Filomat 38:3 (2024), 1069–1083
https://doi.org/10.2298/FIL2403069A

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Initialization of the difference of convex functions optimization algorithm for nonconvex quadratic problems

Saadi Achour[a], Abdelaziz Rahmoune[a], Djamel Ouchenane[a], Asma Alharbi[b], Salah Boulaaras[b,*]

[a]*Laboratory of Pure and Applied Mathematics, University Amar Telidji of Laghouat, BP 37G, Ghardaia Road, Laghouat 03000, Algeria*
[b]*Department of Mathematics, College of Sciences and Arts, ArRas, Qassim University, Buraydah, Saudi Arabia*

Abstract. The Difference of Convex functions Algorithm (DCA) is used to solve nonconvex optimization problems over a certain convex set, specifically quadratic programming ones, generally by finding approximate solutions. DCA efficiency depends on two basic parameters that directly affect the speed of its convergence towards the optimal solution. The first parameter is the selected decomposition and the second is the assigned initial point. The aim of this study was to create a new algorithm that allows overcoming the need for a pre-selected initial estimate of the DCA. To achieve this aim, we performed an experimental study with 107 test problems using an implementation framework with MATLAB. Assessment was based on key performance indicators: (*a*) the time required to reach the initial point and solution and (*b*) the number of iterations. We selected three initial points, the first (x_0^{lin}) is the minimum of the linear part of the nonconvex quadratic problem (NCQP), the second (x_0^{cvx}) is the approximate global minimum of the convex part, and the third (x_0^{cve}) is the approximate global minimum of the concave part. We compared between the minimuma computed by DCA for each of the three initial estimates. The results demonstrated clear advantage of the DCA algorithm with (x_0^{lin}). Based on this outcome, we constructed a novel algorithm called Initialized DCA (IDCA) that allows implementation of the DCA with the best initial estimate without the requirement for a pre-determined initial point.

1. Introduction

Nonconvex quadratic programming (NCQP) is a problem that focuses on minimizing a nonconvex quadratic function over a certain convex set. This problem covers a very important area in applied mathematics, which is a part of nonlinear programming approach. The term ”convex” is not limited to optimization, but also can extends beyond that to other disciplines of functional analysis, as shown in [2, 3, 5]. Many practical problems can be formulated as quadratic programming (QP) problems, or at least need quadratic programming methods to resolve them. We can cite logistic-transport [24], telecommunication [26], network security [7], bioinformatics [13], finance [11], management[23], mechanics [29], image processing [8], petrochemistry [39], optimal control [40], inverse problems [15, 21], and support vector machine [37] as examples of relevant applications of nonconvex programming.

2020 *Mathematics Subject Classification*. Primary 90C20, 90C26; Secondary 41A29, 46N10, 65D05.
Keywords. Nonconvex quadratic programming, local minimum, global minimum, DC Algorithm, initial estimate
Received: 10 May 2023; Revised: 30 August 2023; Accepted: 14 September 2023
Communicated by Maria Alessandra Ragusa
* Corresponding author: Salah Boulaaras
Email addresses: saadi.achour@lagh-univ.dz (Saadi Achour), a.rahmoune@lagh-univ.dz (Abdelaziz Rahmoune), d.ouchenane@lagh-univ.dz (Djamel Ouchenane), ao.alharbi@qu.edu.sa (Asma Alharbi), s.boularas@qu.edu.sa (Salah Boulaaras)

There exist several methods to solve nonconvex quadratic problems under particular conditions. For difference of convex functions algorithms (DCA), quadratic functions are the simplest smooth DC functions (functions that can be decomposed into two convex functions), whose derivatives are readily available and easy to manipulate.

Quadratic programming is useful because any twice differentiable function can be approximated by a quadratic function in the neighborhood of a given point. Moreover, quadratic problems are known to be NP-hard [27, 35], which makes them one of the most interesting and challenging classes of optimization problems.

Certain combinatorial optimization problems can also be studied as quadratic optimization problems [38]. For instance, a $0-1$ constraint of the form $x_i \in \{0,1\}, i = 1,..,p$ can be written as quadratic constraints $\left(\sum_{i=1}^{p} x_i (x_i - 1) \geq 0,\ 0 \leq x_i \leq 1\right)$ [27]. The general quadratic problem consists of a quadratic objective function and a set of linear inequality constraints, as shown below:

$$\begin{cases} \min Q(x) = \frac{1}{2}x^t Qx + c^t x \\ \qquad Ax \leq b,\ x \geq 0 \end{cases} \tag{1}$$

Where c is a n-vector, b is a m-vector, A is a $m \times n$ matrix and Q is a $n \times n$ matrix.

Without any loss of generality, it may be assumed that Q is symmetric. If this is not the case, then it can be converted to symmetric matrix by replacing Q with $\frac{Q + Q^t}{2}$, which does not change the value of the objective function $Q(x)$. Similarly, any problem where the variables are not necessarily nonnegative can be converted by a linear transformation to (1).

If the matrix Q is positive semidefinite or positive definite, then (1) becomes a convex optimization problem. Since any local optimum is equivalent to the global optimum in convex problems, (1) can be solved by any of the several algorithms for convex quadratic programming. In particular, it is well known that convex quadratic problems belong to class P (the class of problems solvable in polynomial time).

When the matrix Q has eigenvalues of mixed signs, (1) presents the toughest quadratic optimization problems. While many algorithms have been developed for the more particular cases of bilinear and concave quadratic problems, few approaches have been proposed for global optimization of problem (1) for the case of indefinite quadratic problems.

Most efforts to solve this difficult class of problems have focused on reducing the indefinite quadratic problem to either a bilinear or a concave minimization problem [22]. However, there are a few algorithms that directly solve this class. As a special case, if the problem (1) has box constraints:

$$\begin{cases} \min Q(x) = \frac{1}{2}x^t Qx + c^t x \\ \qquad x^L \leq x \leq x^U \end{cases} \tag{2}$$

Without loss of generality, we can consider $x^L = 0$ and $x^U = 1$.

This gives a direct relationship between box-constrained quadratic problems and one of the fundamental problems of combinatorial optimization, namely, minimizing a quadratic function of $0-1$ variables. In 1981, Hansen et al. [16] proposed the necessary conditions of optimality for problem (2).

A paper entitled "Globally solving nonconvex quadratic programming problems via completely positive programming" published by Jieqiu Chen and Samuel Burer in 2012 [9] introduced a global optimization algorithm for quadratic programming problems. This algorithm combines two ideas from the literature; finite branching based on the first-order KKT conditions and polyhedral semidefinite relaxations of completely positive (or copositive) programs. Through a series of computational experiments comparing the algorithm with existing codes on a diverse set of test instances, the authors demonstrated that the algorithm is an attractive choice for global resolution of nonconvex QP.

A very important approach is DCA, which plays the central role in the construction of local and global approaches that are based on convex analysis and optimization. This is because most nonconvex optimization problems are reformulated as DC programs, especially nonconvex quadratic programming problems.

DC functions have many important properties that were explored in the 1950s by Aleksandrov [4] and Hartman [17]. One of the main properties is their stability relative to operations frequently used in optimization. The DC algorithm was introduced in 1985 by P.D. Tao [32] for concave programming. It was then widely developed by P. D. Tao [33, 34], L. T. Hoai An [18, 19].

This article builds on the work in [1]; where we proposed an experimental study with the 30 test problems of Thaoi [36] to allow assessment of key performance indicators (convergence time and closeness to the global minimum), having selected two initial points for DCA in the quadratic case. Instead, in this work, we have examined three initial points for three different problem types along with Thaoi. This resuted on up to 107 problem tests, which is supposed to reinforce the experiment credibility. Along with that, we have considered the number of itterations as additional evaluation metric and we decomposed the timing evaluation depending on the different steps of the problems resolution. Finally, based on the obtained new results we have proposed an intialized variant of DCA algorithm.

This paper is organized as follows: In Section 2, we cover DCA theory. Section 3 we presents a possible DC decomposition approach for quadratic functions. Section 4 is devoted to the presentation of our proposed numerical method of comparison. The numerical results are presented and discussed in Section 5. Finally, we conclude the paper and offer some future directions in Section 6.

2. Difference of Convex Functions: Theory and Algorithm

DCA is an iterative method of local optimization based on local optimality and duality in DC programming. This algorithm was introduced by Pham Dinh Tao in 1985 [32] and then extensively developed by Pham Dinh Tao [32], Le Thi Hoai An et al. since 1994 [18, 19].

This approach is completely different from conventional sub-gradient methods in convex optimization. It makes it possible to construct two sequences x^k and y^k, which are candidates for optimal solutions of the primal and dual DC programs, respectively. The limits of the sequences x^k and y^k are generalized KKT points of these programs, while $\left\{(g-h)(x^k)\right\}$ and $\left\{(h^*-g^*)(y^k)\right\}$ are decreasing and tend to the same limit $\beta = (g-h)(x^*) = (h^*-g^*)(y^*)$.

We have the following properties:

Property 2.1. *The following elements are verified:*

1. *The sequences $\left\{(g(x^k)-h(x^k))\right\}$ and $\left\{(g^*(x^k)-h^*(x^k))\right\}$ decrease and tend to the same β limit that is greater than or equal to the global optimal value α;*
2. *If $(g-h)(x^{k+1}) = (g-h)(x^k)$ the algorithm stops at the iteration $k+1$, and the point x^k (resp. y^k) is a critical point of $g-h$ (resp. h^*-g^*);*
3. *If the optimal value of the problem (P_{dc}) is finite and if the sequences x^k and y^k are bounded, then any adherence value x^* of the sequence $\left\{x^k\right\}$ (resp. y^* of the sequence $\left\{y^k\right\}$) is a critical point of $g-h$ (resp. of h^*-g^*).*

The description of the DC algorithm is as follows:
Given $x^0 \in X$ chosen in advance, the sequences $\{x^k\}$ and $\{y^k\}$ are defined by:

$$y^k \in \partial h(x^k), x^{k+1} \in \partial g^*(y^k). \tag{3}$$

To construct the two sequences $\{x^k\}$ and $\{y^k\}$, we define two convex programs (D_k) and (P_k), for $k \geq 1$, as following :

$$(D_k) \qquad x^k \in \partial g^*(y^{k-1}) \longrightarrow y^k \in \arg\min\left\{h^*(y) - \left[g^*(y^{k-1}) + \left\langle x^k, y - y^{k-1}\right\rangle\right] : y \in Y\right\} = \partial h(x^k). \tag{4}$$

$$(P_k) \qquad y^k \in \partial h(y^k) \longrightarrow x^{k+1} \in \arg\min\left\{g(x) - \left[h(x^k) + \left\langle y^k, x - x^k \right\rangle\right] : y \in X\right\} = \partial h^*(y^k). \tag{5}$$

Then the point x^{k+1} (resp. y^k) is an optimal solution of the program (P_k) (resp. (D_k)). We can easily understand that (P_k) (resp. (D_k)) is a convex optimization problem obtained by replacing h (resp. g^*) of (P_{dc}) (resp. (D_{dc})) by its affine minor $h_k(x) = h(x^k) + \left\langle y^k, x^k \right\rangle$ at neighborhood of x^k with $y^k \in \partial h(x^k)$ (resp. $g_k^*(y) = g^*(y^{k-1}) + \left\langle x^k, y^{k-1} \right\rangle$ near y^{k-1} with $x^k \in \partial g^*(y^{k-1})$). Then we have the following simple diagram to describe the DC algorithm:

$$\begin{array}{ccc} x^k & \longrightarrow & y^k \in \partial h(x^k) \\ & \swarrow & \\ x^{k+1} \in \partial g^*(y^k) & \longrightarrow & y^{k+1} \in \partial h(x^{k+1}) \end{array} \tag{6}$$

The DC algorithm stops if at least one of the sequences $\left\{(g - h)(x^k)\right\}$, $\left\{(h^* - g^*)(x^k)\right\}$, $\left\{x^k\right\}$, $\left\{y^k\right\}$ converges. In practice, we often use the following stop conditions:

1. $(g - h)(x^{k+1}) - (g - h)(x^k) \le \varepsilon$;
2. $\|x^{k+1} - x^k\| \le \varepsilon$.

To get a $\varepsilon -$ *optimal* solution. Therefore, the DC algorithm can be described in Algorithm 1.

Algorithm 1 DC optimisation Algorithm

```
x^0 given
k ← 0
ε > 0                                          ▹ a defined precision
Step 1 : We calculate y^k ∈ ∂h(x^k)
Step 2 : We determine x^{k+1} ∈ ∂g*(y^k),      ▹ (in general,
    by solving a convex optimization problem)
if x* = x^{k+1} is the optimal solution of the problem then
    stop                                       ▹ the stopping condition is satisfied
else
    k + +
    goto Step 1 :
end if
```

2.1. Quadratic DCA

We consider the following nonconvex quadratic problem:

$$(QP) \begin{cases} \min f(x) = \dfrac{1}{2} x^t Q x + c^t x, \\ x \in S = \{x \in \mathbb{R}^n : Ax \le b, \ x \ge 0\} \end{cases} \tag{7}$$

In this work, we apply DCA approach to solve quadratic programs under linear constraints.

Suppose that we have the decomposition $f = g - h$, (or using one of the existing methods for decomposition, for example, the method prescribed in [6]). Where $g, h \in Conv(S)$ the set of all the convex functions defined in S.

Therefore, we can write the DCA for Quadratic problems as Algorithm 2.

Algorithm 2 DCA for Quadratic Functions

1: x^0 given
2: $k \leftarrow 0$
3: $\varepsilon > 0$ ▹ a defined precision
4: **Step 1 :** We calculate $y^k \in \partial h(x^k) = \nabla h(x^k)$
5: **Step 2 :** We determine $x^{k+1} \in \partial g^*(y^k)$ ▹ by solving the following convexe quadratic problem:

$$(P_c) \quad \left\{ Min \quad g(x,y) - \left\langle x, y^k \right\rangle : x \in \Omega \right\}$$

6: **if** $|\ (g-h)(x^{k+1}) - (g-h)(x^k)\ | \leq \varepsilon$ or $\|x^{k+1} - x^k\| \leq \varepsilon$ **then**
7: **stop** ▹ the stopping condition is satisfied
8: **else**
9: $k++$
10: **goto** *Step* 1 :
11: **end if**

3. Quadratic DC decomposition

The DC decomposition method is a computational approach that decomposes the objective function into two convex functions. There are several decomposition methods for quadratic functions, but we will focus on the conventional approch, which involves transforming the quadratic function to its canonical form.

3.1. Problem statement

A minimization problem of a quadratic function with linear constraints is presented in the following standard form:

$$(P_m) \begin{cases} \min f(x) = \frac{1}{2} x^t Q x + c^t x, \\ \quad Ax \leq 0, \ x \geq 0, \end{cases} \tag{8}$$

where Q is an indefinite symmetric matrix ($Q \neq 0$). c and x, are *vectors* of $\mathbb{R}^n$ and A is a matrix of dimension $m \times n$, with $rank(A) = m < n$ and $b \in \mathbb{R}^m_+$.

It is clear that without restricting the generalization we can limit the study to the following problem:

$$(P_m) \begin{cases} \min f(x) = \sum_{i=1}^n \alpha_i x_i + \beta_i x_i^2, \\ \quad Ax \leq 0, \ x \geq 0. \end{cases} \tag{9}$$

i.e. the matrix Q for the problem (P_m) will be diagonal. This writing is called the canonical form of a quadratic function, where $\alpha_i, \beta_i \in \mathbb{R}$.

We present a transformation method of (P_m) to its canonical form in the next subsection. For more details about the canonical transformation methods we refer the reader to [6, 10, 20, 25, 38].

3.2. Writing a quadratic form in its canonical form

Some objective functions have two properties: the coefficients of $x_i^2|_{i=1,2,\ldots,n}$ are identical; and there is no term involving $x_i x_j|_{i,j\in 1,2,\ldots,n}$. Therefore, the sets of levels for each objective function are spheres that can be traced by inspection. It is not immediately obvious how the sets of levels should be drawn. The purpose of this section is to show a method by which these sets can be drawn easily [20, 38]. The relevant mathematical tools are the eigenvectors and the eigenvalues. For more information on these concepts, the reader can refer to any of the many available linear algebra resources.

Consider a general quadratic function of n variables:

$$f(x) = \frac{1}{2}x^tQx + c^tx, \tag{10}$$

where x and c are vectors of $\mathbb{R}^n$ and Q is a square n order symmetric matrix.

Let S be the $n\times n$-matrix whose columns are the eigenvectors of Q and let D be the $n\times n$-diagonal matrix of the corresponding eigenvalues. The property that defines S and D is:

$$QS = SD. \tag{11}$$

An elementary property of eigenvectors is that they are orthogonal. The condition that they are also of standard unity is:

$$S^tS = I, \tag{12}$$

where I designates the $n \times n$-identity matrix. By multiplying the left part of (11) by S^t and using (12) we find:

$$S^tQS = D. \tag{13}$$

Let x_0 be the point which minimizes f. Being an unconstrained quadratic minimization problem, the optimality conditions imply that the gradient of f at x_0 must be zero. Let $g(x)$ be the gradient of f. By writing $f(x)$ explicitly in terms of the components of x, it is easy to see that:

$$g(x) = c + Qx. \tag{14}$$

Since $g(x_0) = 0,\ x_0$ is the solution of the linear equation

$$Qx_0 = -c. \tag{15}$$

We then introduce a change of variable y linked to x by

$$x = Sy + x_0. \tag{16}$$

Substituting this expression for x in (10):

$$\begin{aligned}
f(x) = &\quad f(Sy + x_0) && (17)\\
= &\quad \tfrac{1}{2}(Sy + x_0)^tQ(Sy + x_0) + c^t(Sy + x_0) && (18)\\
= &\quad \tfrac{1}{2}y^TS^tQSy + \tfrac{1}{2}x_0QSy + \tfrac{1}{2}y^tS^tQx_0 + \tfrac{1}{2}x_0Qx_0 + c^tSy + c^tx_0 && (19)\\
= &\quad (\tfrac{1}{2}x_0Qx_0 + c^tx_0) + (\tfrac{1}{2}x_0QSy + \tfrac{1}{2}y^tS^tQx_0 + c^tSy) + (\tfrac{1}{2}y^tS^tQSy) && (20)\\
= &\quad (\tfrac{1}{2}x_0Qx_0 + c^tx_0) + (x_0QSy + c^tSy) + (\tfrac{1}{2}y^tS^tQSy) && (21)\\
= &\quad f(x) = f(x_0) + g^t(x_0)Sy + \tfrac{1}{2}y^tS^tQSy. && (22)
\end{aligned}$$

We chose x_0 such that $g(x_0) = 0$. With this, (10) is simplified to:

$$f(x) = f(x_0) + \frac{1}{2}y^tDy. \tag{23}$$

Since D is diagonal, (23) expresses the coordinates of f in y only in terms of $y_i^2, i = 1, ..., n$. In particular, (23) does not include any linear term in y and no mixed product terms $y_i, y_j,\ i, j \in \{1, \dots, n\}$.

Using (16), it is easy to plot sets of levels for the coordinates f in y. In summary, the sets of levels for f can be drawn using the following steps:

1. Calculate S and D by solving the problem of calculating the eigenvalues for Q;
2. Solve the system $Qx_0 = -c$;
3. Draw the axes $y_j|_{j=1,2,...,n}$ in the space $x_i|_{i=1,2,...,n}$ by plotting the column vectors of S centered on x_0.

3.3. Decomposition

To write a decomposition of a quadratic form in its canonical form as a difference of two convex functions, the quadratic terms are separated by the signs of the coefficients as follows:

$$\begin{aligned} f(x) = \quad & \textstyle\sum_{i=1}^{n} \alpha_i x_i + \beta_i x_i^2 = \left(\sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2\right) + \left(\sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2\right) && (24)\\ = \quad & \left(\textstyle\sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2\right) - \left(-\textstyle\sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2\right) && (25)\\ = \quad & g(x) - h(x). && (26) \end{aligned}$$

This method allows obtaining the DC decomposition for a quadratic form.

The principle of DCA is based on solving a convex quadratic problem on each iteration. To solve the convex problem, there are several efficient methods.

4. Method

In order to obtain a good initial estimate (or at least an acceptable one) of the DC algorithms for nonconvex quadratic problems, we propose three procedures as below:

Procedure 01: creating the DC initial estimate by minimizing the linear part of the quadratic function. In order to obtain this point, we use SIMPLEX algorithm [12].

While problem (1) is the target quadratic problem, procedure 01 is based on solving the following sub problem:

$$\begin{cases} \arg\min c^t x \\ Ax \leq b,\ x \geq 0 \end{cases} \tag{27}$$

We denote the solution of problem (27) by x_0^{lin}, then we have:

$$c^t x_0^{lin} = \begin{cases} \min c^t x \\ Ax \leq b,\ x \geq 0 \end{cases} \tag{28}$$

Procedure 02: to create the second initial point, we use the eigenvalues of Q to create f in its canonical form, then we can separate the quadratic terms by the signs of the coefficients as follows:

$$\begin{aligned} f(x) = \quad & \textstyle\sum_{i=1}^{n} \alpha_i x_i + \beta_i x_i^2 && (29)\\ = \quad & \left(\textstyle\sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2\right) + \left(\textstyle\sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2\right) && (30)\\ = \quad & g(x) + h(x). && (31) \end{aligned}$$

The second proposed initial point is x_0^{cvx} given by:

$$g(x_0^{cvx}) = \begin{cases} \min g(x) = \sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2 \\ Ax \leq b,\ x \geq 0 \end{cases} \tag{32}$$

Then,

$$x_0^{cvx} = \begin{cases} \arg\min g(x) = \sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2 \\ Ax \leq b,\ x \geq 0 \end{cases} \tag{33}$$

To reach this initial point, we use Interior-Point-Method integrated on Matlab under the function **quadprog**.

Procedure 03: we follow the same process in Procedure 02 for the third initial point x_0^{cve}. After the decomposition of the nonconvex quadratic function, we minimize the concave part as follows:

$$h(x_0^{cve}) = \begin{cases} \min h(x) = \sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2 \\ Ax \leq b,\ x \geq 0 \end{cases} \tag{34}$$

Then,

$$x_0^{cve} = \begin{cases} \arg\min h(x) = \sum_{\beta_i<0} \alpha_i x_i + \beta_i x_i^2 \\ Ax \le b,\ x \ge 0 \end{cases} \tag{35}$$

To reach x_0^{cve} we used the commercial software **CPLEX 12.8**[1)].

Remark 4.1. *The proposed initial estimates x_0^{lin}, x_0^{cvx} and x_0^{cve} are feasible solutions of problem* (1) *(i.e. x_0^{lin}, x_0^{cvx} and x_0^{cve} belongs to the convex constraints set of the problem* (1) *because* (1), (27), (32) *and* (34) *have the same set of constraints.*

In order to compare the three procedures, we have developed an implementation using Matlab 2018a. To reach the initial estimates, we used the following algorithms:

1. **Simlex algorithm [12]:** for solving the intermediate linear program, we used the simplex algorithm integrated in Matlab under the function **linprog**[2)];
2. **Interior-point algorithm [28]:** the function **quadprog**[3)] implemented in Matlab.
3. **Branch and bound algorithm [31]:** the branch and bound global algorithm implemented in CPLEX12.8 (the function **cplexqp**[4)] with the parameters **optimalitytarget** and **timelimit** set to 3 and 10,800 s, respectively);

4.1. Approach

We generated 107 test problems with different approaches, as detailed below in subsection 4.1.1.

- We compared the different estimates for the 107 test problems;
- The test problems were divided into four types under the names Rosen, Thoui, st, and ext;
- The dimensions of the 107 test problems were between 5 and 75;
- The number of constraints was between 1 and 50.

To assess the effect of the selection of the initial point selection method on DCA efficiency, we considered the following approaches:

1. **Linear initial point x_0^{lin}:** resulting from the minimization of the linear part of the objective function.
2. **Convex initial point x_0^{cvx}:** resulting from the minimization of the convex part of the objective function.
3. **Concave initial point x_0^{cve}:** resulting from the minimization of the concave part of the objective function.

For each initial point, we took as evaluation metrics:

1. CPU cooled time while getting the initial point.
2. The whole CPU cooled time to find the solution, including that of the initial point and that of the DCA procedure.
3. The effected number of iterations of the DCA algorithm.

We have studied each evaluation metric in function of the following variable parameters:

1. **Problem dimension:** the size of the vector c in (1).
2. **Number of constraints:** the size of the vector b in (1).
3. **Problem type:** Thoai, Rosen, ext, and st.

The test problems used in the experimental study of this paper can be downloaded here[5)]. We did not use these initial test problems. Instead, based on them, we generated new sets of test problems to meet the study's requirements, as demonstrated below.

---

1) `https://www.ibm.com/support/pages/cplex-optimization-studio-v128`
2) `https://www.mathworks.com/help/optim/ug/linprog.html`
3) `https://www.mathworks.com/help/optim/ug/quadprog.html`
4) `https://www.ibm.com/docs/en/icos/12.7.1.0?topic=apis-cplex-matlab-toolbox`
5) `https://drive.google.com/drive/folders/1zG4YcRXvzXDg6Aifph8sf3t2tBOKjbtR`

4.1.1. *Generation of test problems*

We performed the experiments on a set of 107 test problems where:

1. The test problems are divided into four types: Thoai, Rosen, ext and st; each type is generated using a different approach;
2. We generated decomposed quadratic functions as g and h separately in order to avoid calculating the eigenvalues and eigenvectors for the decomposition method described in 3.3 (especially in the case of large dimensions; $n > 40$);
3. We generated the concave part of the target quadratic problems using the algorithm of Rosen [30], Thaoi [36], and Globallib [14] (for st and ext problems);
4. The type of the test problem is characterized by the concave part, where we consider these concave problems as the initial basis for the nonconvex test problems;
5. After generating the concave part, we generated the convex part according to the dimension of the concave subproblems;
6. We generated the convex part by creating a random $n \times n - matrix\ A$, according to the dimension of the concave subproblem. Then, we calculated the matrix positive defined $A \times A^t$, which represents the quadratic matrix of the convex part;
7. We created a random vector with the same dimension of the generated test problem which is considered as the linear part of the convex part of the decomposed test problems.

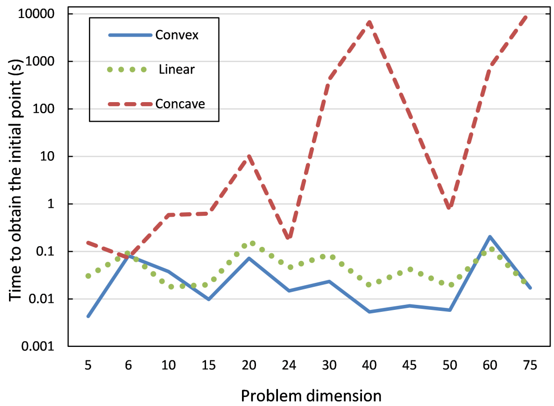
4.1.2. *Environment*

We carried out the tests on a personal computer with 8 GB random access memory capacity. The computer had an i5-7200U Intel(R) Core(TM) processor, with 4 cores each working between 2.50GHz and 2.70GHz frequency, operated with Windows 10. The work was conducted in Matlab2018a environment.

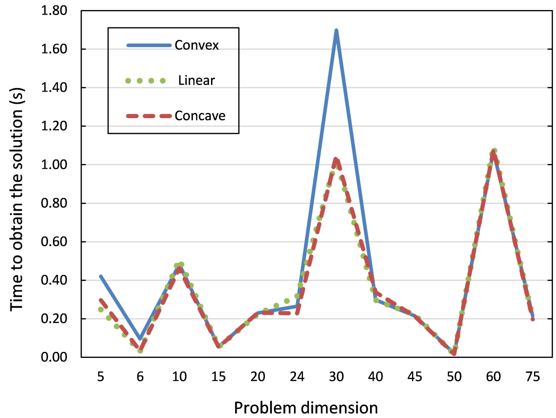
5. Results and Discussion

Following procedures described in Section 4, we obtained the results shown in the figures below. Fig.1 presents the effect of dimension on the time required to reach the initial point (Fig.1a), the time to reach the solution (Fig.1b), the time to reach both points (Fig.1c), and also on the number of performed iterations during the procedure of resolving the problem (Fig.1d).

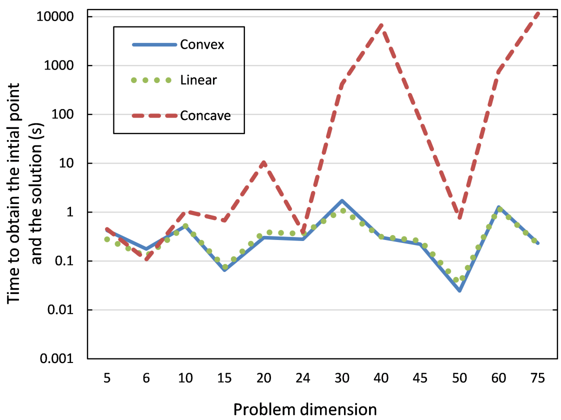
A similar description can be used for (Fig.2) and (Fig.3), but instead of dimension, these two figures cover the number of constraints and problem type, respectively.



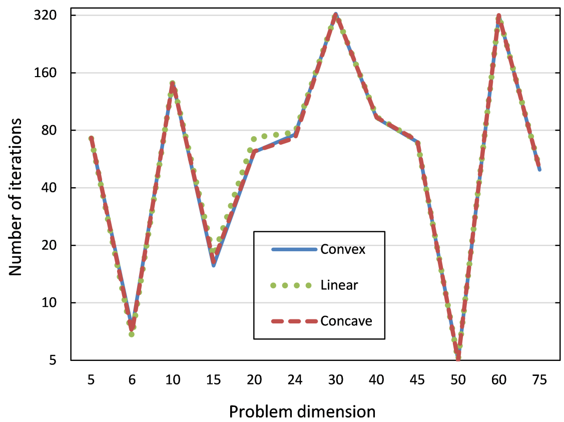
(a)



(b)



(c)



(d)

Figure 1: The effect of problem dimension on the time required to reach the initial point (a), time to reach the solution (b), the combined time (c), and the number of performed iterations during the procedure of resolving the problem (d).

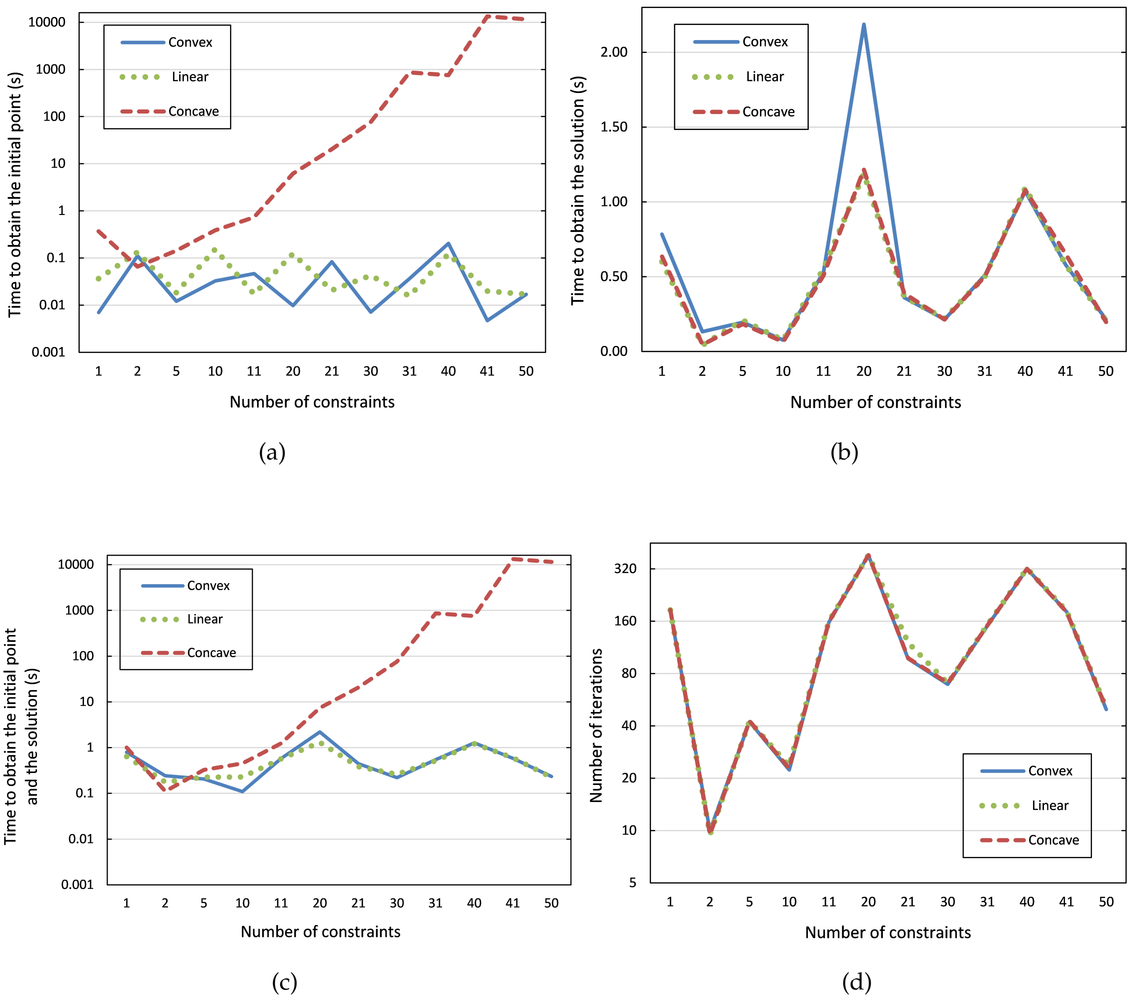


Figure 2: The effect of the number of constraints on the time required to reach the initial point (a), time to reach the solution (b), the combined time (c), and the number of performed iterations during the procedure of resolving the problem (d).

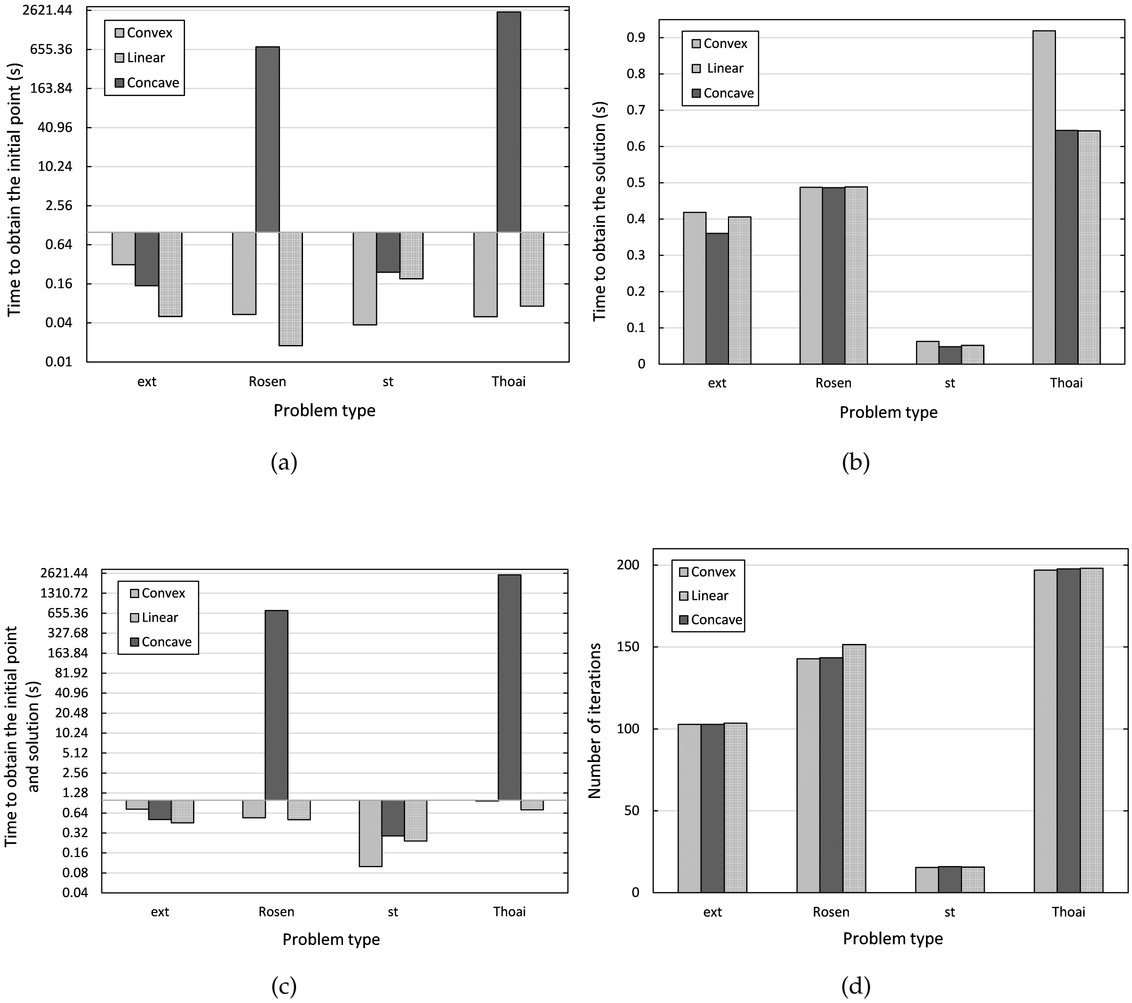


Figure 3: The effect of the type of problem on the time required to reach the initial point (a), time to reach the solution (b), the combined time (c), and the number of performed iterations during the procedure of resolving the problem (d).

Based on Fig.1, the concave approach is the slowest to reach the initial point and generally takes an expanding form proportional to the increase in dimension. This is likely because of the branch and cut algorithm that takes a long time to find the minimum of concave quadratic problem, especially in the case of high dimensions. On the other hand, the linear approach performed generally better than the convex one (notice the logscale). This could be explained by the efficiency of the SIMPLEX algorithm to calculate x_0^{lin}.

In Fig.1b, we note the overlap of the three approaches (except at dimension 30 for the convex approach, which we consider as an outlier).

The most practical measure is the overall time of resolution, which is presented in Fig.1c. We see the large increase in resolution time of the concave method with the increase in dimension.

Contrary to the concave approach, the dimension seems to have no effect on the resolution time of convex and linear approaches; the two curves are almost overlapped. The number of iterations in the three approaches is almost the same and seems to have no correlation with the problem dimension.

Surprisingly, the three approaches had a similar behavior as a function of the number of constraints as the number of dimensions (Fig.2). We may explain this trend as a proposition that the DCA of the three initial estimates are close to each other (so the initial time to find the initial estimate is the most important).

In Fig.3a, we notice different timing behavior for the approaches with each problem type. The most important observation is the irrational results of the concave approach with Rosen and Thaoi problem types. These long times with the two problems are caused by the large dimensions and the number of the constraints of Rosen and Thaoi problems.

The solution time was similar with the three methods for all problems and varied from one problem to another with st showing the best results.

For the general solution time, we see that the linear approach did the best except with st type, where the concave approach outperformed the linear one. The number of iterations with each approach were similar for each problem type.

A clear observation is that the best initial estimate for the DCA according to all the criteria used in this paper is x_0^{lin}. Therefore, we came up with a new algorithm that we called Initialized Difference of Convex Functions Algorithm (IDCA). This algorithm allows us to calculate the minimum of a nonconvex quadratic problem with a convex constraint set using a DC algorithm, without the need of the initial point, since IDCA calculates the best initial point for DCA. The IDCA is described below 3

Algorithm 3 Initialized DC Algorithm

insert the objective function $f(x) = \frac{1}{2}x^tQx + c^tx$ and the convex constraint set Ω
calculate x_0^{lin} by solving the linear optimization problem (27): $arg\min_{x\in\Omega} c^tx$
$x^0 = x_0^{lin}$
$k \leftarrow 0$
$\varepsilon > 0$ ▷ a defined precision
Step 1 : We calculate $y^k \in \partial h(x^k) = \nabla h(x^k)$
Step 2 : We determine $x^{k+1} \in \partial g^*(y^k)$ ▷ by solving the following convexe quadratic problem:

$$(P_c) \quad \left\{Min \quad g(x,y) - \left\langle x, y^k \right\rangle : x \in \Omega\right\}$$

if $\mid (g-h)(x^{k+1}) - (g-h)(x^k) \mid \leq \varepsilon$ or $\|x^{k+1} - x^k\| \leq \varepsilon$ **then**
  stop ▷ the stopping condition is satisfied
else
  $k++$
  goto *Step* 1 :
end if

6. Conclusion

From the performed tests, we have the following observations:

- The entire solution time with the concave approach increased dramatically with the problem dimension and number of constraints. It is clearly not an optimal choice for high configurations;
- The problem dimension and number of constraints have no clear effect on the iteration number of problem resolution (in the cases we explored);
- The linear approach performs best with ext, Rosen and Thaoi problem types, and the convex approach performs best for st;
- The concave approach takes a long time with the Rosen and Thaoi types of problem due to the relatively large dimensions and constraints number of these problem types;

- The best initial estimate for the DCA according to all of the criteria used in this paper is x_0^{lin};
- We developed a new algorithm, Initialized Difference of Convex Functions Algorithm (IDCA 3), which allows us to calculate the minimum of a nonconvex quadratic problem with a convex constraint set using DC algorithm (without the need for determination of the initial estimate).

As a future direction, it is recommended to compare the new IDCA against existing DC decomposition methods in order to create a complete evaluation of these algorithms.

References

[1] S. Achour, *Experimental Study on the Effect of the Initial Point of the Difference of Convex Functions Algorithm on Solving Nonconvex Quadratic Problems*, Algerian Journal of Engineering Architecture and Urbanism, **5**(2)(2021), 458–467.

[2] A. O. Akdemir, S. I. Butt, M. Nadeem, M. A. Ragusa, *Some new integral inequalities for a general variant of polynomial convex functions*, AIMS Math., **7**(2022), 20461–20489.

[3] A. O. Akdemir, S. Aslan, M. A. Dokuyucu, E. Çelik, *Exponentially Convex Functions on the Coordinates and Novel Estimations via Riemann-Liouville Fractional Operator*, Journal of Function Spaces, **2023**(2023).

[4] A. D. Aleksandrov, *On the surfaces representable as difference of convex functions*, Sib. Élektron. Mat. Izv., **9**(2012), 360–376.

[5] O. Alabdali, G. Allal, *Optimal estimates of approximation errors for strongly positive linear operators on convex polytopes*, Filomat, **36**(2)(2022), 695–701.

[6] M. Al Kharboutly, *Résolution d'un problème quadratique non convexe avec contraintes mixtes par les techniques de l'optimisation DC*, PhD diss., Normandie Université, (2018).

[7] K. Aoki, T. Satoh, *Economic Dispatch with Network Security Constraints Using Parametric Quadratic Programming*, IEEE Transactions on Power Apparatus and Systems, **12**(1982), 4548–4556.

[8] M. Brand, D. Chen, *Method for performing image processing applications using quadratic programming*, U.S. Patent, **8**(2014), 533–761.

[9] J. Chen, S. Burer, *Globally solving nonconvex quadratic programming problems via completely positive programming*. Math.Prog.Comp., **1**(1)(2012), 33–52.

[10] A. Chikhaoui, B. Djebbar, A. Belabbaci, A. Mokhtari, *Optimization of a quadratic function under its canonical form*, Asian journal of applied sciences, **2**(6)(2009), 499–510.

[11] G. Cornuejols, R. Tütüncü, *Optimization methods in finance*, Vol. 8, Cambridge University Press, 2006.

[12] G. B. Dantzig, *Linear programming and extensions*, Princeton university press, 1963.

[13] S. Gao, D. Bertrand, N. Nagarajan, *FinIS: Improved in silico Finishing Using an Exact Quadratic Programming Formulation*. Algorithms in Bioinformatics: 12th International Workshop, WABI 2012, Ljubljana, Slovenia, (2012), 314–325.

[14] Globallib: Gamsworld global optimization library. `http://www.gamsworld.org/global/globallib.htm`.

[15] E. Haber, U. M. Ascher, D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse problems, **16**(5)(2000), 1263.

[16] P. Hansen, B. Jaumard, M. Ruiz, J. Xiong, *Global Minimization of Indefinite Quadratic Functions Subject to Box Constraints*, Naval Research Logistics, **40**(3)(1993).

[17] P. Hartman, *On functions representable as a difference of convex functions*, Pacific J. Math., **9**(1959), 707–713.

[18] L. T. Hoai An, *An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints*, Mathematical Programming, **87**(3)(2000), 401–426.

[19] L. T. Hoai An, P. D. Tao, *A continuous approach for large-scale constrained quadratic zero-one programming*, Optimization, **45**(3)(2001), 1–28.

[20] R. Horst, P. M. Pardalos, *Handbook of global optimization: Nonconvex Optimization and Its Applications*, Vol. 2, Springer New York, 1995.

[21] J. L. Hu, Z. Wu, H. McCann, L. E. Davis, C. G. Xie, *Sequential quadratic programming method for solution of electromagnetic inverse problems*, IEEE transactions on antennas and propagation, **53**(8)(2005), 2680–2687.

[22] H. Konno, *An algorithm for solving bilinear knapsack problems*, Journal of the Operations Research Society of Japan, **24**(4)(1981), 360–374.

[23] E. L. Lawler, *The quadratic assignment problem*, Management science, **9**(4)(1963), 586–599.

[24] H. Leonpacher, S. S. Douglas, N. H. Woolley, D. Kraft, *Simulation and Optimization of Logistic Processes Involving Sloshing Media*, High Performance Scientific and Engineering Computing: Proceedings of the International FORTWIHR Conference on HPSEC, Munich, **3**(1999), 209–219.

[25] L. D. Muu, T. Q. Phong, P. D. Tao, *Decomposition methods for solving a class of nonconvex programming problems dealing with bilinear and quadratic functions*, Computational Optimization and Applications, **4**(1995), 203–216.

[26] S. Nordebo, Z. Zang, I. Claesson, *A semi-infinite quadratic programming algorithm with applications to array pattern synthesis*, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, **48**(3)(2001), 225–232.

[27] P. M. Pardalos, M. Panos , G. P. Rodgers, *Computational aspects of a branch and bound algorithm for quadratic zero-one programming*, Computing, **45**(2)(1990), 131–144.

[28] F. A. Potra, S. J. Wright, *Interior-point methods*, Journal of Computational and Applied Mathematics, **124**(1-2)(2000), 281–302.

[29] M. A. Z. Raja, M. A. Manzar, S. M. Shah, Y. Chen, *Integrated intelligence of fractional neural networks and sequential quadratic programming for Bagley-Torvik systems arising in fluid mechanics*. Journal of Computational and Nonlinear Dynamics, **15**(5)(2020), 051003.

[30] Y. Y. Sung, J. B. Rosen, *Global minimum test problem construction*, Math Program, **24**(1)(1982), 353–355.

[31] Studio, I. I. C. O. V12. 8.0, 2018. URL `http://www-01.ibm.com/support/docview.wss`.

[32] P. D. Tao, *Algorithmes de calcul d'une forme quadratique sur la boule unité de la norme maximum*, Numerische Mathematik, **45**(1985), 377–440.

[33] P. D. Tao, *Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients optimization*, North-Holland Mathematics Studies, **129**(1986), 249–271.

[34] P. D. Tao, *Duality in D.C. (difference of convex functions) optimization*, International Series of Numer. Math., **84**(1988).

[35] M. Telli, M. Bentobache, A. Mokhtari, *A successive linear approximation algorithm for the global minimization of a concave quadratic program*, Computational and Applied Mathematics, **39**(4)(2020), 1–28.

[36] N. V. Thoai, *On the construction of test problems for concave minimization algorithms*, J. Glob. Optim., **5**(4)(1994), 399–402.

[37] T. B. Trafalis, H. Ince, *Support vector machine for regression and applications to financial forecasting*, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, Neural Computing: New Challenges and Perspectives for the New Millennium IEEE, **6**(2000), 348–353.

[38] H. Tuy, *Nonconvex Quadratic Programming*, Convex Analysis and Global Optimization, (2016), 337–390.

[39] S. Wenzel, R. Paulen, S. Krämer, B. Beisheim, S. Engell, *Shared resource allocation in an integrated petrochemical site by price-based coordination using quadratic approximation*, Proc. of IEEE European Control Conference, (2016), 1045–1050.

[40] L. Yanning, E. Canepa, C. Claudel, *Optimal control of scalar conservation laws using linear/quadratic programming: Application to transportation networks*, IEEE Transactions on Control of Network Systems, **1**(1)(2014), 28–39.