incorporating a minorant function into an optimization method, it is possible to accelerate convergence while maintaining numerical stability.

This paper investigates the use of minorant functions within the framework of barrier methods for convex quadratic programming. The goal is to demonstrate that this approach not only improves the computational performance compared to classical line search methods but also enhances the robustness of the algorithm when solving large-scale or ill-conditioned optimization problems.

Therefore, we will explore the solution to the following quadratic programming problem :

$$\begin{cases} \min\limits_{x\in D} \quad f(x) \quad = \frac{1}{2}x^tQx + c^tx, \end{cases} \tag{1.1}$$

where $D = \{x \in \mathbb{R}^n : Ax \geq b\}$.

The following assumptions hold:

1. We know a point $x_0 \in \mathbb{R}^n$ such that $Ax_0 > b$.
2. $c \in \mathbb{R}^n$, Q is a $\mathbb{R}^{n\times n}$ symmetric semidefinite matrix.
3. $b \in \mathbb{R}^p$, A is a $(p \times n)$ full rank matrix.
4. The set of optimal solutions of (1.1) is nonempty and bounded.

We have now replaced the problem (1.1) with the following newly perturbed problem (1.2) as an approximation.

$$\begin{cases} \min \quad f_\mu(x) \\ x \in \mathbb{R}^n, \end{cases} \tag{1.2}$$

The barrier function $f_\mu : R^n \to (-\infty, +\infty]$ is given as follows:

$$f_\mu(x) = \begin{cases} f(x) - \mu \sum_{i=1}^m \ln < e_i, Ax - b > & \text{if } Ax - b > 0 \\ +\infty & \text{otherwise,} \end{cases}$$

where $< x, y >$ denoted the scalar product of $x, y \in \mathbb{R}^n$ and given by $< x, y >= x^ty = \sum_{i=1}^n x_iy_i$, $(e_1, e_2, ..., e_m)$ is the canonical basis in $\mathbb{R}^m$ and μ is a strictly positive barrier parameter.

Generally, a classical Newton descent method is used to solve this problem. In our new approach, instead of minimizing f_μ, along the descent direction at a current point x, we propose an approximate function G for which the optimal the solution of the step size α is obtained explicitly.

Let's minimize the function G such that :

$$G(\alpha) = \frac{1}{\mu}(f_\mu(x_\mu + \alpha d) - f_\mu(x_\mu)) \geq \breve{G}(\alpha), \ \forall \alpha > 0,$$

with $G(0) = \breve{G}(0) = 0$, $G'(0) = \breve{G}'(0) < 0$.

The best quality of the approximations $\breve{G}$ of G is ensured by the condition $G''(0) = \breve{G}''(0)$.

The idea of this new approach consists in introducing one original process to calculate the displacement step α based on minorant functions.

Recently, many topics concerning the solving of quadratic programming problems have been developed by interior point methods. These methods separated into three main categories: affine methods, potential reduction methods, and central trajectory methods.

Independent of prior works, various logarithmic barrier interior-point methods have been developed that utilize approximate functions, specifically majorant and minorant functions, to guide the optimization process. These methods provide a systematic way to construct search directions and step sizes while ensuring convergence within the feasible region. In particular, Crouzeix et al. [4] introduced such techniques for semidefinite programming (SDP) problems, demonstrating how approximate functions can be used to effectively handle the constraints and maintain feasibility. Extending this approach, A. Leulmi et al. [10] proposed a novel barrier method based on a new minorant function for SDP, which allowed for more efficient computation of step sizes and improved numerical performance. This idea was further generalized in [9], where the concept of the minorant function was adapted to nonlinear programming (NLP) problems, providing a systematic framework for constructing minorant functions in more general optimization settings.

Our proposed minorant function is novel and, to the best of our knowledge, has not yet been explored in the quadratic optimization literature. One significant advantage of such approximate functions is that they enable the computation of step sizes efficiently and with minimal computational cost, in contrast to classical line search methods, which are often time-consuming and expensive. Building on these prior efforts, our objective is to develop an approach for efficiently solving quadratic optimization problems using this new minorant function.

The aim of this paper is to present a comparative numerical study between line search methods and our new minorant function to compute the step-size along the direction in barrier logarithmic methods.

So, in this work, we address this problem. More precisely, we first introduce in section 2 of our paper, the existence and uniqueness of the minimum point of the new objective function given by Leulmi et al. [13] , as well as the convergence of the perturbed problem into the initial one.

The section 3 describes the algorithm and the minorant function employed to compute the step size by Leulmi et al. in [13] for solving the initial problem. To lend further support to our theoretical results, the numerical experiments are carried out to illustrate the good accuracy of the studied approach over the line search method in section 4. Finally Section 5 should summarize the main conclusions and perspectives.

Firstly, we need to study the convergence of (1.2) to (1.1).

Let us define ψ the function defined on $\mathbb{R} \times \mathbb{R}^n$ by:

$$\psi(\mu, x) = \begin{cases} f(x) + \sum_{i=1}^{n} \xi(\mu, x_i) & \text{if} \quad x \geq 0,\ Ax \geq b, \\ +\infty & \text{if not,} \end{cases}$$

where $\xi : \mathbb{R}^2 \longrightarrow (-\infty, +\infty]$ is a convex, lower semicontinuous and proper function given by:

$$\xi(\mu, \alpha) = \begin{cases} \mu \ln(\mu) - \mu \ln(\alpha) & \text{if} \quad \alpha > 0 \text{ and } \mu > 0, \\ 0 & \text{if} \quad \alpha \geq 0 \text{ and } \mu = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

So, the function ψ is convex, lower semicontinuous and proper function.

Furthermore, the function g defined by:

$$g(\mu) = \inf_x [\psi_\mu(x) := \psi(\mu, x) : x \in \mathbb{R}^n] \tag{Pr}$$

is convex. Notice that the problems (1.2) and (1.1) coincide in $\mu = 0$ then $\alpha = g(0)$.

For a fixed μ, the function ψ_μ is convex, lower semicontinuous and proper. As the asymptotic function of ψ_μ is defined by:

$$(\psi_\mu)_\infty (d) = \lim_{\alpha \to +\infty} \frac{\psi_\mu(x_0 + \alpha d) - \psi_\mu(x_0)}{\alpha},$$

Therefore, the asymptotic functions of ψ_μ and f are related by:

$$(\psi_\mu)_\infty (d) = \begin{cases} (f)_\infty (x) & \text{if} \quad d \geq 0, \ Ad = 0, \\ +\infty & \text{if not.} \end{cases}$$

In addition, assumption (4) can be rewritten as:

$$\{d \in \mathbb{R}^n : (f)_\infty (x) \leq 0, \ d \geq 0, \ Ad = 0\} = \{0\}.$$

Afterward,

$$\left\{d \in \mathbb{R}^n : (\psi_\mu)_\infty (d) \leq 0\right\} = \{0\}$$

and from [13], the strictly convex problem (1.2) admits a unique optimal solution x_μ^* for each μ. The solution of the problem (1.1) reduces to the solution of the series of problems (1.2). The sequence of the solutions x_μ of (1.2) should converge to the solution of (1.1) when μ tends to 0.

We are now ready to present the convergence result of (1.2) to (1.1) which is proved in Lemma 1 on [3].

Let $\mu > 0$, for all $x \in D$, we define $\psi(x, \mu) = f_\mu(x)$.

Lemma 1.1. [3] *Let $\mu > 0$ and suppose that x_μ is an optimal solution to problem (1.2), If the sequence $\{x_\mu\}$ satisfies*

$$\lim_{\mu \to 0} x_\mu = x^*,$$

then x^* is an optimal solution to problem (1.1).

In the next section, the perturbed problems are solved using the minorant function.

Solving Perturbed Problem Via Minorant Function

In this section, we aim to find the solution to the perturbed problem $x_{k+1} = x_k + \alpha_k d_k$. To achieve this, we first apply the Newton method to compute the descent direction d_k. Next, we determine the displacement step α_k using our new minorant function. Finally, we outline a standard prototype algorithm.

1.1. **Newton descent direction.** We apply a logarithmic barrier interior point method, which is based on the necessary and sufficient optimality conditions and involves generating a sequence of iterations

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

where d_k denotes the descent direction obtained as the solution of the following linear system:

$$\nabla^2 f_\mu(x_\mu) d_k = -\nabla f_\mu(x_\mu). \tag{1.3}$$

Unfortunately, this strategy does not guarantee that each iterate x_{k+1} generated by the algorithm is reached. Thus, we cannot guarantee that $A(x_k + d_k) > b$. In order to prevent this difficulty, we present a step size α_k at each iteration, which can ensure the feasibility of $x_{k+1} = x_k + \alpha_k d_k$.

1.2. **Step Size Calculation Procedure.** Two primary procedures are used for computing the step size α_k :

(1) Line search methods : Such as the method of
Goldstein-Armijo, Fibonacci, Wolfe,... etc. They are based on the unidimensional function's minimization:

$$\phi(\alpha) = \min_{\alpha>0} f_\mu(x_\mu + \alpha d).$$

They are time-consuming and unfortunately very sensitive.

(2) Minorant function : Leulmi et al. [10] were the first to propose this technique for positive semidefinite programming problems and it relies on approximating the function

$$G(\alpha) = \frac{1}{\mu}(f_\mu(x_\mu + \alpha d) - f_\mu(x_\mu)),$$

with another function for which the minimum can be computed straightforwardly. This function allows one to compute the displacement step at each iteration in a relatively short execution time and fewer instructions than the line search approach.

By using the Lemma 2 in [3], the function G can be written, for all $\alpha \in [0, \widehat{\alpha}[$, such that $\widehat{\alpha} = \min_{i \in I_-}\{\frac{-1}{y_i}\}$ and $I_- = \{i : y_i < 0\}$, as follows:

$$G(\alpha) = \frac{1}{\mu}\left(\frac{1}{2}\alpha^2 d^t Q d - \alpha d^t Q d\right) + \alpha\left(\sum_{i=1}^{m} y_i - y^2\right) - \sum_{i=1}^{m} \ln(1 + \alpha y_i) \tag{1.4}$$

$$= \frac{1}{\mu}\left(\frac{1}{2}\alpha^2 d^t Q d - \alpha d^t Q d\right) + \vartheta\alpha - \sum_{i=1}^{m} \ln(1 + \alpha y_i), \tag{1.5}$$

where $\vartheta = \sum_{i=1}^{m} y_i - y^2$, $y_i = \frac{<e_i, Ad>}{<e_i, Ax-b>}$, $i \in \{1, ..., m\}$ and $\|y\|$ denoted the Euclidean norm of $y \in \mathbb{R}^n$ and given by $\|y\| = \sqrt{\langle y, y\rangle}$.

- Prototype algorithm :
Here, we present the prototype algorithm designed to find an optimal solution $\bar{x}$ of the problem (1.1).

Begin algorithm

Initialization

Start with x_0 a strictly feasible solution of (1.1), $\mu > 0$, ε a given precision and $k = 0$.

While $\|\nabla f_\mu(x_k)\| > \varepsilon$ do

– Determine the solution to the system $\nabla^2 f_\mu(x_k)d_k = -\nabla f_\mu(x_k)$.

– Calculate the displacement step α_k.

– Take $x_{k+1} = x_k + \alpha_k d_k$ and $k = k + 1$.

– Put $\mu = \sigma\mu,\ 0 < \sigma < 1$.

End While

We have obtained a good approximate solution of the problem (P).

End algorithm.

In the following, we provide some useful inequalities that will be used throughout the paper.

Wolkowicz [17] introduced the following inequalities:

$$\begin{aligned} \bar{z} - \sigma_z\sqrt{n-1} &\leq \min_i z_i \leq \bar{z} - \frac{\sigma_z}{\sqrt{n-1}}, \\ \bar{z} + \frac{\sigma_z}{\sqrt{n-1}} &\leq \max_i z_i \leq \bar{z} + \sigma_z\sqrt{n-1}. \end{aligned}$$

We note that Merikhi et al. [4] established certain useful inequalities involving the maximum and minimum of $z_i > 0$ for any $i = 1,\ ..,\ n$

$$n\ln(\bar{z} - \sigma_z\sqrt{n-1}) \leq A \leq \sum_{i=1}^{n} \ln(z_i) \leq B \leq n\ln(\bar{z}), \tag{11}$$

where

$$A = (n-1)\ln(\bar{z} + \frac{\sigma_z}{\sqrt{n-1}}) + \ln(\bar{z} - \sigma_z\sqrt{n-1})$$

and

$$B = \ln(\bar{z} + \sigma_z\sqrt{n-1}) + (n-1)\ln(\bar{z} - \frac{\sigma_z}{\sqrt{n-1}}).$$

Here, $\bar{z}$ and σ_z represent the mean and standard deviation, respectively, of the series $\{z_1,\ z_2,\ ...,\ z_n\}$ of n real numbers, defined as follows:

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i \quad \text{and} \quad \sigma_z^2 = \frac{1}{n}\sum_{i=1}^{n} z_i^2 - \bar{z}^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2,$$

where $z_i = 1 + \alpha y_i$, $\bar{z} = 1 + \alpha\bar{y}$ and $\sigma_z = \alpha\sigma_y$.

Now, we give our main result of the paper.

1.3. **New minorant function.** In 2024, Leulmi et al: [13] proposed three minorant functions. Among them, we select their best minorant function as a basis for our approach:

$$\widetilde{G}(\alpha) = \left(n\bar{y} - \|y\|^2\right)\alpha - \frac{1}{\mu}\widehat{\alpha}d^tQd - p\ln\left(1 + \phi\alpha\right), \quad \forall\alpha > 0,\ 0 < p < 1.$$

With $\phi = \frac{\|y\|^2}{(n-1)\gamma+\beta}$, $\gamma = \bar{y} - \frac{\sigma_y}{\sqrt{n-1}}$ and $\beta = \bar{y} + \sigma_y\sqrt{n-1}$.

In this work, we introduce a new variant, denoted $G_{\min}$, which is constructed by modifying the parameters and structure of the original function. This modification leads to stronger convexity properties and allows for more efficient computation of the step size. This new function constitutes the main contribution of our paper:

We now consider our new function $G_{\min}$ defined by:

$$G_{\min}(\alpha) = \frac{\|y\|^2}{\delta}\alpha - \frac{1}{\mu}\widehat{\alpha}d^tQd - \tau \ln\left(1 + \frac{\|y\|^2}{\delta}\alpha\right), \quad \forall \alpha > 0,\ 0 < \tau < 1.$$

where $\delta = \bar{y} + \frac{\sigma_y}{\sqrt{n-1}}$.

Lemma 1.2. *$G_{\min}$ is strictly convex for all $\alpha \geq 0$, and we have:*

$$G_{\min}(\alpha) \leq G(\alpha).$$

and verifies the derivative conditions at $\alpha = 0$:

$$G_{\min}(0) = 0 \text{ and } G''_{\min}(0) = -G'_{\min}(0) = \|y\|^2.$$

In order to strengthen the theoretical foundation of our approach and address the convergence aspects, we provide here the key result establishing the convexity of

$G_{\min}(\alpha)$ and the inequality $G_{\min}(\alpha) \leq G(\alpha)$, which plays a central role in proving the global convergence of the algorithm and improving the step-size computation.

Proof. Let $P(\alpha) = G(\alpha) - G_{\min}(\alpha)$, then

$$P(\alpha) = \alpha\left(\vartheta - \frac{\|y\|^2}{\delta}\right) - \sum_{i=1}^{m} \ln(1 + \alpha y_i) + \tau \ln\left(1 + \frac{\|y\|^2}{\delta}\alpha\right).$$

In order to establish $G_{\min}(\alpha) \leq G(\alpha)$ it suffices to demonstrate that $P(\alpha) \geq 0$.

Clearly, we get $P(0) = 0$ and

$$P'(\alpha) = \vartheta - \frac{\|y\|^2}{\delta} - \sum_{i=1}^{m} \frac{y_i}{1 + \alpha y_i} + \tau \frac{\frac{\|y\|^2}{\delta}}{1 + \frac{\|y\|^2}{\delta}\alpha},$$

so, we obtained $P'(0) = 0$.

Furthermore, for all $\alpha \geq 0$, we have

$$P''(\alpha) = \sum_{i=1}^{m} \frac{y_i^2}{(1 + \alpha y_i)^2} - \tau \frac{\left(\frac{\|y\|^2}{\delta}\right)^2}{\left(1 + \alpha\frac{\|y\|^2}{\delta}\right)^2} \geq 0.$$

The fact that $|y_i| \leq \|y\|$ and $\beta \leq \|y\|$, it is easy to see the convexity of $P(\alpha)$, for $\forall \alpha \geq 0$, (since $P''(\alpha) \geq 0$).

Therefore :

$$\alpha\left(\sum_{i=1}^{m} y_i - y^2\right) - \sum_{i=1}^{m} \ln(1 + \alpha y_i) \geq \frac{\|y\|^2}{\delta}\alpha - \tau \ln\left(1 + \frac{\|y\|^2}{\delta}\alpha\right),$$

as

$$\frac{1}{2\eta}\alpha^2 d^t Q d > 0 \quad \text{and} \quad -\frac{1}{\eta}\widehat{\alpha} d^t Q d < -\frac{1}{\eta}\alpha d^t Q d, \forall \alpha \in [0, \widehat{\alpha}[.$$

We obtain

$$\underbrace{\frac{1}{2\eta}\alpha^2 d^t Q d + \alpha\left(\sum_{i=1}^{m} y_i - y^2\right) - \sum_{i=1}^{m}\ln(1+\alpha y_i) - \frac{1}{\eta}\alpha d^t Q d}_{G(\alpha)}$$

$$\geq \underbrace{\frac{\|y\|^2}{\delta}\alpha - \tau \ln\left(1 + \frac{\|y\|^2}{\delta}\alpha\right) - \frac{1}{\eta}\widehat{\alpha} d^t Q d}_{G_{\min}(\alpha)}.$$

The proof is thus finished. □

This inequality ensures that the step-size obtained by minimizing $G_{\min}$ leads to a monotonic decrease of the barrier function and, consequently, the objective function. This property is essential for proving global convergence of the algorithm.

Observe that the minimum of the convex function $G_{\min}$ occurs when $G'_{\min}(\alpha) = 0$ on $[0, \widehat{\alpha}[$, where $\widehat{\alpha} = \sup\{\alpha : 1+\alpha y_i > 0, \text{ for all } \ i = 1, ..., m\}$ and $y_i = \frac{<e_i, Ad>}{<e_i, Ax-b>}$, $i \in \{1, ..., m\}$.

By a simple calculus, one can derive the following explicit unique minimum point

$$\alpha^* = \frac{\beta}{\|y\|^2}(\tau - 1).$$

Moreover, this convexity result allows us to bound the number of iterations needed to reach a given accuracy using existing interior-point complexity results, which further supports the efficiency of the proposed method.

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

Lemma 1.3. [13] *From iteration k to $k+1$, the function f_μ experiences a significant decrease. Specifically, if x_k and x_{k+1} denote feasible solutions at iterations k and $k+1$, then:*

$$f_\mu(x_{k+1}) < f_\mu(x_k). \tag{1.6}$$

Combining the convexity of $G_{\min}$ with Lemma A11, we obtain a sufficient decrease in f_μ at each iteration, which establishes the descent property of the algorithm and supports its convergence.

2. Experimental Results

Here, we provide a series of comparative numerical analysis tests to confirm and consolidate the numerical performances of the best minorant function $G_{\min}(\alpha)$ given in [13], with respect to Wolfe's line search method. We have tested examples of different sizes. The experiments were implemented in MATLAB R2023a On a

Ryzen 7 5800H (4.5 GHz) with 16 GB RAM and an RTX 3070 graphics card with 8 GB VRAM, comparing our proposed function with Wolfe's line search method. For this purpose, we consider the following quadratic problem:

$$\alpha = \min[f(x) : x \geq 0,\ Ax \geq b],$$

where $f(x) = \frac{1}{2}x^tQx + c^tx$.

In the tables below, we have taken $\varepsilon = 1.0e-005$ and we denoted by:

- (iter) the number of iterations needed to achieve convergence to the optimal solution.
- (time) the time of computation in seconds (s).
- (MF) the strategy of new minorant functions introduced in this paper.
- (LS) the classical Wolfe's line search..

2.1. **Examples.** To illustrate the effectiveness of the proposed approach, we present several numerical examples. These examples are designed to highlight the performance of $G_{\min}(\alpha)$ compared to Wolfe's line search method under different problem sizes and conditions.

Example 1: The matrix Q is defined as follows:

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

$$A[i,j] = \begin{cases} 1 & \text{if } i = j \text{ or } j = i+m,\ i = 1,..,m \text{ and } j = 1,..,n \\ 0 & \text{otherwise.} \end{cases}$$

$$c[j] = j \text{ and } b[i] = \frac{i+1}{2},\ \forall i = 1,..,m.$$

with $n = 2m$. We test this example for different value of n. The table below summarizes the results achieved in this study.

ex(m,n)	MF		LS	
	iter	time	iter	time
200×400	14	5.49965	38	22.52401
300×600	20	39.00861	45	88.11235
600×1200	25	45.76899	66	148.62103
1000×2000	37	89.09854	70	2004.11257
1500×3000	67	700.05445	101	2453.92312

Example 2: The matrix Q is defined as follows:

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

$$A[i,j] = \begin{cases} 1 & \text{if } i = j \text{ or } j = m+i,\ i = 1,..,m \text{ and } j = 1,..,n \\ 0 & \text{otherwise.} \end{cases}$$

$$c[i] = -1,\ c[m+i] = 0 \text{ and } b[i] = 2,\ \forall i = 1,..,m.$$

with $n = 2m$. We test this example for different value of n.

The table below summarizes the results achieved in this study.

ex(m,n)	MF		LS	
	iter	time	iter	time
200×400	5	3.66689	26	22.52401
300×600	7	42.94765	35	97.10345
600×1200	16	59.56780	48	224.32120
1000×2000	23	89.52229	51	497.01165
1500×3000	31	199.23147	78	1321.03278

Example 3: The matrix Q is defined as follows:

$$\begin{aligned} Q[i,j] &= \left\{ \frac{1}{j+i} \quad \text{for} \quad i,j = 1,..,n. \right. \\ A[i,j] &= \begin{cases} 1 & \text{if } i = j \text{ or } j = m+i,\ i = 1,..,m \text{ and } j = 1,..,n \\ 0 & \text{otherwise.} \end{cases} \\ c[j] &= 2j \text{ and } b[i] = i^2,\ \forall i = 1,..,m. \end{aligned}$$

with $n = 2m$. We test this example for different value of n.

The table below summarizes the results achieved in this study.

ex(m,n)	MF		LS	
	iter	time	iter	time
200×400	5	3.23451	16	29.12331
300×600	7	10.98761	27	84.15001
600×1200	22	30.92312	55	153.92210
1000×2000	26	127.00345	68	2213.11431
1500×3000	43	245.12003	124	3121.11303

To make a more informed decision, we choose two other examples.

Example 4: The matrix Q is defined as follows:

$$\begin{aligned} Q[i,j] &= \begin{cases} 1 & \text{if } i = j = 1 \text{ or } i = j = n \\ 1 & \text{if } i = 2,...,n \\ 4 & \text{if } i = 2,...,n-1, \end{cases} \\ A[i,j] &= \begin{cases} 0 & \text{if } i \neq j \text{ or } i+1 \neq j \\ 1 & \text{if } i = j \text{ or } i = j+m, \end{cases} \\ b[i] &= 4 \text{ and } \ c[j] = \frac{j+1}{2},\ \forall i = 1,..,n,\ j = 1,..,m. \end{aligned}$$

with $n = 2m$. We test this example for different value of n.

The table below summarizes the results achieved in this study.

ex(m,n)	MF		LS	
	iter	time	iter	time
200×400	3	0.89022	18	12.19942
300×600	5	1.00112	21	112.59831
600×1200	15	30.23466	53	405.03456
1000×2000	25	79.41231	67	2003.67781
1500×3000	37	177.09123	132	3234.71205

Example 5: The matrix Q is defined as follows:

$$\begin{aligned} Q &= blockdiag(B, B, \ldots, B) \text{ where } B = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix} \\ A[i,j] &= \begin{cases} 1 & \text{if } j = i \text{ or } j = i+m \text{ or } j = i+2m, \\ 0 & \text{otherwise.} \end{cases} \\ b[i] &= \frac{i+1}{3},\ \forall i = 1, .., m, \\ \text{and } c[j] &= j,\ \forall j = 1, .., n. \end{aligned}$$

with $n = 2m$. We test this example for different value of n.

The table below summarizes the results achieved in this study.

ex(m,n)	MF		LS	
	iter	time	iter	time
200×400	14	4.23091	38	22.52998
300×600	20	29.67853.	45	88.11675
600×1200	25	45.775634	66	148.62486
1000×2000	37	78.22341	70	2004.11239
1500×3000	67	600.23459	101	2453.92051

Key observations:

(1) The $G_{\min}$ method (minorant function) significantly reduces both the number of iterations and the CPU time.
(2) The time savings become especially substantial for larger problem sizes (for example, for $\mathbf{1500 \times 3000}$,there is more thana $\mathbf{70}\%$ reduction in computation time).
(3) Both methods converge to the same optimal value.

2.2. **Remark.** From the three tables, we can clearly observe that both the number of iterations and the computing time are significantly reduced when using the proposed approximate approach compared to Wolfe's line search method. This remarkable improvement demonstrates the efficiency and robustness of our new minorant function $G_{\min}(\alpha)$. In particular, the reduction in computational effort highlights the ability of our function to achieve faster convergence while maintaining accuracy. These results confirm that our approach not only performs better than the classical line search method but also provides a practical and reliable alternative for solving perturbed optimization problems.

3. Conclusion

Based on the results of our numerical study, we can conclude that using a minorant function as part of the optimization strategy is more efficient both in terms of computation time and the number of iterations required when compared to the traditional line search method. These encouraging results indicate that the minorant function approach holds significant potential for improving the performance of interior-point algorithms.

Looking ahead, this motivates us to develop even more effective approximation functions to further enhance algorithmic efficiency. We also aim to extend our investigation to a broader class of optimization problems, including those that go beyond quadratic programming.

Furthermore, there is no reason not to consider applying this approach in other areas of mathematics as well. For instance, the idea could be explored in the context of nonlinear dynamical systems, where barrier-based methods might help analyze system stability or identify periodic orbits while ensuring state constraints are satisfied. It could also be applied to probability and statistics, for example, in parameter estimation problems under constraints, or in shaping probability distributions to meet desired properties such as non-negativity and normalization.

Beyond these, potential applications include machine learning, where such techniques could improve constrained model training, network optimization, ensuring robust and efficient resource allocation, and applied mechanics, where positive semi-definiteness of stress matrices is required for stability analysis. These directions open promising opportunities for developing more general and powerful tools for solving complex, real-world mathematical problems.

References

1. Bazraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). Nonlinear Programming,Willey-Interscience, Copyright by John Wiley & Sons, Inc. All rights reserved. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada. https://doi.org/10.1002/0471787779. https://onlinelibrary.wiley.com/doi/book/10.1002/0471787779?utm_source=chatgpt.com
2. Bonnans, J. F., Gilbert, J. C., Lemaréchal, C. & Sagastiz àbal, C. (2003). Numerical optimization: theoretical and practical aspects. Mathematics and Applications, Vol. 27. Springer-Verlag, Berlin. DOI : 10.1002/0471787779.
3. Chaghoub, S., & Benterki, D. (2021). A Logarithmic Barrier Method Based On a New Majorant Function for Convex Quadratic Programming. IAENG International Journal of Applied Mathematics 51 pp 1-6.
4. Crouzeix, J. P., & Merikhi, B. (2008). A logarithm barrier method for semi-definite programming. RAIRO-Operations Research, 42(2), 123-139.https://doi.org/10.1051/ro:2008005
5. Crouzeix, J. P., & Seeger, A. (1996). New bounds for the extreme values of a finite sample of real numbers. Journal of mathematical analysis and applications, 197(2), 411-426.https://doi.org/10.1006/jmaa.1996.0029
6. Dib, Z., & Leulmi, A. (2025). A new numerical scheme for solving linear optimization problems. Gulf Journal of Mathematics, Vol 20, Issue 1 pp 303-312.https://doi.org/10.56947/gjom.v20i.2916
7. Kacem, N., & Leulmi, A. (2025). An explicit step length for solving an optimization problem. Gulf Journal of Mathematics, Vol 19, Issue 2 pp 398-411.https://doi.org/10.56947/gjom.v19i2.2801
8. A. Leulmi, A. Etude d'une méthode barrière logarithmique via minorants functions pour la programmation semi-dé finie. Thèse de doctorat, Université de Biskra (2018).
9. Leulmi, A. (2024). An Efficient Penalty Method without a Line Search for Nonlinear Optimization. Axioms, 13(3), 176. https://doi.org/10.3390/axioms13030176
10. A. Leulmi, B. Merikhi, D. Benterki, Study of a Logarithmic Barrier Approach for Linear Semidefinite Programming, Journal of Siberian Federal University. Mathematics & Physics 2018, 11(3), 1–13.

11. Leulmi, A., & Leulmi, S. (2019). Logarithmic barrier method via minorant function for linear programming. Journal of Siberian Federal University Mathematics and physics, 12(2), 191-201. https://doi.org/10.17516/1997-1397-2019-12-2-191-201
12. Leulmi, S., & Leulmi, A. (2023). A novel adaptive method based on new minorant-majorant functions without line search for semidefinite optimization. Nonlinear Dynamics and Systems Theory, 23(3), 310-322.
13. Leulmi, A., Ziadi, R., Souli, C., Saleh, M. A., & Almaymuni, A. Z. (2024). An Interior Point Algorithm for Quadratic Programming Based on a New Step-Length. International Journal of Analysis and Applications, 22, 233-233.https://doi.org/10.28924/2291-8639-22-2024-233
14. Manickam, K. (2015). Optimization techniques for mixed finite element methods of asemilinear elliptic PDEs superconvergence. Gulf Journal of Mathematics, 3(1),55-66. https://doi.org/10.56947/gjom.v3i1.17013
15. Nesterov, Y., & Nemirovsky, A. (1994). Interior-point polynomial methods in convex programming , Studies in Applied Mathematics, Vol.13, SIAM, Philadelphia.https://doi.org/10.1137/1.9781611970791
16. Papadimitriou, C.H., & Steiglitz, K. (1982). Combinatorial Optimization: Algorithms and Complexity (Prentice-Hall, Englewood Cliffs).
17. Wolkowicz, H., & Styan, G. P. H. (1980). Bounds for eigenvalues using traces. Linear Algebra and Appl.29, 471-506. https://doi.org/10.1016/0024-3795(80)90077-4
18. Ye, Y., & Tse, E. (1989). An Extension of Karmarkar's Projective Algorithm for Convex Quadratic Programming, Math. Programming, vol. 44, pp. 157–179.https://doi.org/10.1007/BF01587086
19. Ye, Y. (1987). Interior algorithms for linear, quadratic, and linearly constrained convex programming," Ph.D. Thesis, Department of Engineering Economic Systems, Stanford University (Stanford, CA). https://scholar.google.com/scholar_lookup?doi=10.5555%2F914114 . https://dl.acm.org/doi/10.5555/914114
20. Ye, Y., & Tse, E. (1986). A Polynomial Algorithm for Convex Quadratic Programming, Stanford Univ. https://pure.uvt.nl/ws/portalfiles/portal/960493/POLYNOMI.PDF?utm_source=chatgpt.com

[1] Brother Mentouri University of Constantine-1, Algeria, Laboratory of: Mathematical Modeling and Simulation
Email address: farid.leulmi@umc.edu.dz,farid_leulmi@yahoo.fr

[2]Department of Mathematics, 20 August 1955 University of Skikda, Algeria.
Email address: s.leulmi@univ-skikda.dz,leulsou@yahoo.fr

[3]Department of Mathematics, 20 August 1955 University of Skikda, Algeria.
Email address: a.moumenbekkouche@univ-skikda.dz