

## Research Article

# A Branch-and-Bound Algorithm Embedded with DCA for DC Programming

Meihua Wang,<sup>1</sup> Fengmin Xu,<sup>1</sup> and Chengxian Xu<sup>2</sup>

<sup>1</sup> Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, China

<sup>2</sup> SKLMSE Laboratory and Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, China

Correspondence should be addressed to Fengmin Xu, fengminxu@mail.xjtu.edu.cn

Received 10 October 2011; Revised 16 March 2012; Accepted 17 March 2012

Academic Editor: Wanquan Liu

Copyright © 2012 Meihua Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The special importance of Difference of Convex (DC) functions programming has been recognized in recent studies on nonconvex optimization problems. In this work, a class of DC programming derived from the portfolio selection problems is studied. The most popular method applied to solve the problem is the Branch-and-Bound (B&B) algorithm. However, "the curse of dimensionality" will affect the performance of the B&B algorithm. DC Algorithm (DCA) is an efficient method to get a local optimal solution. It has been applied to many practical problems, especially for large-scale problems. A B&B-DCA algorithm is proposed by embedding DCA into the B&B algorithms, the new algorithm improves the computational performance and obtains a global optimal solution. Computational results show that the proposed B&B-DCA algorithm has the superiority of the branch number and computational time than general B&B. The nice features of DCA (inexpensiveness, reliability, robustness, globality of computed solutions, etc.) provide crucial support to the combined B&B-DCA for accelerating the convergence of B&B.

## 1. Introduction

DC programming is an important subject in optimization problems. This paper studies one class of DC programming, which is originally derived from the portfolio investment problems.

Consider the following problem:

$$\begin{aligned} \min_x \quad & f(x) = p(x) + \phi(x) \\ \text{s.t.} \quad & x \in \mathcal{U} \\ & l \leq x \leq u. \end{aligned} \tag{Q}$$

here  $p(x) = (1/2)x^T Hx + c^T x$ , with  $H$  being a positive definite matrix.  $\phi(x) = \sum_{i=1}^n \phi_i(x_i)$ , with  $\phi_i(x_i)$  being a concave function for  $i = 1, 2, \dots, n$ . Decision vector  $x = (x_1, x_2, \dots, x_n)^T \in \mathcal{U}$ , with  $\mathcal{U} = \{x \mid Ax \leq b\}$  being a polyhedron,  $A \in R^{m \times n}$ ,  $b \in R^m$ . In addition,  $x$  is restricted to the lower bounds  $l = (l_1, l_2, \dots, l_n)^T$  and the upper bounds  $u = (u_1, u_2, \dots, u_n)^T$ .

Falk and Soland propose a B&B algorithm for separable nonconvex programming problems in [1], where the objective function is a separable nonconvex function. Phong et al. give a decomposition B&B method for globally solving linearly constrained indefinite quadratic minimization problems in [2, 3], where the objective function is  $p(x) + \phi(y)$  with  $p(x)$  being a convex quadratic function, however, the concave part  $\phi(y)$  is a function of  $y$  rather than  $x$ . Konno and Wijayanayake [4] propose a B&B algorithm to solve portfolio optimization problems under concave transaction costs. The algorithm is proposed by introducing linear underestimated functions for concave transaction cost functions, and is successively used to solve optimal portfolio selection problems with 200 assets. Honggang and Chengxian give a B&B algorithm to solve this class of DC programming with the proposed largest distance bisection in [5], and tests show the efficiency of the method for the problem with 20–160 dimensions. More representatively, a convex minorant of the DC function  $f(x) = p(x) + \phi(x)$  is defined by  $p + \text{co}_c(\phi)$  for B&B algorithm in DC programming, where  $\text{co}_c(\phi)$  denotes the convex envelope of the concave function  $\phi$  on the set  $C$ , and this is called DC relaxation in DC programming which has been completely studied in [6] and is important for nonconvex programming problems. The performance of the B&B depends on the branching strategy and bounding technology. The main concern of the above B&B algorithms is to solve the underestimated problem of initial problem to obtain an upper bound and also a lower bound for the optimal value, then divide the problem into two subproblems according to some rules, and repeat the above steps for a selected subproblem. By constantly reducing the upper bound and increasing the lower bound for the optimal value, we can obtain a global optimal solution. The main contribution of our work is to improve the upper bound for the optimal value by a local optimization algorithm for the DC programming self rather than the underestimated problem.

DCA is an effective local optimization method based on local optimality and the duality for solving DC programming, especially for large-scale problems. DCA has been first proposed by Tao and An [7] in their preliminary form in 1985 as an extension of subgradient algorithms to DC programming. Then, the method becomes classic and increasingly popular due to the joint work by An and Tao since 1994. Crucial developments and improvements for DCA from both theoretical and computational aspects have been completed, see [7–12] and references therein. In particular, the work by An et al. [12] investigates DC Programming and DCA for solving large-scale (until 400 000 dimensions) nonconvex quadratic programming. Since solving large-scale convex quadratic programs is expensive, the combination of DCA and interior point method (IP) can reduce the computation time and this outperforms the reference code LOQO by Vanderbei (Princeton). DCA has been widely used to a lot of different and nonconvex optimization problems, such as trust region subproblem, nonconvex quadratic programming, and so on (see [6, 13–17]). Furthermore, it always provides a global optimal solution for the problems [11]. The very complete reference including a list of real-world applications of DC Programming and DCA has been summarized by An and Tao (see the website [18]).

DCA is an efficient method for DC programming which allows to solve large-scale DC programming. In this paper, we will obtain a local optimization solution by the DCA method, the optimal value of which is also an upper bound for the optimal value of the problem (Q). In most cases, the local optimal solution is always a global optimal solution. Embedding this

upper bound into the B&B algorithms can improve the convergence speed and guarantee the global optimality of the solution when it is used to solve the problem (Q). Computational tests will be conducted for general problem and especially for portfolio selection problem, the results show that the proposed B&B-DCA algorithm can solve problems efficiently and have the superiority of the branch number and computational time.

The rest of the paper is organized as follows. Local optimization method DCA for the problem (Q) is described in Section 2. The B&B method embedded with DCA algorithm is given in Section 3. Computational tests are stated in Section 4. Conclusion and future research are shown in Section 5.

## 2. Local Optimization Method DCA

### 2.1. DCA for General DC Programming

Consider the following general DC programming:

$$\gamma = \inf\{F(x) = g(x) - h(x), x \in \mathbb{R}^n\}, \quad (P)$$

where  $g(x)$  and  $h(x)$  are lower semicontinuous proper convex functions on  $\mathbb{R}^n$ . Such a function  $F(x)$  is called a DC function, and  $g(x) - h(x)$  is called a decomposition of  $F(x)$ , while  $g(x)$  and  $h(x)$  are DC components of  $F(x)$ . In addition, a constrained DC programming whose feasible solution set  $\mathbb{C}$  is convex can also be transformed into an unconstrained DC programming by adding the indicator function of  $\mathbb{C}$  (it is equal to 0 in  $\mathbb{C}$ , infinity elsewhere) to the first DC component  $g(x)$ .

Let  $g^*(y) = \sup\{\langle x, y \rangle - g(x), x \in \mathbb{R}^n\}$  be the conjugate function of  $g(x)$ . Then, the dual programming of (P) can be expressed as

$$\gamma_D = \inf\{h^*(y) - g^*(y), y \in \mathbb{R}^n\}. \quad (P_D)$$

The perfect symmetry exists between primal programming (P) and dual programming (P<sub>D</sub>): the dual of (P<sub>D</sub>) is exactly (P). Remark that if the optimal value  $\gamma$  is finite, we have

$$\text{dom}(g) \subset \text{dom}(h), \quad \text{dom}(h^*) \subset \text{dom}(g^*). \quad (2.1)$$

here,  $\text{dom}(g) = \{x \in \mathbb{R}^n \mid g(x) < +\infty\}$ . Such inclusion will be assumed throughout the paper.

The necessary local optimality condition [6] for the primal problem (P) is

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset \quad (2.2)$$

(such a point  $x^*$  is called a critical point for  $g(x) - h(x)$ ) and

$$\partial h(x^*) \subset \partial g(x^*), \quad (2.3)$$

where subdifferential [19] of  $h(x)$  at  $x_0$  is denoted by

$$\partial h(x_0) = \{y \in \mathbb{R}^n \mid h(x) \geq h(x_0) + \langle x - x_0, y \rangle, \forall x \in \mathbb{R}^n\}. \quad (2.4)$$

Let  $\mathcal{P}$  and  $\mathcal{D}$  denote global solutions sets of problem  $(P)$  and  $(P_D)$ , respectively. According to the work by Toland [20], the relationship between them is

$$\left[ \bigcup_{y^* \in \mathcal{D}} \partial g^*(y^*) \right] \subset \mathcal{P}, \quad \left[ \bigcup_{x^* \in \mathcal{P}} \partial h(x^*) \right] \subset \mathcal{D}. \quad (2.5)$$

Under technical conditions, this transportation also holds true for local optimal solutions of problems  $(P)$  and  $(P_D)$ , more details can be found in [8–11].

Based on local optimality conditions and duality in DC programming, the DCA consists in the construction of two sequence  $\{x^k\}$  and  $\{y^k\}$  [17], such that the sequences  $\{g(x^k) - h(x^k)\}$  and  $\{h^*(y^k) - g^*(y^k)\}$  are decreasing, and  $\{x^k\}$  (resp.,  $\{y^k\}$ ) converges to a primal feasible solution  $\tilde{x}$  (resp., a dual feasible solution  $\tilde{y}$ ) which satisfies local optimality conditions and

$$\tilde{x} \in \partial g^*(\tilde{y}), \quad \tilde{y} \in \partial h^*(\tilde{x}). \quad (2.6)$$

Then, the basic scheme of DCA can be expressed as follows:

$$y^{k+1} \in \partial h(x^k), \quad x^{k+1} \in \partial g^*(y^k). \quad (2.7)$$

In other words,  $x^{k+1}$  and  $y^{k+1}$  are the solution of the convex programming  $(P_k)$  and  $(P_{Dk})$ , respectively,

$$\inf \left\{ g(x) - \left[ h(x^k) + \langle x - x^k, y^k \rangle \right], x \in \mathbb{R}^n \right\}, \quad (P_k)$$

$$\inf \left\{ h^*(y) - \left[ g^*(y^k) + \langle y - y^k, x^{k+1} \rangle \right], y \in \mathbb{R}^n \right\}. \quad (P_{Dk})$$

In the following, we will show main convergence properties of DCA which have been proposed and proven in [10, 11, 17]. First,  $\mathbb{C}$  (resp.,  $\mathbb{D}$ ) is used to denote a convex set containing the sequence  $\{x^k\}$  (resp.,  $\{y^k\}$ ) and  $\rho(g, \mathbb{C})$  (or  $\rho(g)$  if  $\mathbb{C} = \mathbb{R}^n$ ) the modulus of strong convexity of the function  $g(x)$  on  $\mathbb{C}$  are given by:

$$\rho(g, \mathbb{C}) = \sup \left\{ \rho \geq 0 : g(x) - \left( \frac{\rho}{2} \right) \|x\|^2 \text{ is convex}, x \in \mathbb{C} \right\}. \quad (2.8)$$

(1) The sequences  $\{g(x^k) - h(x^k)\}$  and  $\{h^*(y^k) - g^*(y^k)\}$  are decreasing and.

- (i)  $g(x^{k+1}) - h(x^{k+1}) = g(x^k) - h(x^k)$  if  $y^k \in \partial g(x^k) \cap \partial h(x^k)$ ,  $y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1})$  and  $[\rho(g, \mathbb{C}) + \rho(h, \mathbb{C})] \|x^{k+1} - x^k\| = 0$ . Furthermore, if  $g$  or  $h$  are strictly convex on  $\mathbb{C}$ ,  $x^{k+1} = x^k$ .

- (ii)  $h^*(y^{k+1}) - g^*(y^{k+1}) = h^*(y^k) - g^*(y^k)$  if  $x^{k+1} \in \partial g^*(y^k) \cap \partial h^*(y^k)$ ,  $x^{k+1} \in \partial g^*(y^{k+1}) \cap \partial h^*(y^{k+1})$  and  $[\rho(g^*, \mathbb{D}) + \rho(h^*, \mathbb{D})] \|y^{k+1} - y^k\| = 0$ . Furthermore, if  $g^*$  or  $h^*$  are strictly convex on  $\mathbb{D}$ , then  $y^k = y^{k+1}$ .

In such cases, DCA terminates at the  $k$ th iteration (finite convergence of DCA).

- (2) If  $\rho(g, \mathbb{C}) + \rho(h, \mathbb{C}) > 0$  (resp.,  $\rho(g^*, \mathbb{D}) + \rho(h^*, \mathbb{D}) > 0$ ), then the series  $\{\|x^{k+1} - x^k\|^2\}$  (resp.,  $\{\|y^{k+1} - y^k\|^2\}$ ) converges.
- (3) If the optimal value  $\gamma$  of the primal problem (P) is finite and the infinite sequence  $x^k$  and  $y^k$  are bounded, then every limit point  $\tilde{x}$  (resp.,  $\tilde{y}$ ) of the sequence  $x^k$  (resp.,  $y^k$ ) is a critical point of  $g(x) - h(x)$  (resp.,  $h^*(y) - g^*(y)$ ).
- (4) DCA has a linear convergence rate for general DC programming.

## 2.2. DCA Applied for Solving Problem (Q)

Problem (Q) is a special form of general DC programming (P), with  $g(x) = p(x)$  and  $h(x) = -\phi(x)$ . According to the description of DCA in Section 2.1, we need to compute  $\partial(-\phi)(x)$  and  $\partial p^*(y)$ . According to knowledge of modern convex analysis, we have  $\partial(-\phi)(x) = \prod_{i=1}^n \partial(-\phi_i)(x_i)$ .

As can be seen, if and only if  $(-\phi)(x)$  is differential at  $x$ ,  $\partial(-\phi)(x)$  reduces to a singleton which is exactly  $\{\nabla(-\phi)(x) = ((-\phi_1)'(x_1), \dots, (-\phi_n)'(x_n))^T\}$ . For the computation of  $\partial p^*(y)$ , we need to solve the following convex quadratic programming:

$$\min_x \{p(x) - \langle x, y \rangle, x \in \mathcal{U} \& l \leq x \leq u\} \quad (2.9)$$

because its solution is exactly  $\partial p^*(y) = \{\nabla p^*(y)\}$ . Finally, DCA applied for solving problem (Q) can be described as follows.

*Algorithm 2.1* (The DCA for the Problem (Q)).

### 1° Initialization

Let  $\epsilon$  be a sufficiently small positive number. First select an initial point  $x^0 \in \mathbb{R}^n$ . Set  $t = 0$ , goto 2°.

### 2° Iteration

Set  $y^t = \nabla(-\phi)(x^t)$ , that is,  $y_i^t = \nabla(-\phi_i)(x_i^t)$ ,  $i = 1, 2, \dots, n$ , and then solve the following quadratic programming

$$\min_x \{p(x) - \langle x, y^t \rangle, x \in \mathcal{U} \& l \leq x \leq u\} \quad (2.10)$$

the solution of (2.10) is denoted as  $x^{t+1}$ , goto 3°.

### 3° Stop Criterion

If  $\|x^{t+1} - x^t\| \leq \epsilon$ , then stop and we get a local optimal solution  $x^* = x^{t+1}$ . Otherwise, set  $t = t + 1$ , goto 2°.

We can obtain a local optimal solution by Algorithm 2.1 efficiently for the problem (Q) with different dimensions.

## 3. B&B Algorithm Embedded with DCA Methods

In most cases, B&B [5] methods are used to obtain the global optimal solution of the problem (Q). The main concern of the existing B&B algorithms is to solve the underestimated problem of the problem (Q). However, the computational cost of the algorithm will be very large along with the increasing dimension of the problem. In this section, we improve the upper bound for the optimal value by the local optimization algorithm DCA for DC programming self rather than the underestimated problem. DCA will be embedded into the B&B algorithm to accelerate the convergence of B&B.

### 3.1. The Description of the B&B-DCA Algorithm

In this subsection, we present the B&B-DCA method for the problem (Q). Let  $S_0 = \{l \leq x \leq u\}$  be the initial set which needs to be branched. We replace each concave function  $\phi_i(x_i)$  in  $\phi(x)$  by its underestimated function  $\underline{\phi}_i(x_i)$  over the set  $S_0$ :

$$\underline{\phi}_i(x_i) = a_i + b_i x_i, \quad (3.1)$$

where

$$b_i = \frac{\phi_i(u_i) - \phi_i(l_i)}{u_i - l_i}, \quad a_i = \phi_i(l_i) - b_i l_i, \quad i = 1, 2, \dots, n. \quad (3.2)$$

Then, we let

$$\underline{f}(x) = p(x) + \underline{\phi}(x) = p(x) + \sum_{i=1}^n \underline{\phi}_i(x_i) \quad (3.3)$$

be the underestimated function of the objective  $f(x) = p(x) + \phi(x)$  in problem (Q). We solve the following quadratic programming problem:

$$\begin{aligned} \min_x \quad & p(x) + \underline{\phi}(x) \\ \text{s.t.} \quad & x \in U \\ & l \leq x \leq u. \end{aligned} \quad (Q)$$

Let  $\underline{x}^0$  be an optimal solution of the problem  $(\underline{Q})$ , then we get an upper bound  $f(\underline{x}^0)$  and a lower bound  $f(\underline{x}^0)$  for the optimal value  $f(x^*)$  ( $x^*$  is a global optimal solution) of the primal problem  $(Q)$ .

Then, Algorithm 2.1 is used to obtain a local optimal solution for which  $\underline{x}^0$  is set as the initial iteration point  $x^0$ . The relative optimal solution when Algorithm 2.1 stops are noted as  $\tilde{x}^0$ . Then, we set the upper bound for the optimal value  $f(x^*)$  (where  $x^*$  is a global optimal solution of the problem  $(Q)$ )  $\alpha^0 = \min\{f(\underline{x}^0), f(\tilde{x}^0)\}$ .

**Theorem 3.1.** *Let  $\underline{x}^0$  be an optimal solution of the problem  $(\underline{Q})$ , let  $\tilde{x}^0$  be a local optimal solution obtained by DCA method, and let  $x^*$  be a global optimal solution of the problem  $(Q)$ , then we have*

$$f(\underline{x}^0) \leq f(x^*) \leq \alpha^0. \quad (3.4)$$

where  $\alpha^0 = \min\{f(\underline{x}^0), f(\tilde{x}^0)\}$ .

*Proof.* The following relationship holds true:

$$\begin{aligned} f(\underline{x}^0) &= \min_x \{f(x) \mid x \in \mathcal{U} \cap \mathcal{S}_0\} \\ &\leq \min_x \{f(x) \mid x \in \mathcal{U} \cap \mathcal{S}_0\} \\ &= f(x^*) \\ &\leq \alpha^0 \\ &= \min\{f(\underline{x}^0), f(\tilde{x}^0)\}. \end{aligned} \quad (3.5)$$

This gives the conclusion. □

Before continuing to describe the algorithm, we need to know the ‘‘Rectangle Subdivision Process’’, that is, divide the set  $\mathcal{S}_0$  into a sequence of subsets  $\mathcal{S}_k$  by means of hyperplanes parallel to certain facets [5]. The family of subrectangles can be represented by a tree with root  $\mathcal{S}_0$  and subnodes. A node is a successor of another one if and only if it represents an element of the latter node. An infinite path in the tree corresponds to an infinite nested sequence of rectangles  $\mathcal{S}_k$ ,  $k = 0, 1, \dots$ . ‘‘Rectangle Subdivision Process’’ plays an important role in the B&B method. In order to ensure the convergence of the algorithm, the concept of ‘‘Normal Rectangular Subdivision’’ (NRS) has been introduced in [21].

*Definition 3.2* (see [21]). Assumed that  $\underline{\phi}^k(x) = \sum_{i=1}^n \underline{\phi}_i^k(x_i)$  is the linear underestimated function of  $\phi(x)$  over the set  $\mathcal{S}_k$ ,  $\underline{x}^k$  is the optimal solution of the underestimated problem of  $(Q)$ , then a nested sequence  $\mathcal{S}_k$  is said to be normal if

$$\lim_{k \rightarrow \infty} \left| \phi^k(\underline{x}^k) - \underline{\phi}^k(\underline{x}^k) \right| = 0. \quad (3.6)$$

A rectangular subdivision process is said to be normal if any nested sequence of rectangles generated from the process is normal.

If  $\alpha^0 - \underline{f}(\underline{x}^0) \leq \delta$  with  $\delta$  a given sufficient small number, then  $\underline{x}^0$  (when  $f(\underline{x}^0) \leq f(\tilde{x}^0)$ ) or  $\tilde{x}^0$  (when  $f(\underline{x}^0) > f(\tilde{x}^0)$ ) is an  $\delta$ -approximate global optimal solution of the problem (Q). Otherwise, the problem (Q) will be divided into two subproblems:

$$\min\{f(x) = p(x) + \phi(x), x \in \mathcal{U} \& x \in \mathcal{S}_1\}, \quad (Q_1)$$

$$\min\{f(x) = p(x) + \phi(x), x \in \mathcal{U} \& x \in \mathcal{S}_2\}, \quad (Q_2)$$

where

$$\begin{aligned} \mathcal{S}_1 &= \{x \mid l_s \leq x_s \leq h_s, l_i \leq x_i \leq u_i, i \neq s, i = 1, 2, \dots, n\}, \\ \mathcal{S}_2 &= \{x \mid h_s \leq x_s \leq u_s, l_i \leq x_i \leq u_i, i \neq s, i = 1, 2, \dots, n\} \end{aligned} \quad (3.7)$$

$h_s$  is decided by the NRS process such as  $\omega$ -subdivision.

Similar to the problem (Q), we can get the underestimated quadratic programming (noted as  $(\underline{Q}_1)$  and  $(\underline{Q}_2)$ ) for each of the subproblem  $(Q_1)$  and  $(Q_2)$  by replacing the concave part  $\phi(x)$  by their respective underestimated function  $\underline{\phi}^1(x)$  and  $\underline{\phi}^2(x)$ :

$$\min\{\underline{f}^1(x) = p(x) + \underline{\phi}^1(x), x \in \mathcal{U} \& x \in \mathcal{S}_1\}, \quad (\underline{Q}_1)$$

$$\min\{\underline{f}^2(x) = p(x) + \underline{\phi}^2(x), x \in \mathcal{U} \& x \in \mathcal{S}_2\}. \quad (\underline{Q}_2)$$

- (1) If either of them is infeasible, then the corresponding subproblem  $(Q_1)$  or  $(Q_2)$  is infeasible, and we will delete it.
- (2) If at least one subproblem is feasible, we can get the optimal solution  $\underline{x}^1$  (or  $\underline{x}^2$ ) of the underestimated subproblem  $(\underline{Q}_1)$  (or  $(\underline{Q}_2)$ ) for the subproblem  $(Q_1)$  (or  $(Q_2)$ ). Let upper bound  $\alpha^1 = \min\{\alpha^0, f(\underline{x}^1), f(\underline{x}^2)\}$ , then delete the subproblem  $(Q_1)$  or  $(Q_2)$  of which the lower bound  $\underline{f}(\underline{x}^1)$  or  $\underline{f}(\underline{x}^2)$  is larger than  $\alpha^1 - \delta$ .

Remarkably, if  $f(\underline{x}^1) < \alpha^0 - \delta$  or  $f(\underline{x}^2) < \alpha^0 - \delta$ , Algorithm 2.1 is used for solving the subproblem  $(Q_1)$  or  $(Q_2)$ . The corresponding optimal solution is noted as  $\tilde{x}^1$  or  $\tilde{x}^2$ . The upper bound for the optimal value will be updated,  $\alpha^1 = \min\{\alpha^1, f(\tilde{x}^1), f(\tilde{x}^2)\}$ .

We delete those subproblems of which the lower bound are larger than  $\alpha^1 - \delta$ . Then we select one subproblem from  $(Q_1)$  and  $(Q_2)$ , which has a smaller lower bound for optimal value, and divide it into two subproblems. Repeat this process until no subproblems exist.

In the following, we will give the detailed description of B&B-DCA algorithm.

*Algorithm 3.3* (The Combined B&B-DCA Algorithm).

### 1° Initialization

Set  $k = 0$ ,  $l^0 = l$ ,  $u^0 = u$ , give the tolerance  $\delta > 0$  a sufficient small number. Solve the underestimated problem  $(\underline{Q})$  to obtain an optimal solution  $\underline{x}^0$ . Then use Algorithm 2.1 (DCA)



to solve the problem  $(Q)$ , the resulting optimal solution is noted as  $\tilde{x}^0$ . Set problems set  $\mathbb{M} = \{Q_0 \triangleq Q\}$ , upper bound  $\alpha^0 = \min\{f(\underline{x}^0), f(\tilde{x}^0)\}$ , lower bound  $\beta(Q_0) = \underline{f}(\underline{x}^0)$ .  $x^0 = \operatorname{argmin}\{\alpha^0\}$ .

### 2° Stop Criterion

Delete all  $Q_i \in \mathbb{M}$  with  $\beta(Q_i) > \alpha^k - \delta$ . Let  $\mathbb{M}$  be the set of remained subproblems. If  $\mathbb{M} = \emptyset$ , stop, and  $x^k$  is an  $\delta$ -global optimal solution of the problem  $(Q)$ . Otherwise, goto 3°.

### 3° Branch

Select a problem  $(Q_j)$  from the set of problems  $\mathbb{M}$ :

$$\min\{f(x) = p(x) + \phi(x), x \in \mathcal{U} \& x \in S_j\} \quad (Q_j)$$

with

$$\beta^k = \beta(Q_j) = \min\{\beta(Q_t), Q_t \in \mathbb{M}\}. \quad (3.8)$$

Then divide  $S_j$  into  $S_{j,1}$  and  $S_{j,2}$  according to an NRS process, the relative subproblems are noted as  $(Q_{j,1})$  and  $(Q_{j,2})$ , set  $\mathbb{M} = \mathbb{M} \setminus (Q_j)$ .

### 4° Bound

For the subproblem  $(Q_{j,m})$ ,  $m = 1, 2$ , solve the underestimated subproblem of  $(Q_{j,m})$  to obtain the optimal solutions  $\underline{x}^{j,m}$ . Let  $\beta(Q_{j,m}) = \underline{f}^{j,m}(\underline{x}^{j,m})$ ,  $\alpha^{j,m} = f(\underline{x}^{j,m})$ . Then set  $\alpha^{k+1} = \min\{\alpha^k, \alpha^{j,1}, \alpha^{j,2}\}$  and  $x^{k+1} = \operatorname{argmin}\{\alpha^{k+1}\}$ .

### 5° Deciding Whether to Call DCA Procedure

For  $m = 1, 2$ . If  $\alpha^{j,m} < \alpha^k - \delta$ , Algorithm 2.1 (DCA) is applied to solve the subproblem of  $(Q_{j,m})$ , the resulting solution is denoted by  $\tilde{x}^{j,m}$ . Then set  $\alpha^{k+1} = \min\{\alpha^{k+1}, f(\tilde{x}^{j,m})\}$ , and  $x^{k+1} = \operatorname{argmin}\{\alpha^{k+1}\}$ , goto 6°; otherwise, goto 6°.

### 6° Iteration

Let  $\mathbb{M} = \mathbb{M} \cup \{Q_{j,1}, Q_{j,2}\}$ ,  $k = k + 1$  and goto 2°.

Since the DCA method is an efficient local optimization method for DC programming, the combination of DCA and B&B algorithm will guarantee the global optimality and accelerate the convergence of general B&B algorithm (see [5]) for the problem  $(Q)$ . Due to the decrease of upper bound  $\alpha$ , the convergence speed of the B&B algorithm will have some improvement. However, we do not need to implement the DCA in each subproblem. Only when some conditions are satisfied, the DCA procedure will be called so as to prevent from the overusing of DCA.

### 3.2. The Convergence of B&B-DCA Algorithm

**Theorem 3.4.** *The sequence  $\{x^k\}$  generated by the B&B-DCA algorithm converges to a global optimal solution of the problem (Q) as  $k \rightarrow \infty$ .*

*Proof.* If the algorithm terminates at finite iterations  $k$ ,  $x^k$  is a global optimal solution of the problem (Q) from the definition of  $\alpha^k$  and  $\beta^k$ .

If the algorithm does not stop at finite iterations, it must generate an infinite nested sequence  $S_{k_t}$  of rectangles. From the definitions of the upper bound  $\alpha^k$  and the lower bound  $\beta^k$ , we know that the sequence  $\{\alpha^k - \beta^k\}$  is nonincreasing and a nonnegative number. Since the rectangle bisection process satisfies an NRS process, we have the following expression:

$$\lim_{t \rightarrow \infty} \left| \phi^{k_t}(\underline{x}^{k_t}) - \underline{\phi}^{k_t}(\underline{x}^{k_t}) \right| = 0. \quad (3.9)$$

Obviously, this means that

$$\lim_{t \rightarrow \infty} \alpha^{k_t} - \beta^{k_t} = 0. \quad (3.10)$$

Then we have

$$\lim_{k \rightarrow \infty} \alpha^k - \beta^k = 0. \quad (3.11)$$

Furthermore,  $\beta^k \leq f(x^*) \leq \alpha^k$ , so the sequence generated by the algorithm above converges to a global optimal solution as  $k \rightarrow \infty$ .  $\square$

We can see that NRS process plays an important role in the convergence of the B&B-DCA algorithm

## 4. Computational Tests

In this section, we will test the performance of proposed B&B-DCA algorithm for randomly generated datasets and the results will be compared with that of general B&B algorithm (see [5]) for problem (Q). Specifically, portfolio selection problem with concave transaction costs is also studied in Section 4.2. All the computational tests are coded by MATLAB (CPLEX is integrated to solve the relative quadratic programming) and run on a personal computer with Pentium Pro Dual 2.66 GHZ and 2 GB memory.

### 4.1. Problems with Randomly Generated Datasets

Datasets with different dimensions will be generated to test the performance of the B&B-DCA and general B&B algorithms. We will conduct numerical experiments of the proposed algorithms with dimensions from 50 to 400 for the problem (Q). In the following, we will give the generating process of the data sets and values of some parameters.

For the objective function  $f(x) = p(x) + \phi(x)$  in the problem (Q), the separable logarithm function is used to specify the concave part  $\phi(x)$  because of its wide applications in economic and financial problems. Let

$$\phi(x) = \sum_{i=1}^n \phi_i(x_i) = \sum_{i=1}^n \ln(\theta_i x_i + \gamma_i), \quad (4.1)$$

where  $\theta_i$  and  $\gamma_i$  are randomly generated in regions  $[2, 3]$  and  $[3, 5]$ , respectively, in uniform distribution.

For the convex part  $p(x) = (1/2)x^T H x + c^T x$ , we generate  $n$  sequence  $q^j \in R^m (m > n)$ ,  $j = 1, 2, \dots, n$  in region  $[-1, 1]$  in uniform distribution, and the covariance matrix of  $q^j$  which is a positive definite matrix is noted as  $H$ . So, we can ensure that the function  $p(x)$  is convex. The coefficients  $c_i$  are randomly generated in regions  $[-1, 1]$ , respectively, in uniform distribution.

The feasible solution sets in the problem (Q) are the intersection of

$$\mathcal{U} = \left\{ x \mid \sum_{i=1}^n x_i = 1 \right\}, \quad (4.2)$$

$$l_i \leq x_i \leq u_i, \quad l_i = 0, \quad u_i = 1, \quad i = 1, 2, \dots, n. \quad (4.3)$$

The tolerance  $\epsilon$  and  $\delta$  in Algorithm 2.1 and Algorithm 3.3 are both set equal to  $1e - 5$ .

B&B-DCA and general B&B algorithms [5] are used to solve the problem (Q) on the same datasets and parameters from dimension 50 to 400. In order to make the results more reliable, we will generate randomly five datasets to test the proposed B&B-DCA and general B&B algorithms for each dimension. The stop criterion of the algorithms is either obtaining an  $\delta$ -optimal solution or the branch number greater than 20000.

The NRS process has an important effect on the convergence of the B&B-DCA and general B&B algorithms. To our knowledge, the NRS process includes the exhaustive bisection,  $\omega$ -bisection, adaptive bisection, and largest distance bisection [5]. Some evidences in [2, 5] show that the  $\omega$ -bisection and largest distance bisection are greater than the other two methods. Since the calculation of  $\omega$ -bisection is much more simple and have similar performance to the largest distance bisection,  $\omega$ -bisection is applied in our proposed B&B-DCA and general B&B algorithms for problem (Q).

*Definition 4.1* ( $\omega$ -subdivision [2, 14]). *With the  $\omega$ -subdivision process, the bisection index  $s$  is determined by*

$$\phi_s(\underline{x}_s^k) - \underline{\phi}_s(\underline{x}_s^k) = \max \left\{ \phi_i(\underline{x}_i^k) - \underline{\phi}_i(\underline{x}_i^k), i = 1, 2, \dots, n. \right\}, \quad (4.4)$$

where  $\underline{x}^k$  is the optimal solution of underestimated problem for  $(Q_k)$ . Then, bisection point  $h_s^k = \underline{x}_s^k$ .

**Table 1:** Computational results with the B&B, B&B-DCA and DCA, methods.

Dim	B&B		B&B-DCA			DCA	
	Avg Bran	Time(s)	Avg Bran	Time(s)	Num DCA	Time(s)	Num Glob
$n = 50$	56.4	7.15	47.6	6.29	1	0.46	5
$n = 100$	195.8	40.70	145.4	30.10	1	0.58	5
$n = 150$	309	83.00	174.2	47.35	1	0.70	5
$n = 200$	327.6	182.95	310.2	174.28	1	1.26	5
$n = 250$	486.4	382.98	399.6	315.46	1	2.02	5
$n = 300$	428.2	422.59	341.2	333.98	1	2.09	5
$n = 350$	951.8	1231.52	756	969.82	1.2	3.51	4
$n = 400$	988	1753.09	666.6	1183.18	1.2	4.28	4

### Results and Analysis

We find that optimal value computed by two algorithms are equal. In the following, we show the average branch number (Avg Bran) and mean CPU times (Time) for each dimension by the B&B-DCA and general B&B algorithms in Table 1. The average number of calling DCA process (Num DCA) in the B&B-DCA algorithm is also given. Furthermore, we test the performance of DCA applied to the problem ( $Q$ ). The number of times (Num glob) when a global optimal solution is obtained after one process of DCA is also given in Table 1. Average CPU times (Time) for one process of DCA method is also present.

From the results and Table 1, we have the following comments.

- (i) General B&B and the proposed B&B-DCA algorithms can efficiently solve the problem ( $Q$ ) from the dimension 50 to 400. Due to the nice features of DCA (inexpensiveness, reliability, robustness, globality of computed solutions, etc.), the proposed B&B-DCA algorithm shows great superiority than general B&B algorithm not only in the average branch number but also in mean CPU time. Take dimension  $n = 400$ , for example, the average branch number by general B&B is 988, however the one by B&B-DCA is only 666.6. Then, the relative mean CPU time has a decrease of nearly 600 seconds by the B&B-DCA algorithm. So, embedding DCA into the B&B algorithm is quite necessary.
- (ii) The DCA method always gives a good approximation for optimal solution of problem ( $Q$ ) within short CPU time. It can be seen that it gives a global optimal solution five times during five computational tests for the dimension from 50 to 300. Even so, embedding DCA into B&B algorithm is necessary to guarantee the global optimal solution. Furthermore, if we are able to give a new method to obtain a more tighter lower bound for the optimal value  $f(x^*)$  in the bound process, the proposed B&B-DCA algorithm can compute global optimal solution within a shorter time.

### 4.2. Portfolio Selection with Concave Transaction Costs

In this subsection, the proposed B&B-DCA and general B&B algorithms are applied to solve portfolio selection problem with concave transaction costs. It is pointed that concave

**Table 2:** Computational results for portfolio selection problem.

Dim	B&B		B&B-DCA			DCA	
	Avg Bran	Time(s)	Avg Bran	Time(s)	Num DCA	Time(s)	Num Glob
data 1: $n = 31$	72.95	18.03	64.10	16.10	2.00	0.29	14
data 2: $n = 85$	117.22	34.21	110.14	28.79	1.78	0.32	15
data 3: $n = 89$	194.95	40.70	175.36	32.86	1.42	0.25	17
data 4: $n = 98$	292.53	86.96	257.19	77.58	2.11	0.30	14
data 5: $n = 225$	113.11	76.77	100.34	72.55	1.16	0.62	18

transaction costs function is more reliable [4, 5]. The Mean-Variance (M-V) model can be written as [22]:

$$\begin{aligned}
 \min_x \quad & U(x) = \frac{\lambda}{2} x^T V x - (1 - \lambda) (R^T x - C(x)) \\
 \text{s.t.} \quad & e^T x = 1 \\
 & Ax \leq b \\
 & l \leq x \leq u.
 \end{aligned} \tag{PS}$$

Vector  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$  is the decision portfolio with  $x_i$  the investment weight in each asset.  $R = (\bar{r}_1, \dots, \bar{r}_n)^T \in \mathbb{R}^n$  denotes the expected return rate and  $V = (\sigma_{ij})_{n \times n} \in \mathbb{R}^{n \times n}$  is the covariance matrix of the return rate for assets. Then,  $x^T V x$  gives the risk (Variance) and  $R^T x - C(x)$  gives net return of the portfolio  $x$ , where  $C(x) = \sum_{i=1}^n C_i(x_i)$  denotes the nondecreasing concave transaction costs function, and  $C_i(x_i)$  is curved in Figure 1.

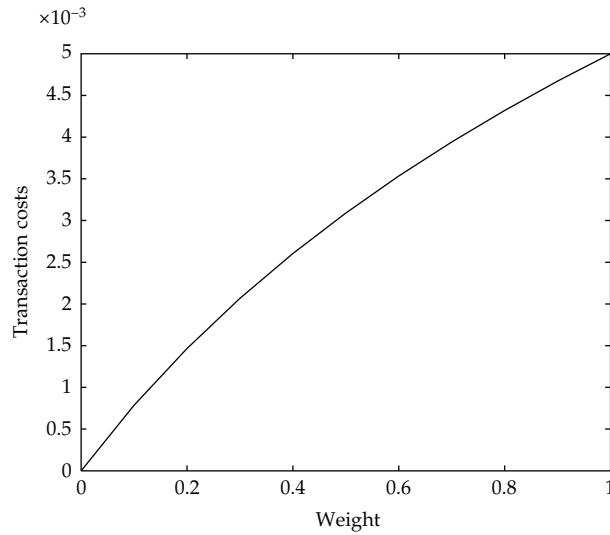
The sum of investment weight in each asset should be one, that is,  $e^T x = 1$ , where  $e \in \mathbb{R}^n$  denotes the vector with all entries equal to 1.  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $l$  and  $u$  are the limitations of lower bound and upper bound of the investment  $x$ . Parameter  $\lambda \in (0, 1)$  is the risk aversion index decided by the investor.

In general, the covariance matrix  $V$  is symmetric and positive definite. Then  $(\lambda/2)x^T V x$  and  $(1 - \lambda)(R^T x - C(x))$  are DC components of function  $U(x)$ . The proposed B&B-DCA and general B&B algorithms can be used to solve the problem (PS).

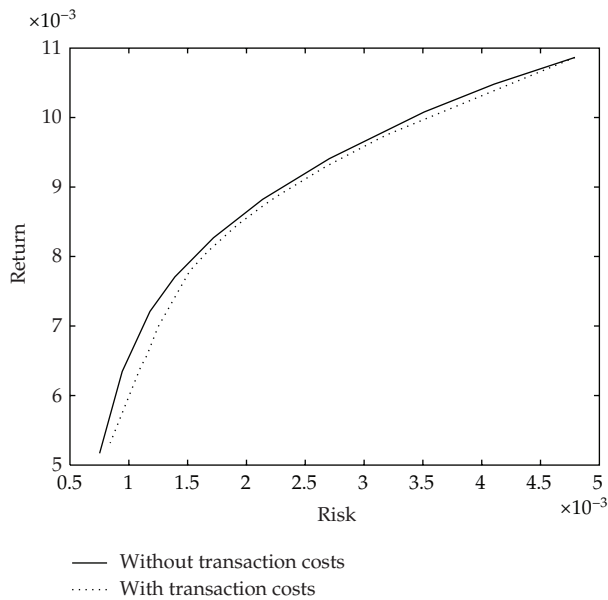
Tests will be performed on five datasets from the OR-Library (see [23]) which is a publicly available collection of test datasets for a variety of operations research problems. Each data set contains 290 weekly return rates for each stock. The data is computed from component stocks of Hang Seng, DAX, FTSE, S&P, and Nikkei Index for each dataset respectively. We can compute the relative expected return rate vector  $R$  and covariance matrix  $V$ . For each dataset, we will give the results for different value of  $\lambda$  from 0.05 to 0.95 (19 different values for  $\lambda$ ) in Table 2.

Similar to randomly generated datasets, we show average branch number (Avg Bran), average CPU time (Time), average number of calling DCA (Num DCA) for B&B and B&B-DCA, also total number (Num glob) when a global optimal solution is obtained after one process of DCA in Table 2.

As can be seen from Table 2, similar conclusions can be obtained to Table 1. First, the proposed B&B-DCA can accelerate the convergence of B&B to some extent in the aspects of branch number and CPU time. Second, DCA can compute a global optimal solution in



**Figure 1:** Concave transaction costs function  $C_i(x_i)$ .



**Figure 2:** Concave transaction costs function.

short time and most cases. However, B&B is needed to confirm the globality of computed solutions. When a global solution is found, the loose lower bound for optimal value cannot guarantee fast convergence of the B&B algorithm. How to obtain a well-defined lower bound is a challenging and practical study.

Additionally, Figure 2 presents the efficient frontiers generated from the the M-V portfolio models without transaction costs (transaction costs function  $C(x) = 0$ ) and with concave transaction costs ( $C(x)$  is a separable nondecreasing concave function). No

consideration of concave transaction costs will lead to inefficiently solutions. This will provide wrong guidance for the investors.

## 5. Conclusions and Future Research

In this paper, a class of DC programming is studied. General B&B is usually adopted to solve such problems. Based on existing local optimization method in DC programming, we have proposed a new global method B&B-DCA to solve the problem. DCA is an efficient local optimization method based on local optimality and the duality for solving DC programming, especially for large-scale problems.

Numerical tests on randomly generated datasets show that the proposed B&B-DCA has great superiority of branch number and computational time than general B&B algorithm with different dimensions. In addition, portfolio selection problem with transaction costs can be solved efficiently. The proposed B&B-DCA can be applied to solved other practical problems which can be modeled by this class of DC programming.

We find that DCA method always provides a global optimal solution, but the lower bound for the optimal value cannot guarantee fast convergence rate of B&B. If we can give a new method to obtain a more tighter lower bound, the proposed B&B-DCA algorithm can solve the problem with much shorter time. This seems significant in solving practical problems. Furthermore, other global optimization methods like filled function methods and so on can be combined with DCA to solve DC Programming. Some of these are under our current consideration.

## Acknowledgment

This work is supported by National Natural Science Foundations of China: 10971162, 11101325, and 71171158.

## References

- [1] J. E. Falk and R. M. Soland, "An algorithm for separable nonconvex programming problems," *Management Science*, vol. 15, pp. 550–569, 1969.
- [2] T. Q. Phong, L. T. H. An, and P. D. Tao, "Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems," *Operations Research Letters*, vol. 17, no. 5, pp. 215–220, 1995.
- [3] T. Q. Phong, L. T. H. An, and P. D. Tao, "On globally solving linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method," *Operations Research*, vol. 30, no. 1, pp. 31–49, 1996.
- [4] H. Konno and A. Wiyayanayake, "Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints," *Mathematical Programming B*, vol. 89, no. 2, pp. 233–250, 2001.
- [5] X. Honggang and X. Chengxian, "A branch and bound algorithm for solving a class of D-C programming," *Applied Mathematics and Computation*, vol. 165, no. 2, pp. 291–302, 2005.
- [6] P. D. Tao, N. Nguyen Canh, and L. T. H. An, "An efficient combined DCA and B&B using DC/SDP relaxation for globally solving binary quadratic programs," *Journal of Global Optimization*, vol. 48, no. 4, pp. 595–632, 2010.
- [7] P. D. Tao and L. T. H. An, "Convex analysis approach to d.c. programming: theory, algorithms and applications," *Acta Mathematica Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.
- [8] L. T. H. An, "Contribution à l'optimisation non convexe et à l'optimisation globale. Théorie, Algorithmes, Applications. Habilitation à Diriger des Recherches," Université de Rouen, 1997.

- [9] L. T. H. An, P. D. Tao, and M. Le Dung, "Exact penalty in d.c. programming," *Vietnam Journal of Mathematics*, vol. 27, no. 2, pp. 169–178, 1999.
- [10] L. T. H. An and P. D. Tao, "Large-scale molecular optimization from distance matrices by a d.c. optimization approach," *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 77–114, 2003.
- [11] L. T. H. An and P. D. Tao, "The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems," *Annals of Operations Research*, vol. 133, pp. 23–46, 2005.
- [12] P. D. Tao, L. T. H. An, and F. Akoa, "Combining DCA (DC algorithms) and interior point techniques for large-scale nonconvex quadratic programming," *Optimization Methods & Software*, vol. 23, no. 4, pp. 609–629, 2008.
- [13] P. D. Tao and L. T. H. An, "A d.c. optimization algorithm for solving the trust-region subproblem," *SIAM Journal on Optimization*, vol. 8, no. 2, pp. 476–505, 1998.
- [14] L. T. H. An and P. D. Tao, "A continuous approach for globally solving linearly constrained quadratic zero-one programming problems," *Optimization. A Journal of Mathematical Programming and Operations Research*, vol. 50, no. 1-2, pp. 93–120, 2001.
- [15] L. T. H. An, P. D. Tao, and N. V. Thoai, "Combination between global and local methods for solving an optimization problem over the efficient set," *European Journal of Operational Research*, vol. 142, no. 2, pp. 258–270, 2002.
- [16] L. T. H. An, M. Moeini, and T. P. Dinh, "DC programming approach for portfolio optimization under step increasing transaction costs," *Optimization. A Journal of Mathematical Programming and Operations Research*, vol. 58, no. 3, pp. 267–289, 2009.
- [17] L. T. H. An, M. Moeini, and P. D. Tao, "Portfolio selection under downside risk measures and cardinality constraints based on DC programming and DCA," *Computational Management Science*, vol. 6, no. 4, pp. 459–475, 2009.
- [18] DC Programming and DCA, "Local and Global Approaches—Theory, Algorithms and Applications," <http://lita.sciences.univ-metz.fr/~lethi/>.
- [19] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1970.
- [20] J. F. Toland, "On subdifferential calculus and duality in nonconvex optimization," *Société Mathématique de France. Bulletin. Mémoire*, no. 60, pp. 177–183, 1979.
- [21] R. Horst and H. Tuy, *Global optimization*, Springer-Verlag, Berlin, Second edition, 1993, Deterministic approaches.
- [22] H.-G. Xue, C.-X. Xu, and Z.-X. Feng, "Mean-variance portfolio optimal problem under concave transaction cost," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 1–12, 2006.
- [23] J. E. Beasley, N. Meade, and T.-J. Chang, "An evolutionary heuristic for the index tracking problem," *European Journal of Operational Research*, vol. 148, no. 3, pp. 621–643, 2003.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

