New Algorithms for maximizing the difference of convex functions

Aharon Ben-Tal

Technion - Israel Institute of Technology, abental@technion.ac.il

Luba Tetruashvili

Afeka - Academic College of Engineering in Tel-Aviv, lubat@afeka.ac.il

Abstract

Maximizing the difference of 2 convex functions over a convex feasible set (the so called DCA problem) is a hard problem. There is a large number of publications addressing this problem. Many of them are variations of widely used DCA algorithm [20]. The success of this algorithm to reach a good approximation of a global optimum, depends crucially on the choice of its starting point. In the algorithm developed in our paper MDCF (Maximizing the Difference of Convex Functions) a major effort is to generate a good starting point. This is obtained by using the COMAX algorithm for maximizing a convex function [6]. The solution found by COMAX is a basis for obtaining a good strating point for MDCF.

Another contribution of the paper is the algorithm for solving problems with an indefinite quadratic objective function and compact and convex feasible set. The problem is first converted to maximizing a difference of convex quadratic functions. The new algorithm QMDCF is a specific adaptation of MDCF to this case.

The performance of the two new algorithms developed in the paper is tested numerically, and results are compared to the performance of classical DCA, and some other algorithms.

1 Introduction

In this paper we develop an algorithms for the problem of maximazing the difference of convex functions (DCF):

$$\max_{x \in X} \{f(x) - g(x)\}, \tag{1.1}$$

where the feasible set $X = \{x \in R^n | h_i(x) \leq 0, i = 1, ..., m\}$ is compact and all involved functions $f : R^n \to R$, $g : R^n \to R$ and $h_i : R^n \to R$ $i = 1, ..., m$ are convex C^2 (twice continuously differentiable functions). Problem DCF is nonconvex and covers many nonconvex optimization problems. In fact, in 1972 Hartman [10] proved that any nonconvex problem with C^2 functions can be remodeled as a DCF problem.

There are several types of approaches for solving DCF problem: Technics like branch and bound, Semidefinite programming, Use of coopositive programming and more.

Perhaps the most popular type of algorithms for solving DCF is DCA [20], [21] and references therein.

Like many algorithms for nonconvex optimization, DCA suffers from what could be labeled as "the curse of needing a starting point x^0". Deriving such good x^0 is an intractable problem in itself, and strongly influences to which point DCA will converge. In numerical studies typically a large number of randomly chosen feasible points are generated, of which the best one is chosen as x^0. In many reports of numerical studies this value of x^0 is not always exposed, and even if it does, the time to obtain it is not always counted in the total running time of the complete algorithm.

The algorithm developed in this paper MDCF (Maximizing Difference of Convex Functions) significantly differs from DCA. Unlike DCA which linearize the "bad function" $f(x)$:

$$L_f(x, x^0) := f(x^0) + \nabla f(x^0)^T(x - x^0),$$

MDCF linearize a "good function" $g(x)$ around x_g-the minimizer of $g(x)$ over $x \in X$:

$$L_g(x, x_g) := g(x_g) + \nabla g(x_g)^T(x - x_g).$$

Note, that computing x_g is a tractable convex optimization problem. After replacing $g(x)$ by the linear function $L_g(x, x_g)$ one is left with the problem of maximization of a convex function:

$$\max_{x \in X} f(x) - L_g(x, x_g).$$

For such problems we use the recent COMAX algorithm, published in [6], and summarised here in section 2. The full MDCF algorithm does not need to guess a starting point, and moreover uses only tractable convex problems, directly or as one having a "hidden convexity property".

Another contribution of the current work is a new method for solving optimization problems with general quadratic objective function and compact and convex feasible set. This problem was studied extensively in many works [13], [14],[19], [12], [11], [16], [17], to mention just few. The method proposed in [13] uses branching method which is based on the first-order KKT conditions and semidefinite relaxations of copositive programs. A different branch and bound (BB) based method for solving general quadratic problem with box constraints can be found in [14]. The use of branch and bound ideas is widespread approach to tackle general quadratic problem. Variations of BB approach can be found in [19].

A version of MDCF for general quadratic problems (QMDCF) is based on a specified approach which is less computationally expensive and more suitable for large scale problems.

The paper is organized as follows. In Section 2 the COMAX algorithm is reviewed on which our method is relied. In Section 3 we present our new method MDCF for general smooth DCF problems. In section 4 we present the new method QMDCF for optimization of nonconvex quadratic function. The results of numerical experiments are presented in Section 5.

2 Review of COMAX

COMAX is a published algorithm [6] for maximization of a convex objective function over a convex feasible set:

$$\max_{x \in R^n} \{f(x) | h_i(x) \leq 0, i = 1, ..., m\}, \qquad (2)$$

where f, h_i $i = 1, ..., n$ are convex and twice differentiable functions and the feasible set $X = \{h_i(x) \leq 0, i = 1, ..., m\}$ is compact. Below we give a short description of COMAX.

COMAX algorithm
Phase 1: finding a "good" starting point

1. Approximation of the objective function $f(x)$
    - Find $\bar{x} = \operatorname{argmin}\{f(x) | x \in X\}$. This a tractable convex optimization problem.

- Approximate the function $f(x)$ by its' second-order Taylor expansion around $\bar{x}$:

$$q(x) := f(\bar{x}) + (x - \bar{x})^T(\nabla f(\bar{x}))) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\bar{x})(x - \bar{x})$$

2. Approximation of the feasible set $X := \{h_i \leq 0, i = 1, ..., m\}$ by an ellipse [5]:

   - Find the analytic center x_{ac} of the feasible set X by solving the following convex problem:

$$x_{ac} = \text{argmin}\{\phi(x) = -\sum_{i=1}^{m} \log -h_i(x)\}.$$

   - Set $X_1 := \{x|(x - x_{ac})'\nabla^2\phi(x)(x - x_{ac}) \leq 1\}$, X_1 is the inscribed ellipsoid of X around x_{ac}.
   - Set $X_2 := \{x|\frac{1}{((1+2/sqrt(m))*m)^2}(x - x_{ac})'\nabla^2\phi(x)(x - x_{ac}) \leq 1\}$, X_2 is the circumscribing ellipsoid of X around x_{ac}.

3. Use hidden convexity method [7] to find $x_{in} = \text{argmax}\{q(x)|x \in X_1\}$.

4. Use hidden convexity method to find $x_{out} = \text{argmax}\{q(x)|x \in X_2\}$.

5. Use line search to find $x_{mid} := \alpha x_{in} + (1 - \alpha)$, $\alpha \in (0, 1)$ such that $x_{mid} \in X$.

Phase 2: convergence to local/global solution

Apply the Gradient Ascent method to solve the problem

$$\max_{x \in X}\{f(x)\}$$

- Step 1: $x^0 := x_{mid}$
- Step k: Set $x^k := \text{argmax}\{(\nabla f(x^{k-1})|x \in X\}$
- Output: x^N.

The approximation of the feasible set by an ellipse is not the only approximation used in COMAX. For more details, see [6].

When the feasible set is a polytope $X := \{x \in R^n | Ax \leq b, A \in R^{m \times n}, b \in R^m\}$ we also can approximate X by so called Chebyshev ball. To find the center (Chebyshev center) x_{ch} of largest inscribed ball of the polytope X the following linear programming problem should be solved:

$$\begin{aligned} \max_{r, x_{ch}} \quad & r \\ \text{s.t.} \quad & a_i x_{ch} + \|a_i\| r \leq b_i, \;\; i = 1, ..., m \\ & r \geq 0 \end{aligned} \tag{2.1}$$

Remark: In COMAX there are more options to approximate the feasible set. See [6] section 5.1.

3 The MDCF algorithm

The classical DCA algorithm starts with a "guess" of a feasible starting point x^0 around which the function $f(x)$ is linearized, and the following convex problem $\max_{x \in X}\{L_f(x, x^0) - g(x)\}$ is solved. The convergence to a global solution of problem DCF depends on the choice of x^0 around which the first linearization of $f(x)$ is constructed. Our method MDCF starts with linearization of **the good function** $g(x)$ around its minimum x_g and the obtained problem is

$\max_{x \in X}\{f(x) - L_g(x, x_g)\}$ which is in our algorithm solved by COMAX. This is done in the first stage of MDCF which aims to find a "promising" starting point. In the second stage of MDCF we use a standard gradient ascent algorithm in which at each iteration one solves the convex problem $\max_{x \in X}\{L_f(x, x_{it}) - g(x)\}$, where $L_f(x, x_{it})$ denotes the linearization of $f(x)$ around the solution from the previous iteration. Below we describe MDCF in details.

The new method MDCF
Stage 1: finding a "good" starting point by COMAX algorithm

1. Approximation of the objective function $f(x) - g(x)$ by a quadratic function:
   - Find minimum of $g(x)$, $x_g = \text{argmin}\{g(x)|x \in X\}$.
   - Find $\bar{x} = \text{argmin}\{f(x) - \nabla g(x_g)^T x | x \in X\}$.
   - Approximate the function $f(x) - \nabla g(x_g)^T x$ by the second-order Taylor expansion around $\bar{x}$, which is the quadratic function:

$$q(x) := f(\bar{x}) - \nabla g(x_g)^T \bar{x} + (x - \bar{x})^T (\nabla f(\bar{x}) - \nabla g(x_g)) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\bar{x})(x - \bar{x})$$

2. Approximation of the feasible set $X := \{h_i \leq 0, i = 1, ..., m\}$ by an ellipse [5]:
   - Find the analytic center x_{ac} of the feasible set X by solving the following problem:

$$x_{ac} = \text{argmin}\{\phi(x) = -\sum_{i=1}^{m} \log -h_i(x)\}.$$

   - Set $X_1 := \{x|(x - x_{ac})' \nabla^2 \phi(x)(x - x_{ac}) \leq 1\}$, X_1 is the inscribed ellipsoid of X around x_{ac}.
   - Set $X_2 := \{x| \frac{1}{((1+2/sqrt(m))*m)^2}(x - x_{ac})' \nabla^2 \phi(x)(x - x_{ac}) \leq 1\}$, X_2 is the circumscribing ellipsoid of X around x_{ac}.

3. Use hidden convexity method to find $x_{in} = \text{argmax}\{q(x)|x \in X_1\}$.

4. Use hidden convexity method to find $x_{out} = \text{argmax}\{q(x)|x \in X_2\}$.

5. Use line search to find $x_{mid} := \alpha x_{in} + (1 - \alpha)$, $\alpha \in (0, 1)$ such that $x_{mid} \in X$.

6. Apply the Gradient Ascent method to solve the problem

$$\max_{x \in X}\{f(x) - \nabla g(x_g)^T x\}$$

   - Step 1: $x^0 := x_{mid}$
   - Step k: Set $x^k := \text{argmax}\{(\nabla f(x^{k-1}) - \nabla g(x_g))^T x | x \in X\}$
   - Output: x^N
   - Set $x_{start} := x^N$.

Stage 2: convergence to local/global maximum

- Replace $f(x)$ by its linear approximation at the point x_{start}. Set $x^0 := \text{argmax}\{\nabla f(x_{start})^T x - g(x)|x \in X\}$.

- Step k: Set $x^k := \text{argmax}\{\nabla f(x^{k-1})^T x - g(x)|x \in X\}$.

4 QMDCF algorithm for problems with general quadratic objective function

In this section we describe a version of MDCF method called QMDCF adapted to general quadratic problem:

$$\max_{x \in R^n} \{q(x) = 0.5x^T Dx + b^T x | h_i(x) \leq 0, i = 1, ..., m\}, \qquad (2)$$

where $D \in R^{n \times n}$ is an indefinite symmetric matrix and $b \in R^n$ and the feasible set $X := \{h_i \leq 0, i = 1, ..., m\}$ is convex and bounded. The idea is to rewrite the general quadratic function as DC function and apply MDCF with slight changes as described below.

QMDCF for problems with general quadratic objective function

Stage 1: finding a "good" starting point for DCA method

1. **Converting problem to DC problem** Define diagonal matrix A such that its' diagonal entry in row i is equal to the sum of all absolute values in row i of matrix D plus any positive value. Set $D_1 = D + A$ and $D_2 = A$. Note, that D_1 is a positive definite matrix since it has a dominant positive diagonal.

2. Set $f(x) := 0.5x^T D_1 x + b^T x$ and $g(x) := 0.5x^T D_2 x$. Thus $q(x) = f(x) - g(x)$.

3. Approximation of the feasible set $X := \{h_i \leq 0, i = 1, ..., m\}$ by an ellipsoid constraint:

   - Find the analytic center x_{ac} of the feasible set X by solving the following problem:

   $$x_{ac} = \text{argmin}\{\phi(x) = -\sum_{i=1}^{m} \log -h_i(x)\}.$$

   - Set $X_1 := \{x | (x - x_{ac})' \nabla^2 \phi(x)(x - x_{ac}) \leq 1\}$, X_1 is the inscribed ellipsoid of X around x_{ac}.
   - Set $X_2 := \{x | \frac{1}{((1+2/sqrt(m))*m)^2}(x - x_{ac})' \nabla^2 \phi(x)(x - x_{ac}) \leq 1\}$, X_2 is the circumscribing ellipsoid of X around x_{ac}.

4. Use hidden convexity method to find $x_{in} = \text{argmax}\{q(x) | x \in X_1\}$.

5. Use hidden convexity method to find $x_{out} = \text{argmax}\{q(x) | x \in X_2\}$.

6. Use line search to find $x_{mid} := \alpha x_{in} + (1 - \alpha)$, $\alpha \in (0, 1)$ such that $x_{mid} \in X$.

Stage 2: convergence to local/global maximum

- Replace $f(x)$ by its linear approximation at the point x_{mid}. Set $x^0 := \text{argmax}\{\nabla f(x_{mid})^T x - g(x) | x \in X\}$.

- Step k: Set $x^k := \text{argmax}\{\nabla f(x^{k-1})^T x - g(x) | x \in X\}$.

5 Numerical experiments

The comparison between our method MDCF and the DCA algorithm was one of the goals in our numerical experiments. Since the second stage in MDCF coincides with the main steps in DCA method, the two methods differ in the choice of the starting point. In the original version of the DCA method, the starting point is chosen usually randomly. Here in all examples DCA was run 100 times from 100 different randomly chosen starting points. As we report below in all examples our MDCF algorithm reaches the larger values in all the objective functions.

Another goal was to check the ability of MDCF to converge to a global optimum. For this purpose we tested nonconvex problems with known solutions.

5.1 The difference of quadratic convex function and the logsumexp function

In this section we generated examples of DCF problems, where $f(x)$ is a convex quadratic function and $g(x) = log \sum_{i=1}^{k} \exp(a_i^T x + b_i)$. So the DCF problem is:

$$\max\{F(x) := 0.5x^T Dx + c^T x - \log \sum_{i=1}^{k} \exp(a_i^T x + b_i) | x \in X\},$$

and the feasible set $X \subset R^n$ is an intersection of three ellipsoids. We generated 10 examples with different values of n and k. In all example, DCA method was run 100 times from randomly chosen starting points which were chosen uniformly from the interval $[-1000, 1000]^n$. MDCF was run just once. Fot DCA we report the distribution of the best obtained values of objective function.

	n	k	Value (MDCF)	Value (DCA)	Value (DCA)	Time(MDCF)	Time (DCA)
Ex.1	100	10	**42.45**	40.83(53%)	42.45(47%)	1.12	39.45
Ex.2	100	20	**281356**	275582(51%)	281356(49%)	1.42	365.82
Ex.3	100	20	**280400**	272858(50%)	280400(50%)	1.57	327.58
Ex.4	200	10	**1129166**	583898(22%)	1129166(78%)	0.89	139.74
Ex.5	200	10	**2539955**	1431726(14%)	2539955(86%)	1.19	171.72
Ex.6	200	10	**11415**	5807(48%)	11415(52%)	0.73	44.55
Ex.7	500	5	**70384**	15168(37%)	70384(63%)	1.42	241.2
Ex.8	500	5	**28279363**	6444318(2%)	28279363(98%)	4.62	600.15
Ex.9	1000	5	**28158328**	383084(1%)	28158328(99%)	11.11	1317.85
Ex.10	1000	5	**7039468**	95598(97%)	7039468(3%)	8.14	751.72

Table 1: Objective values and running time in seconds of MDCF and DCA on qudratic minus logsumexp.

As we can see in the table DCA recognized two local points, while MDCF was run ones and succeeded to converge to the point with the larger value in a significant shorter time. Note in particular Example 10, where DCA reached the best value only in 3% of the one hundred problems.

5.2 Maximization of convex function

The goal of tests in this section was to check the ability of our method to reach the optimal solution. For this purpose we took nonconvex maximization problems with known solutions from the article of Ben-Tal, Selvi [4]. By adding and extracting the same convex function from the convex objective function we generated problems in DCF form. We solved 7 problems listed in Table 2.

Problem 1

$$\begin{aligned}
\max_{x\in R^{20}} \quad & 0.5\sum_{i=1}^{20}(x_i-2)^2+\sum_{i=1}^{3}\exp(a_i^Tx)-\sum_{i=1}^{3}\exp(a_i^Tx) && (5.1)\\
s.t. \quad & Dx \le d && (5.2)\\
& x \ge 0, && (5.3)
\end{aligned}$$

Problem 2

$$\begin{aligned}
\max_{x\in R^{20}} \quad & 0.5\sum_{i=1}^{20}(x_i+5)^2+\sum_{i=1}^{3}\exp(a_i^Tx)-\sum_{i=1}^{3}\exp(a_i^Tx) && (5.4)\\
s.t. \quad & Dx \le d && (5.5)\\
& x \ge 0, && (5.6)
\end{aligned}$$

where

$$D^\top = \begin{bmatrix}
-3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & 1\\
7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & -1 & 1\\
0 & -5 & 1 & 1 & 0 & 2 & -1 & -1 & -9 & 1\\
-5 & 1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 1\\
1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 1\\
1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 1\\
0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1\\
2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 1\\
-1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 7 & 1\\
-1 & -9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & 1\\
-9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & 1\\
3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & 1\\
5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 1\\
0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 1\\
0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & 1\\
1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1\\
7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1\\
-7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 1\\
-4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 1\\
-6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & 1
\end{bmatrix}, d = \begin{bmatrix} -5\\ 2\\ -1\\ -3\\ 5\\ 4\\ -1\\ 0\\ 9\\ 40. \end{bmatrix}$$

and $a \in R^{20}$ is chosen uniformly.

Problem 3-7

$$\begin{aligned}
\max_{x\in R^{n}} \quad & x^TL^TLx+\sum_{i=1}^{3}\exp(a_i^Tx)-\sum_{i=1}^{3}\exp(a_i^Tx) && (5.7)\\
s.t. \quad & Dx \le d && (5.8)\\
& 0 \le x \le x_u, && (5.9)
\end{aligned}$$

where $L \in R^{m\times n}$ and $D \in R^{q\times n}$ are matrices generated randomly with entries uniformly sampled from 0-1 and d has integer entries sampled uniformly from a range (d_l, d_u).
As can be seen, in Table 2, in five problems the objective value reached by MDCF was equal or slightly better (larger) and just in two problems the objective value was smaller but very close.

5.3 Quadratic objective function over polyhedral set

In this section we tested performance of QMDCF for linearly constrained quadratic problem:

$$\max_{x\in R^n}\{0.5x^TDx+b^Tx | l \le x \le u, a^Tx \le c\},$$

a, b, l, u are vectors in R^n, c is a real scalar, $D \in R^{n\times n}$ is an indefinite symmetric matrix where all parameters involved in the model D, b, l, u, a, c were generated randomly from the uniform

	n	q	x_u	d_l	d_u	Value (MDCF)	Best known value	Time (MDCF)
P.1	20	10				**394.75**	394.75	1.53
P.2	20	10				**884.75**	884.74	1.57
P.3	10	15	5	20	60	**4674.7**	4674.7	0.92
P.4	50	62	3	30	60	175704	175710	3.86
P.5	100	130	2	80	120	**692624**	692610	10.64
P.6	200	240	2	160	240	6020787	6020800	37.6
P.7	240	280	1	150	300	**1855740**	1855700	47.6

Table 2: Objective values of MDCF and running time on problem of maximization of convex function.

distribution. Note that the feasible set is bounded as intersection of box and linear constraint. We generated four examples of dimensions 200, 500, 700 and 1000. In each example DCA was run 100 times from 100 randomly chosen starting points. In the following table, next to each of the obtained values of the objective function we indicated the percentage of the problems with the same value. In all examples QMDCF method reached larger values (in bold inside the table). For DCA results: two numbers separated by dash denote minimal and maximal values of the objective function. The percentage next to the numbers denotes the part of runs in which the values were between theses two values.

n	DCA	DCA	DCA	DCA	DCA	DCA	Time	QMDCF
200	9.83(6%)	10.06(6%)	12.06(11%)	14.42(28%)	14.49(18%)	**18.57**(31%)	11.9s	**18.57** 1.74s
500	569771- 696383(30%)	702062- 797640(49%)	802915- 816059 (8%)	821612- 838567 (7%)	842576- 870341 (4%)	892028- 906125 (2%)	28.75s	**985849** 2.43s
700	1849341(3%)	1851133 (5%)	1851466- 1856063(14%)	1856165- 1856442(41%)	**1856607**(37%)		34.15s	**1856607** 3.09s
1000	2392345- 2429108(24%)	2430199- 2439639(26%)	2440139- 2457318(39%)	2460063- 2469670(9%)	2479688- 2487927(2%)		47.49s	**2524909** 11.92s

Table 3: Objective values and running time in seconds of MDCF and DCA on quadratic over polyhedral set.

In two of the four problems DCA did not reach the best value. This examples illustrates the advantage of using QMDCF. Note, that for the case $n = 1000$ the best (non optimal) value was obtained only in 2% of the starting points and and it has the maximal value short of the value obtained by QMDCF by 36982.

5.4 Problems with known optimal values from Enhbat (1996)

In this section we tested MDCF on the following simple noneconvex problems solved in the article of Enhbat [2]. The solutions for the problems are known.

$$\max\{\sum_{i=1}^{n}(n-1-0.1i)x_i^2| -l-i \le x_i \le 1+5i,\ i=1,\dots,n\}, \quad (P10)$$

$$\max_{x\in R^n}\{x^T Cx| -(n-i+1) \le x_i \le n+0.5i,\ i=1,\dots,n\}, \quad (P12)$$

where

$$C = \begin{bmatrix} n & n-1 & n-2 & \dots & 2 & 1 \\ n-1 & n & n-1 & \dots & 3 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & 3 & \dots & n-1 & n \end{bmatrix}.$$

The objective convex quadratic functions were rewritten as the difference of two convex quadratic functions. This was obtained by adding and extracting randomly generated convex quadratic part. The results are summarized in the following table.

	n	Enhbat(1996)	Value (MDCF)
P10	3	721.4	721.4
P10	10	83712	83712
P10	30	6440531	6440531
P10	60	101506747	101506747
P10	80	319560716	319560716
P10	100	778330545	778330545
P10	150	3927744505	3927744505
P12	2	45.5	45.5
P12	5	3604	3604
P12	10	109333.5	109333.5
P12	30	25766625.5	25766625.5
P12	40	108196334	108196334
P12	70	1767930209	1767930209
P12	80	3444342668	3444342668
P12	90	6203290501	6203290501
P12	99	9986343609	9986343609

Table 4: MDCF on test problems with known global solutions from Enhbat(1996) .

In all problems the new method attained the optimal value.

5.5 QMDCF vs semidefinite and copositive programming

In this section we demonstrate performance of QMDCF algorithm as applied to problems of minimizing a (non-convex) quadratic function over a simplex presented in the article "Solving standard quadratic optimization problems via semidefinite and copositive programming" [12]. We solved four instances of the standard quadratic optimization of the following form:

$$\min_{x \in R^n} \{x^T Q x | \sum_{i=1}^{n} x_i = 1, x \geq 0\},$$

where Q is a symmetric indefinite matrix in which all entries are non negative. We reformulated the problem in the DCF form by adding and extracting the leading diagonal matrix and we also replaced the equality constraint by an inequality constraint:

$$\min_{x \in R^n} \{x^T Q x | \sum_{i=1}^{n} x_i \geq 1, x \geq 0\}.$$

These two problems are equivalent since in all optimal points in the new problem the constraint $\sum_{i=1}^{n} x_i = 1$ holds true. Example 5.1 corresponds to a computation of the largest stable set in a pentagon. Example 5.2 deals with a computation of the largest stable set in the complement of the graph of an icosahedron. Example 5.3 is a mathematical model in population genetics. Example 5.4 deals with portfolio optimization and is taken from [18]. In all examples our method found the optimal values while the method presented in [12] did not find the optimal value in Example 5.2.

Example	n	The known optimal values	Objective value (QMDCF)	Time(in seconds)
5.1	5	0.5	0.5	0.56
5.2	12	0.333	0.333	0.52
5.3	5	16.333	16.333	0.51
5.4	5	0.4839	0.4839	2.41

Table 5: Test problems from Bomze [12].

5.6 One more test problem

The following problem is taken from Burer and Letchford [8].

$$(\text{BL}): \max_{x \in R^n} \{x^T Q x + q^T x | 0 \leq x_i \leq 1, i = 1, ..., n\},$$

where

$$Q = \begin{pmatrix} -2.25 & -3 & -3 \\ -3 & 0 & -0.5 \\ -3 & -0.5 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}.$$

The exact optimal value of this problem is 1 and our method reached this value and found optimal solution in less than a second. The BL problem is known as a hard problem and until recently the exact solution could be obtained only with the use of complicated technics such as spatial branching, dynamically generated cutting planes, or an extended-variable formulation. Another recent method [9] that also can reach the true optimal value, solves PSD+RLT+TRI (see details in [9]) relaxation of the BL problem to which a special second order conic (SOC) constraint is added. Although the method in [9] obtains the true optimal value for the BL problem, the solution does not immediately provide a feasible solution (since the obtained matrix is not necessarily of rank 1) and additional steps should be taken to obtain it.

5.7 Comparison of QMDCF with semidefinite relaxation

We generated few syntetic examples of general quadratic problem in the following form

$$\max_{x \in R^n} \{x^T A x | x^T A_i x \leq i = 1, 2\},$$

where A is an arbitrary symmetric matrix and A_1, A_2 are positive semidefinite matrices. The semidefinite relaxation of the above problem is obtained by introducing a matrix variable $X = xx^T$ and relaxing the demand $\text{rank}(X) = 1$:

$$\max\{\text{Tr}(AX) | \text{Tr}(A_i X) \leq i = 1, 2, X \succeq 0\}.$$

The goal was to compare the objective function values obtained by MDCF to the optimal value of semidefinite relaxation. The value of SDP relaxation is an upper bound for the optimal value of the problem in general case. In [3] was proved that in case of two inequality constraints where exactly one the the constraints is active, the value of SDP relaxation equals to the optimal value of the problem.

In six from seven examples QMDCF found values equal to upper bounds which means that QMDCF found the optimal values. In Ex.3 the upper bound differs from the value founded by QMDCF. Indeed, in this example two constraints are equalities in the optimal solutions and therefore SDP relaxation might differ from the optimal value. In Ex.8 with 300 variables, SDP program failed to supply solution. In other problems it can be seen from the table the significant grow of running time of SDP relaxation as the dimension of the problem grows. This phenomena does not occur in QMDCF method.

	n	Value (SDP)	time (SDP)	Value (MDCF)	time (MDCF)
Ex.1	3	1.0937	0.63 sec.	1.0937	12.6 sec.
Ex.2	10	0.8487	0.61 sec.	0.8487	3.88 min.
Ex.3	20	1.0955	1.62 sec.	1.0714	5.56 min.
Ex.4	50	0.3351	2.36 sec.	0.3351	21 sec.
Ex.5	100	0.4980	18.67 sec.	0.4980	1.4 min.
Ex.6	200	0.6665	12 min	0.6665	1.65 min.
Ex.7	250	0.8697	70 min.	0.8697	3 min.
Ex.8	300	out of memory	-	0.9469	7 min.

Table 6: MDCF versus SDP relaxation.

References

[1] Ben-Tal A, Teboulle M *Hidden convexity in some nonconvex quadratically constrained quadratic programming.* Mathematical Programming 72(1):51-63, 1996.

[2] Enhbat R *An algorithm for maximizing a convex function over a simple set.* Journal of Global Optimization 8(4):379-391, 1996

[3] Yinyu Ye, Shuzhong Zhang *New Results on Quadratic Minimization.* SIAM Journal on Optimization, July 2001.

[4] Selvi A, Ben-Tal A, Brekelmans R, Den Hertog D *Convex maximization via adjustable robust optimization.* INFORMS Journal on Computing, Vol. 34, No. 4, 2022.

[5] Jarre F *Optimal ellipsoidal approximations around the analytic center.* Applied Mathematics and Optimization 30(1):15-19,1994

[6] Ben-Tal A, Roos E *An algorithm for maximizing a convex function based on its minimum.* INFORMS journal on computing, 34, Number 6, November/December, 2022.

[7] Ben-Tal A, Den Hertog D *Hidden conic quadratic representation of some nonconvex quadratic optimization problems.* Mathematical Programming 143(1-2):1-29, 2014.

[8] S. Burer and A. N. Letchford *On nonconvex quadratic programming with box constraints.* SIAM J. Optim., 20(2):1073–1089, 2009.

[9] Kurt M. Anstreicher and Diane Puges *Extended Triangle Inequalities for Nonconvex Box-Constrained Quadratic Programming.* Arxiv, 2025.

[10] Philip Hartman, *On Functions Representable as a Difference of Convex Functions.* Pasific J. of Mathematics Vol.9, No. 3, July 1939.

[11] Immanuel M. Bomze and Gabriele Danninger, *A Finite Algorithm for Solving General Quadratic Problems.* Journal of Global Optimization 4: 1-16,1994.

[12] Immanuel M. Bomze and Etienne de Klerk, *Solving standard quadratic optimization problems via linear, semidefinite and copositive programming.* Journal of Global Optimization 24: 163–185, 2002.

[13] Jieqiu Chen and Samuel Burer, *Globally solving nonconvex quadratic programming problems via completely positive positive programming.* Math. Prog. Comp. 4: 33–52, 2012.

[14] Dieter Vandenbussche and George L. Nemhauser, *A branch-and-cut algorithm for nonconvex quadratic programs with box constraints.* Math. Program. Ser. A 102: 559–575, 2005.

[15] Julio Lopez, Sebastian Maldonado, Miguel Carrasco, *Double regularization methods for robust feature selection and SVM classification via DC programming.* Information Science 429: 337–389, 2018.

[16] Le Thi Hoai An and Pham Dinh Tao, *Solving a Class of Linearly Constrained Indefinite Quadratic Problems by D.C. Algorithms.* Journal of Global Optimization 11: 253-285,1997.

[17] Pham Dinh Tao and Le Thi Hoai An, *Difference of convex functions optimization algorithms (DCA) for globally minimizing nonconvex quadratic forms on Euclidean balls and spheres* . Operations Research Letters 19(1996) 207-216.

[18] Berkelaar, A.B., Jansen, B., Roos, C. and Terlaky, T, *Sensitivity Analysis in (Degenerate) Quadratic Programming* . Technical Report TWI 96-26, Reports of the Faculty of Technical Mathematics and Informatics, Delft University of Technology.

[19] Le Thi Hoai An and Pham Dinh Tao, *A Branch and Bound Methhod via d.c. Optimization Algorithms and Ellipsoidal Technique for Box Constrained Nonconvex Quadratic Problems.* Journal of Global Optimization 13: 171-206,1998.

[20] Le Thi Hoai An and Pham Dinh Tao, *The DC(Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems.* Annals of Operations Reasearch 133: 23-46,2005.

[21] Pham Dinh Tao and Le Thi Hoai An, *Convex Analysis Approach to D.C. Programming Theory, Algorithms and Applications.* ACTA MATHEMATICA VIETNAMICA Volume 22, Number 1. 1997, pp. 289-355.

[22] Pasquale L. De Angelis, Panos M. Pardalos, Gerardo Toraldo, *Quadratic Programming with Box Constraints.* M.Bomze et al.(eds.), Developments in Global Optimization, 73-93, 1997.