A NEW CONCAVE REFORMULATION AND ITS APPLICATION IN SOLVING DC PROGRAMMING GLOBALLY UNDER UNCERTAIN ENVIRONMENT

YANJUN WANG* AND KAIJI SHEN

School of Mathematics
Shanghai University of Finance and Economics
Shanghai 200433, China

(Communicated by Jie Sun)

Abstract. In this paper, a new concave reformulation is proposed on a convex hull of some given points. Based on its properties, we attempt to solve DC Programming problems globally under uncertain environment by using Robust optimization method and CVaR method. A global optimization algorithm is developed for the Robust counterpart and CVaR model with two kinds of special convex hulls: simplex set and box set. The global solution is obtained by solving a sequence of convex relaxation programming on the original constraint sets or divided subsets with branch and bound method. Finally, numerical experiments are given for DC programs under uncertain environment with two kinds of constraints: simplex and box sets. Simulation results show the feasibility and efficiency of the proposed global optimization algorithm.

1. Introduction. In this paper, we consider the following DC program:

\[
(DC) : \begin{cases} 
\min & q(x, \omega) = f(x, \omega) - g(x, \omega) \\
\text{s.t.} & x \in D = \text{conv}\{T_1, \cdots, T_m\}
\end{cases}
\]

where both \(f(x, \omega)\) and \(g(x, \omega)\) are convex on \(x\) on a convex hull \(D\) of some given points \(T_1, \cdots, T_m\), and \(\omega\) is a parameter. If the parameter are uncertain, this problem is a DC program under uncertain environment. As shown in Rockafellar [14], it is necessary to define some functions in order to formulate the problem under uncertain environment. Two functions, \(\text{Sup}\) and \(CVaR\), are chosen to treat the uncertainty, which correspond to Robust optimization method and CVaR method, respectively. CVaR (Condition Value at Risk) is an alternative risk measure for VaR. CVaR is the conditional expectation of the loss above VaR for the time horizon and the confidence level. Ben-Tal and Nemirovski [1, 2, 3] survey the result of Robust optimization in uncertain linear, conic quadratic and semi-definite programming.

2020 Mathematics Subject Classification. Primary: 65K05; Secondary: 90C90.
Key words and phrases. Concave reformulation, DC programming, global optimization, convex relaxation, branch and bound.

We gratefully acknowledge the valuable cooperation of Prof. R. Tyrrell Rockafellar (University of Washington). This research was supported by NSFC (11271243).

* Corresponding author: Yanjun Wang.
The previous works only considered the problems under uncertain environment, which are limited to the convex programming. In this paper, the nonconvex objective functions under uncertain environment will be considered. A new definition is proposed: Concave-form function of the function \( g(x, \omega) \), where \( g(x) \) is a function defined on a convex set \( D \). Although there are many Concave-form functions for the given function \( g(x, \omega) \) on \( D \), we present the smallest Concave-form function called Concave-Reformulation (Concave-R) function. A framework of an algorithm based on branch and bound methods is designed to solve the Robust counterpart and CVaR model globally by using the Concave-R function. The global solution is obtained by solving a sequence of convex relaxation programming on the original constraint sets or divided subsets.

The algorithms proposed in this paper can handle Robust DC model and CVaR DC model well in the case the discrete random parameters. If the random parameters are continuous, the Robust model can be solved on some certain assumptions. This paper also proves the connection of the optimal values between the robust counterpart and the CVaR model. When \( \alpha \to 1 \), the optimal value of CVaR model will converge to the optimal value of the Robust counterpart. Finally, the numerical experiment is given for a special DC program under uncertain environment with two kinds of constraints: simplex and box sets. The random parameters are assumed to be discrete. Simulation results show the feasibility and efficiency of the proposed global optimization algorithm.

The paper is organized as follows. In Section 2, a new definition of Concave-form is proposed, and the formulation of the smallest Concave-form is presented denoted as Concave-Reformulation (Concave-R) function. Section 3 discusses how to solve Robust counterpart and CVaR model and reveals how the optimal values between the robust counterpart and the CVaR model are connected. In section 4, a global optimization algorithm is developed based on the framework of branch and bound methods. The numerical experiments are given in Section 5.

2. Concave reformulation of a function on a convex hull. Let \( D \) be the convex hull of some given points \( T_1, T_2, \cdots, T_m \), and \( g(x, \omega) \) is a function on \( x \) defined on the set \( D \) with a parameter vector \( \omega \). In this section we propose a definition: concave-form function of the function \( g(x, \omega_0) \) on \( D \), with the given parameter \( \omega_0 \).

**Definition 2.1.** (Concave-form function) Let \( D \) be the convex hull of some given points \( T_1, T_2, \cdots, T_m \) where \( T_i \in \mathbb{R}^n \) \((i = 1, \cdots, m)\). For the given parameter \( \omega_0 \), a function \( h(x, \omega_0) \) is called a concave-form of the function \( g(x, \omega_0) \) on \( D \), if

\[
h(x, \omega_0) = \max \sum_{i=1}^{m} \lambda_i g(T_i, \omega_0) \quad \text{s.t.} \quad \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, \quad x = \sum_{i=1}^{m} \lambda_i T_i
\]

Although there are many Concave-form functions for any given function \( g(x, \omega_0) \) on \( D \), the one we are interested in is the smallest Concave-form function \( h(x, \omega_0) \) which satisfies \( h(x, \omega_0) \leq h'(x, \omega_0) \) for any Concave-form function \( h'(x, \omega_0) \) of \( g(x, \omega_0) \) on \( D \). We call the smallest Concave-form function as Concave-Reformulation (Concave-R) function. We present a form of Concave-form function \( h(x, \omega_0) \) in the following Lemma, and it will be shown that this function is the Concave-R of \( g(x, \omega_0) \), i.e. the smallest Concave-form.

**Lemma 2.2.** The following function \( h(x, \omega_0) \)

\[
h(x, \omega_0) = \max \sum_{i=1}^{m} \lambda_i g(T_i, \omega_0) \quad \text{s.t.} \quad \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, \quad x = \sum_{i=1}^{m} \lambda_i T_i
\]
is the Concave-R of the function $g(x, \omega_0)$ defined on a convex hull $D$ of $m$ given points $T_i$, $i = 1, \cdots , m$, i.e. the smallest Concave-form function of $g(x, \omega_0)$ on $D$.

Proof. First, for a given parameter vector $\omega_0$, we will show that the function $h(x, \omega_0)$ is a Concave-form. Since the inequalities $h(T_i, \omega_0) \geq g(T_i, \omega_0)$ $(i = 1, \cdots , m)$ can be seen from the expression of $h(x, \omega_0)$, then it suffices to show the concavity of $h(x, \omega_0)$. That is to say, for any points $x_1$ and $x_2$ belonging to $D$, and $\alpha \in [0, 1]$, it holds that

$$h(\alpha x_1 + (1 - \alpha) x_2, \omega_0) \geq \alpha h(x_1, \omega_0) + (1 - \alpha) h(x_2, \omega_0).$$  \hspace{1cm} (1)

It can be observed that the left side of (1) have the following form

$$h(\alpha x_1 + (1 - \alpha) x_2, \omega_0) = \max \lambda g(T_i, \omega_0)$$

s.t. $\sum_{i=1}^{m} \lambda_i = 1$, $\lambda_i \geq 0$.  \hspace{1cm} (2)

$$\alpha x_1 + (1 - \alpha) x_2 = \sum_{i=1}^{m} \lambda_i T_i.$$  

and the right side of (1) can be rewritten as

$$\alpha h(x_1, \omega_0) + (1 - \alpha) h(x_2, \omega_0) = \max \sum_{i=1}^{m} (\alpha \lambda_i^1 + (1 - \alpha) \lambda_i^2) g(T_i, \omega_0)$$

s.t. $\sum_{i=1}^{m} \lambda_i^1 = 1$, $\lambda_i^1 \geq 0$

$$\sum_{i=1}^{m} \lambda_i^2 = 1$, $\lambda_i^2 \geq 0$  \hspace{1cm} (3)

$$x_1 = \sum_{i=1}^{m} \lambda_i^1 T_i$$

$$x_2 = \sum_{i=1}^{m} \lambda_i^2 T_i.$$  

Furthermore, assume that the point $(\bar{\lambda}_1, \bar{\lambda}_2)$ is feasible for the optimization problem in (3), then $\bar{\lambda}_i = \alpha \lambda_i^1 + (1 - \alpha) \lambda_i^2$ must be feasible for the optimization problem in (2), and in this case we have

$$h(\alpha x_1 + (1 - \alpha) x_2, \omega_0) = \sum_{i=1}^{m} \bar{\lambda}_i g(T_i, \omega_0) = \sum_{i=1}^{m} (\alpha \bar{\lambda}_i^1 + (1 - \alpha) \bar{\lambda}_i^2) g(T_i, \omega_0)$$

$$= \alpha h(x_1, \omega_0) + (1 - \alpha) h(x_2, \omega_0).$$

So the inequality (1) holds, i.e. the concave property of the function $h(x, \omega_0)$ holds.

Second, it can be proved that $h(x, \omega_0)$ is the smallest Concave-form, i.e. Concave-R. Let $h'(x, \omega_0)$ be any Concave-form of $g(x, \omega_0)$, then from Definition 2.1, we have $h'(T_i, \omega_0) \geq g(T_i, \omega_0)$. For any $x \in D$, suppose that the value of $h(x, \omega_0)$ is attained at $\lambda_i$, then $x = \sum_{i=1}^{m} \lambda_i T_i$, and we have

$$h'(x, \omega_0) = h'(\sum_{i=1}^{m} \lambda_i T_i, \omega_0) \geq \sum_{i=1}^{m} \lambda_i h'(T_i, \omega_0)$$
\[ \sum_{i=1}^{m} \lambda_i g(T_i, \omega_0) = h(x, \omega_0). \]

It is shown that \( h(x, \omega_0) \) is the Concave-R function. \( \square \)

As we know in convex analysis, the convex hull of a nonconvex function \( g(x, \omega_0) \) denoted by \( \text{Conv} g(x, \omega_0) \) is defined as the largest convex function majorized by \( g(x, \omega_0) \). Similarly we have the definition of the concave hull of a nonconvex function \( g(x, \omega_0) \) denoted by \( \text{Conc} g(x, \omega_0) \). The concave hull of a non-convex function \( g(x, \omega_0) \) is the smallest concave function in all of concave functions which are greater than or equal to \( g(x, \omega_0) \). The following lemma of a concave hull of a nonconvex function can be similarly referred to the convex hull of a nonconvex function in [11].

**Lemma 2.3.** (Concave hull of a function) \( g(x, \omega_0) \) is defined on a closed convex set \( D \), \( \text{Conc} g(x, \omega_0) = \sup \{ \sum_{i=0}^{n} \lambda_i g(x_i, \omega_0) | \sum_{i=0}^{n} \lambda_i x_i = x, \lambda_i \geq 0, \sum_{i=0}^{n} \lambda_i = 1, x_i \in D \}. \)

What is the relationship between the Concave-R and the concave hull of a function? Firstly, both Concave-R and Concave hull are concave functions. Secondly, in terms of Concave-R function \( h(x, \omega_0) \), \( h(T_i, \omega_0) \geq g(T_i, \omega_0) \) can be ensured for each given point \( T_i \), while for any other points in \( D \), \( h(T_i, \omega_0) \geq g(T_i, \omega_0) \) cannot be ensured. For the concave hull \( \text{Conc} g(x, \omega_0) \) can be ensured \( \text{Conc} g(x, \omega_0) \geq g(x, \omega_0) \) for all points \( x \in D \). Finally, the Concave-R is highly relevant to the given points \( T_i \). In other words, given different points set \( \{T_1, \cdots, T_m\} \) would result in different Concave-R. However, there is no connection between the points whose convex hull is the definition region \( D \). Regarding the relationship between these two, we can say the concave hull must be a concave-form and \( \text{Conc} g(x, \omega_0) \geq h(x, \omega_0) \) is always tenable. It will be shown later that these two are equivalent under some special cases. Next the properties of the Concave-R functions will be discussed.

**Proposition 2.1.** If the vectors \( T_1, \cdots, T_m \in R^n \) are linearly independent, the Concave-R \( h(x, \omega_0) \) is an affine function on \( D \) where \( D = \text{Conc} \{T_1, \cdots, T_m\} \).

**Proof.** Let \( A = [T_1, \cdots, T_m]_{n \times m} \). Obviously \( A \) is column full rank, and \( A^T A \) is invertible.

Rewriting the constraint \( x_i = \sum_{i=1}^{m} \lambda_i T_i \) as \( A \lambda = x \), where \( \lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T \) and \( x = (x_1, x_2, \ldots, x_n)^T \), we have the following result,

\[ \lambda = (A^T A)^{-1} A^T x. \]

It shall be noted that \( x \) belongs to the convex hull of the points \( T_1, \cdots, T_m \) which are linearly independent, then there will exist unique \( \lambda \) satisfying \( \lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i = 1, A \lambda = x \).

So \( \lambda = (A^T A)^{-1} A^T x \) is the unique solution, and \( h(x) \) is an affine function \( h(x, \omega_0) = [(A^T A)^{-1} A^T x]^T g(T, \omega_0) = x^T A (A^T A)^{-1} g(T, \omega_0) \) where \( g(T, \omega_0) = [g(T_1, \omega_0), \cdots, g(T_m, \omega_0)]^T \).

\( \square \)

**Proposition 2.2.** If the vectors \( T_1, \cdots, T_{n+1} \in R^n \) generate a simplex \( D \) of dimension \( n \), then the Concave-R \( h(x, \omega_0) \) must be an affine function on \( D \).
Proof. Let $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^{n+1}$, and $A_{n \times (n+1)} = [T_1, \ldots, T_{n+1}]$. Then in the expression of $h(x, \omega_0)$, we can rewrite the constraints as follows

$$\lambda_i \geq 0, \left( \begin{array}{c} A \\ e^T \end{array} \right) \lambda = \left( \begin{array}{c} x \\ 1 \end{array} \right). \tag{4}$$

Since $T_1, \ldots, T_{n+1}$ generate a simplex and the dimension of simplex is $n$, $T_2 - T_1, \ldots, T_{n+1} - T_1$ are linearly independent. It can be seen that

$$\begin{bmatrix} T_1 & T_2 - T_1 & \cdots & T_{n+1} - T_1 \\
1 & 0 & 0 & 0 \end{bmatrix}$$

is full rank. Therefore, the matrix $\left( \begin{array}{c} A \\ e^T \end{array} \right)$ is full rank.

Since $x$ belongs to the simplex, there must exist at least a solution $\lambda$ satisfying (4). So the unique solution of (4) is:

$$\lambda = \left( \begin{array}{c} A \\ e^T \end{array} \right)^{-1} \left( \begin{array}{c} x \\ 1 \end{array} \right)$$

Then the function $h(x, \omega_0)$ has the following form:

$$h(x, \omega_0) = (x^T \ 1) \left( \begin{array}{c} A \\ e^T \end{array} \right)^{-T} g(T, \omega_0).$$

$\square$

**Proposition 2.3.** If $g(x, \omega_0)$ is a convex function defined on $D = \text{conv}\{T_1, \ldots, T_m\}$, then the Concave-R function $h(x, \omega_0)$ must satisfy

$$h(x, \omega_0) \geq g(x, \omega_0), \quad \forall x \in D.$$ 

In this case the Concave-R $h(x, \omega_0)$ and concave hull (Conc $g(x, \omega_0)$) have the same expression.

**Proof.** For any $x \in D$, assume that the value of $h(x, \omega_0)$ is attained at $x = \sum_{i=1}^m \bar{\lambda}_i T_i$ where $\sum_{i=1}^m \lambda_i = 1, \lambda_i \geq 0 (i = 1, \ldots, m)$, we have $h(x, \omega_0) = h(\sum_{i=1}^m \bar{\lambda}_i T_i, \omega_0) \geq \sum_{i=1}^m \bar{\lambda}_i g(T_i, \omega_0) \geq g(x, \omega_0)$. The last inequality follows from the convexity of the function $g(x, \omega_0)$. So the first assertion is true. For the second assertion, we will show it from two equalities. Since $(\text{Conc} g)(x, \omega_0)$ is a concave-form of $g(x, \omega_0)$, $(\text{Conc} g)(x, \omega_0) \geq h(x, \omega_0)$. From the first assertion and definition of $(\text{Conc} g)(x, \omega_0)$, it can be concluded that $(\text{Conc} g)(x, \omega_0) \leq h(x, \omega_0)$. So the second assertion is true. $\square$

**Proposition 2.4.** Let $g(x, \omega_0)$ be convex. Choose some points $\{T_{i_1}, \ldots, T_{i_m}\}$ from $\{T_1, \ldots, T_k\}$. Then the convex hull of $\{T_{i_1}, \ldots, T_{i_m}\}$ denoted by $D_k$ is a subset of the convex hull of the points $\{T_1, \ldots, T_k\}$ denoted by $D$. Let $h_{D_k}(x, \omega_0)$ and $h_D(x, \omega_0)$ be the smallest concave-R of $g(x, \omega_0)$ on $D_k$ and on $D$ respectively. It holds that $h_{D_k}(x, \omega_0) \leq h_D(x, \omega_0), \quad \forall x \in D_k$.

**Proof.** It can be easily seen from the definition of Concave-R in Lemma 2.2. $\square$

In the following context, we will pay attention to the Concave-R of composition function with special structure. Assume that $g(x, \omega_0)$ is a composition function of a linear function with the following form

$$g(x, \omega_0) = p(b^T x, \omega_0)$$
where \( b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n \) is given, the function \( p : \mathbb{R} \to \mathbb{R} \). According to Lemma 2.2, given points \( T_1, \ldots, T_m \) and their convex hull \( D \), the Concave-R \( h(x, \omega_0) \) has the following form

\[
    h(x, \omega_0) = \max \sum_{i=1}^m \lambda_i p(b^TT_i, \omega_0) \quad \text{s.t.} \quad \sum_{i=1}^m \lambda_i = 1, \lambda_i \geq 0, \quad x = \sum_{i=1}^m \lambda_i T_i
\]

Since \( p(b^Tx, \omega_0) \) is convex function, from Proposition 2.3, the Concave-R \( h(x, \omega_0) \) satisfies \( h(x, \omega_0) \geq p(b^Tx, \omega_0), \forall x \in D \). In other words, this Concave-R can be regarded as an overestimator function of \( p(b^Tx, \omega_0) \) on \( D \). Then we will show that the Concave-R is a better overestimator function than the function proposed by Floudas[8], Shen[15], and Wang[16] in case that the convex hull is a box set or a simplex.

(1) \( D \) is a box set. Without loss of generality, let the convex hull \( D \) of \( T_1, \ldots, T_m \) (\( m > 2^n \)) be a box set, then we can choose \( 2^n \) points whose convex hull is also \( D \). Denote these points by \( T_{n_1}, \ldots, T_{n_k} \) which are exactly the vertexes of the box set.

In [8, 15, 16], the authors use the following concave linear overestimator. Let

\[
    L = \min \{ b^T x \mid x \in \{ T_{n_1}, \ldots, T_{n_k} \} \}
\]

and

\[
    U = \max \{ b^T x \mid x \in \{ T_{n_1}, \ldots, T_{n_k} \} \}.
\]

Then a linear concave overestimator of \( g(x, \omega_0) \) on \( D \) denoted by \( l(x, \omega_0) \) can be formed as [8, 15, 16]:

\[
    l(x, \omega_0) = \frac{p(U, \omega_0) - p(L, \omega_0)}{U - L} (b^T x - L) + p(L, \omega_0). \tag{5}
\]

It can be seen from Proposition 2.3 that the Concave-R \( h(x, \omega_0) \) must satisfy \( p(b^T x, \omega_0) \leq h(x, \omega_0) \leq l(x, \omega_0), \forall x \in D \). So we can say that the Concave-R is a better overestimator function than \( l(x, \omega_0) \) developed by Floudas[8], Shen[15], and Wang[16] in some sense. This point can be illustrated by the following example. For a given \( \omega_0 \), let \( g(x, \omega_0) = \exp(2x_1 - x_2), D = \text{conv}\{T_1, T_2, T_3, T_4\} \) where

\[
    [T_1, T_2, T_3, T_4] = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}.
\]

The following concave linear overestimator can be derived from (5):

\[
    l(x, \omega_0) = \frac{e^2 - e^{-1}}{3} (2x_1 - x_2 + 1) + e^{-1}. \tag{6}
\]

We can also derive the Concave-R \( h(x, \omega_0) \) according to our method as follows

\[
    h(x, \omega_0) = \max \sum_{i=1}^4 \lambda_i g(T_i, \omega_0) \quad \text{s.t.} \quad \sum_{i=1}^4 \lambda_i = 1, \lambda_i \geq 0, \quad x = \sum_{i=1}^4 \lambda_i T_i
\]

It can be shown from Figure 1, that \( h(x, \omega_0) \leq l(x, \omega_0) \).

The function \( h(x, \omega_0) \) has no explicit expression except in some special case, while the concave linear overestimator \( l(x, \omega_0) \) has an accurate expression. The partial optimization procedure in \( h(x, \omega_0) \) can be easily combined with the whole optimization problem when we develop an optimization algorithm.
(2) D is a simplex set. By Proposition 2.2, the Concave-R has the following form:

\[ h(x, \omega_0) = \left( x^T \ 1 \right) \begin{pmatrix} A \\ e^T \end{pmatrix}^{-T} p(h^T T, \omega_0) . \]

Since the function \( h(x, \omega_0) \) is the smallest concave-form, so we still have \( h(x, \omega_0) \leq l(x, \omega_0) \) where \( l(x, \omega_0) \) is defined as (5).

Consider the above function example \( g(x, \omega_0) \) in case (1). Let the simplex be the convex hull of \( T_1, T_2, T_3 \), where

\[ [T_1, T_2, T_3] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} . \]

If we use (5) to compute, the linear concave overestimator we get is still (6) in simplex set. We can also get the Concave-R as follows

\[ h(x, \omega_0) = (1 - x_2)e^2 + (1 - x_1)e^{-1} + (x_1 + x_2 - 1)e . \]

It can be seen from Figure 2 that \( h(x, \omega_0) \leq l(x, \omega_0) \).
The Concave-R presented in this section can be regarded as the approximation of the original function. We can make use of this approximation to get some relaxation programming in developing a global optimal algorithm.

**Lemma 2.4.** Assume that we have a sequence of convex hulls \( D_k(k = 1, 2, \ldots) \), where \( D_k = \text{conv}\{T_{k1}, \ldots, T_{km}\} \), and \( D_1 \supseteq D_2 \supseteq D_3 \supseteq \cdots \supseteq D_k \cdots \). If the limitation set of this sequence \( D_k \) is a single point \( T_0 \), the error between the Concave-R \( h(x, \omega_0) \) and the original continuous function \( g(x, \omega_0) \) will converge to zero on convex hull \( D_k \) when \( k \to +\infty \).

**Proof.** The error between the Concave-R \( h(x, \omega_0) \) and the original function \( g(x, \omega_0) \) is

\[
\max_{x \in D_k} \left( h(x, \omega_0) - g(x, \omega_0) \right).
\]

Substituting the form of \( h(x, \omega_0) \) on \( D_k \) into (7), equivalently we have

\[
\begin{align*}
\max_{x, \lambda} \sum_{i=1}^{m} \lambda_i g(T_i, \omega_0) - g(x, \omega_0) \\
\text{s.t.} \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, x = \sum_{i=1}^{m} \lambda_i T_i,
\end{align*}
\]

Since the limitation set of this sequence \( D_k \) is a single point \( T_0 \), and the function \( g(x, \omega_0) \) is continuous on \( D \), then the error between \( h(x, \omega_0) \) on \( D_k \) (denoted as \( h_{D_k}(x, \omega_0) \)) and \( g(x, \omega_0) \) will converge to zero when \( k \to +\infty \). \( \square \)

3. **Two functionals for uncertain DC programming and applications of concave reformulation.** In this section, we will adopt two kinds of functionals to treat the uncertainty in (DC) problem. One is \( \text{Sup} \) that corresponds to Robust optimization model denoted as robust counterpart and the other is \( \text{CVaR} \) that corresponds to CVaR model. We will also prove the connection between the optimal values of the CVaR model and of the robust counterpart. That is to say, when \( \alpha \to 1 \), the optimal value of CVaR model will converge to the optimal value of the Robust counterpart.

### 3.1. Robust model.

By Robust Optimization [1], the Robust counterpart of DC programming problems under uncertain environment can be written as follows

\[(RDC) : \min_{x \in D = \text{conv}\{T_1, \ldots, T_m\}} \max_{\omega \in Y} (f(x, \omega) - g(x, \omega)) \]

where \( Y \) is the uncertainty set corresponding to the parameter \( \omega \). Based on the discussion in Section 2, we can construct a convex relaxation programming by replacing \( g(x, \omega) \) in Robust counterpart of the (DC) under uncertain environment with its Concave-R \( h_D(x, \omega) \) as follows:

\[(RO - CRP) : \min_{x \in D = \text{conv}\{T_1, \ldots, T_m\}} \max_{\omega \in Y} (f(x, \omega) - h_D(x, \omega)) \]

In (RO-CRP), if \( h_D(x, \omega) \) has an explicit formula beyond maximization (for example under some special cases as that in Proposition 2.1 and Proposition 2.2, \( h_D(x, \omega) \) needs to be replaced by the explicit formulation and obtain the convex relaxation programming. In general case, some discussion will be presented in Subsection 4.1.
3.2. The CVaR model. Let $\psi(x, \omega) = f(x, \omega) - g(x, \omega)$ be the continuous loss function, and $\psi(x, \omega) : R^n \times R^m \to R$. Let $D$ and $Y$ be closed subsets of $R^n$ and $R^m$ respectively. Let $\omega : \Omega \to Y \subset R^m$ be a vector of random variables defined on the probability space $(\Omega, \Sigma, P)$ with support set $Y$. We suppose that $\omega(\xi)$ is a continuous random variable with density function $\rho(y)$, and we treat $\psi(x, \omega)$ for $x \in D$ as a random variable induced by $\omega(\xi)$. For each fixed $x$, we denote by $\Psi(x, \cdot)$ on $R$ the resulting distribution function for the loss $\psi(x, \omega(\xi))$, i.e.,

$$\Psi(x, \beta) = P\{\psi(x, \omega) \leq \beta\}.$$ 

We consider a confidence level $\alpha \in (0, 1)$, like $\alpha = 0.95$ or $\alpha = 0.99$. Value-at-risk can be defined as follow:

**Definition 3.1. (VaR)** The $\alpha$-VaR of the loss associated with a decision $x$ is the value

$$VaR_{\alpha}(\psi(x, \omega)) = \inf\{\alpha | \Psi(x, \beta) \geq \alpha\}.$$ 

The situation $\Psi(x, \cdot)$ entails a discontinuity in the behavior of VaR: a jump is sure to occur if a slightly higher confidence level is demanded. Furthermore, although $x$ is fixed, we can find a situation that the misbehavior in the dependence of VaR on $\alpha$ can effect its dependence on $x$ as well. That makes it hard to cope successfully with VaR-centered problems of optimization in $x$.

This problem motivates the researchers to look for a better measure of risk than value-at-risk for practical applications. Such a measure is conditional value-at-risk.

**Definition 3.2. (CVaR)** The $\alpha$-CVaR of the loss associated with a decision $x$ is the value:

$$CVaR_{\alpha}(\psi(x, \omega)) = \text{mean of the } \alpha\text{-tail distribution of } z = \psi(x, \omega).$$

The consequences of this maneuver will be examined in relation to following variants in which the whole interval $[VaR_{\alpha}(\psi(x, \omega)), \infty)$ or its interior $(VaR_{\alpha}(\psi(x, \omega)), \infty)$ is the focus.

The CVaR model is considered for uncertain DC programs under environment

$$(CVaR - DC) = \min_{x \in D = \text{conv}\{T_1, \cdots, T_m\}} CVaR_{\alpha}(f(x, \omega) - g(x, \omega))$$

Based on the discussion about the Concave-R function $h(x, \omega_0)$ of $g(x, \omega_0)$ on $D$ in Section 2 and convexity preserving nature of CVaR, we can exactly construct a convex relaxation programming by replacing $g(x, \omega)$ in (CVaR-DC) with its Concave-R $h_D(x, \omega)$ as follows

$$(CVaR - CRP) : \min_{x \in D = \text{conv}\{T_1, \cdots, T_m\}} CVaR_{\alpha}(f(x, \omega) - h_D(x, \omega)).$$

In (CVaR-CRP), if $h_D(x, \omega)$ have an explicit formulation beyond maximization under some special cases as that in Proposition 2.1 and Proposition 2.2, then $h_D(x, \omega)$ is needed to be replaced by the explicit formulation to obtain the convex relaxation programming. In a general case where $h_D(x, \omega)$ has no an explicit formulation beyond maximization, if the random vectors are discrete, attractive methods in Subsection 4.1 can be applied to solve the above convex relaxation programming (CVaR-CRP).
3.3. The relationship between CVaR and Robust model. From [6], another property of conditional value-at-risk is shown as follows:

$$\lim_{\alpha \to 1} CVaR_\alpha (f(x, \omega) - g(x, \omega)) = \sup_{\omega \in Y} (f(x, \omega) - g(x, \omega)).$$

Then we have the following proposition.

**Proposition 3.1.** Let $D = \text{conv}\{T_1, \cdots, T_m\}$ and the support space of the distribution of random parameter $\omega$ (related to CVaR) is as same as the uncertain set of the uncertainty parameter $\omega$ (related to Robustness), according to the definitions of the CVaR and the Robust, the following formula holds:

$$\min_{x \in D} \lim_{\alpha \to 1} CVaR_\alpha (f(x, \omega) - g(x, \omega)) = \min_{x \in D} \sup_{\omega \in Y} (f(x, \omega) - g(x, \omega)).$$

When $\alpha \to 1$, the optimal value of CVaR model (CVaR-DC) will converge to the optimal value of the Robust counterpart (RDC).

4. Global optimization for Robust and CVaR model. The goal of the this section is to develop a solution procedure for (RDC) and (CVaR-DC) problem based on branch and bound methods. This algorithm needs to solve a sequence of convex relaxation programming over partitioned subsets of $D$ in order to find a global solution.

4.1. How to solve the two kinds of convex relaxation programming. For Robust model, we will discuss how to solve the convex relaxation problem (RO-CRP) in the following two cases.

**Case 1.** In order to make a comparison between the Robust Model and the CVaR model, parameter $\omega$ is selected from a discrete collection of vectors $\omega_1, \omega_2, \cdots, \omega_q$. (RO-CRP) can be formulated as the following form:

$$\begin{cases}
\min_t \\
\text{s.t. } f(x, \omega_k) - h_D(x, \omega_k) \leq t, \quad k = 1, \cdots, q.
\end{cases}$$

$x \in D = \text{conv}\{T_1, \cdots, T_m\}$

The above form is equivalent to the following form:

$$(RO - CRP) \begin{cases}
\min_t \\
\text{s.t. } f(x, \omega_k) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega_k) \leq t \\
\sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0,
\end{cases}$$

$x = \sum_{i=1}^{m} \lambda_i T_i$.

The above problem (RO-CRP) is a convex programming, so it can be solved efficiently.

**Case 2.** If parameter $\omega$ is selected from a infinity set $Y$, the corresponding (RO-CRP) approximation is:

$$\begin{cases}
\min_t \\
\text{s.t. } f(x, \omega) - h_D(x, \omega) \leq t, \quad \forall \omega \in Y.
\end{cases}$$

$x \in D = \text{conv}\{T_1, \cdots, T_m\}$

If the set $Y$ is restricted as a nonempty compact convex set, and the function $f(x, \omega)$ and $g(x, \omega)$ are continuous. If $f(x, \omega)$ is concave on $\omega$, and $g(x, \omega)$ is convex on $\omega$, we can still get an explicit form for (RO-CRP). The details is in the following proposition.
Proposition 4.1. Assume that \( Y \) is a nonempty compact convex set, and the function \( f(x, \omega) \) and \( g(x, \omega) \) are continuous. If \( f(x, \omega) \) is concave on \( \omega \), and \( g(x, \omega) \) is convex on \( \omega \) where \( \omega \in Y \), then the problem (RO-CRP) is equivalent to the following form (RO-CRP):

\[
\begin{align*}
& \min_{x, \lambda} \max_{\omega \in Y} f(x, \omega) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega) \\
& \quad \text{s.t.} \quad \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, x = \sum_{i=1}^{m} \lambda_i T_i.
\end{align*}
\]

Proof. For a given point \( x \in D \), let the set \( \Lambda_x \) be

\[
\Lambda_x = \{ \lambda | \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, x = \sum_{i=1}^{m} \lambda_i T_i \}.
\]

Then we have

\[
\min_{x \in D} \max_{\omega \in Y} (f(x, \omega) - h_D(x, \omega))
\]

\[
= \min_{x \in D} \max_{\omega \in Y} (f(x, \omega) - \max_{\lambda \in \Lambda_x} \sum_{i=1}^{m} \lambda_i g(T_i, \omega))
\]

\[
= \min_{x \in D} \max_{\omega \in Y} \min_{\lambda \in \Lambda_x} (f(x, \omega) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega))
\]

\[
= \min_{x \in D} \max_{\omega \in Y} (f(x, \omega) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega))
\]

\[
= \min_{x \in D, \lambda \in \Lambda_x} \max_{\omega \in Y} (f(x, \omega) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega)).
\]

The third equality can be inferred from the assumption. The final programming (8) is obviously equivalent to the problem (RO – CRP').

The problem (RO-CRP') can be dealt by the robust optimization methods under some assumptions. For example, when \( f(x) \) is a homogeneous quadratic function of \( x \), and the uncertain set has special structure, the above problem (RO-CRP') is a robust convex quadratic programming which can be equivalent with a Second-Order-Cone-Programming (SOCP) which can be solved efficiently by the famous solver Yalmip.

As to the convex relaxation programming (CVaR-CRP) for the CVaR model (CVaR-DC), we can select some sections to approximate the integration. From the definition of the CVaR(X), we have

\[
CVaR_\alpha(f(x, \omega) - h_D(x, \omega)) = \frac{\int_{f(x, \omega) - h_D(x, \omega) \geq Y} R_\alpha(f(x, \omega) - h_D(x, \omega)) p(\omega) d\omega}{1 - \alpha}.
\]

Define the function \( F_\alpha \):

\[
F_\alpha(x, \xi) = \xi + \frac{1}{1 - \alpha} \int_{\omega \in R^n} [f(x, \omega) - h_D(x, \omega) - \xi]^+ p(\omega) d\omega,
\]

where \([t]^+ = t \) when \( t \geq 0 \), but \([t]^+ = 0 \) when \( t \leq 0 \). Note that the function \( F_\alpha(x, \xi) \) is convex and continuously differentiable, and the \( \alpha \)-CVaR of the loss associated with any \( x \in D \) can be determined from the following formula

\[
CVaR_\alpha(f(x, \omega) - h_D(x, \omega)) = \min_{\xi \in R} F_\alpha(x, \xi).
\]

The proof for the above results can be found in R. T. Rockafellar and S. Uryasev[12].
Furthermore, the integral in the definition of $F_{\alpha}(x, \xi)$ can be approximated in various ways. For example, it can be done by sampling the probability distribution of $\omega$ according to its density $P(\omega)$. If the sampling generates a collection of vectors $\omega_1, \omega_2, \cdots, \omega_q$, then the corresponding approximation to $F_{\alpha}(x, \xi)$ is

$$
\tilde{F}_{\alpha}(x, \xi) = \xi + \frac{1}{q(1-\alpha)} \sum_{k=1}^{q} [f(x, \omega_k) - h_D(x, \omega_k) - \xi]^+.
$$

The expression $\tilde{F}_{\alpha}(x, \xi)$ is convex and piecewise linear with respect to $\xi$. The minimization of $\tilde{F}_{\alpha}$ over $X \times R$, in order to get an approximate solution to the minimization of $F_{\alpha}$ over $X \times R$, can in fact be reduced to convex programming. In terms of auxiliary real variables $u_k$ for $k = 1, \cdots, q$, it is equivalent to minimizing the following convex programming:

$$
\begin{align*}
\min & \quad \xi + \frac{1}{q(1-\alpha)} \sum_{k=1}^{q} u_k \\
\text{s.t.} & \quad f(x, \omega_k) - h_D(x, \omega_k) - \xi \leq u_k, \quad k = 1, \cdots, q, \\
& \quad u_k \geq 0, \quad k = 1, \cdots, q, \\
& \quad x \in D = \text{conv}(T_1, \cdots, T_m)
\end{align*}
$$

The above problem is also equivalent to the following form (CVaR-CRP'):

$$
\begin{align*}
\min & \quad \xi + \frac{1}{q(1-\alpha)} \sum_{k=1}^{q} u_k \\
\text{s.t.} & \quad f(x, \omega_k) - \sum_{i=1}^{m} \lambda_i g(T_i, \omega_k) \leq u_k, \quad k = 1, \cdots, q, \\
& \quad \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, \quad x = \sum_{i=1}^{m} \lambda_i T_i, \\
& \quad u_k \geq 0, \quad k = 1, \cdots, q.
\end{align*}
$$

Since this problem (CVaR-CRP') is a convex programming, so it can be solved efficiently.

4.2. Framework of the global optimization algorithm. In this subsection a branch and bound algorithm is developed to solve the (DC) problem. This algorithm needs to solve a sequence of convex relaxation programming over partitioned subsets of $D$ in order to find a global solution. Here we consider two cases for the convex hull $D$: the box set and the simplex set. For the box set, we will adopt a simple and standard bisection rule, which drives all the intervals to zero for the variables along any infinite branch of the branch and bound tree to ensure convergence. For the simplex set, we will adopt simplex division which drives the simplex to a single point along any infinite branch of the branch and bound tree to ensure convergence. Next the two branch rules will be explained in detail.

Assume that the box $D^q$ is going to be divided, where $D^q = \{x : x^L_j(D^q) \leq x_j \leq x^U_j(D^q), j = 1, \cdots, n\}$. The vertex points set is denoted by $V^q$. $D^q$ is the convex hull of the vertex point set $V^q$. Then selection of the branching variable $x_v$ with the maximum length in $D^q$ and the partitioning of $D^q$ are realized by the following rules. Let $e = \arg \max \{x^L_v(D^q) - x^U_v(D^q)\}$, and without loss of generalization, partition $D^q$ by bisecting the interval $[x^L_v(D^q), x^U_v(D^q)]$ into subintervals $[x^L_v(D^q), (x^L_v(D^q) + x^U_v(D^q))/2]$ and $[(x^L_v(D^q) + x^U_v(D^q))/2, x^U_v(D^q)]$. Then we obtain two subboxes $D^{q,1}$ and $D^{q,2}$. Correspondingly, the vertex point sets of these two subboxes denoted by $V^{q,1}$ and $V^{q,2}$ can be obtained by replacing $x^L_v(D^q)$ with $x^L_v(D^q) + x^U_v(D^q)/2$ and $x^U_v(D^q)$ with $x^U_v(D^q) + x^L_v(D^q)/2$. Also $V^{q,2}$ can be obtained by replacing all the values of $x^L_v(D^q)$ with $x^L_v(D^q) + x^U_v(D^q)/2$.

Assume that the simplex $S^q$ is going to be divided, and the vertex point set is denoted by $V^q = \{V_1, \cdots, V_{n+1}\}$. $S^q$ is the convex hull of the vertex point set $V^q$. The partitioning of $S^q$ is realized by the following rules. Select the maximum
length side in $S^q$ connecting two points $V_{e_1}$ and $V_{e_2}$, and then locate the middle point of this section. So we can obtain two sub-simplexes $S^{q-1}$ and $S^{q-2}$. The vertex points set of these two sub-simplexes denoted by $V^{q-1}$ and $V^{q-2}$ can be obtained by replacing the point $V_{e_1}$ in $V^q$ with $\frac{V_{e_1}(S^{q})+V_{e_2}(S^{q})}{2}$, while $V^{q-2}$ can be obtained by replacing the point $V_{e_2}$ by $\frac{V_{e_1}(S^{q})+V_{e_2}(S^{q})}{2}$.

The basic steps of the proposed global optimization algorithm for the robust counterpart (RDC) and (CVaR-DC) are summarized as follows.

**Algorithm statement**

**Step 0. Initialization.**
0.1: A convergence tolerance $\delta$ is selected, and let $k = 0$, initial feasible node $Q_k = \{q(k)\} = \{1\}$, initial node indice $q(k) = 1$, $D^{q(k)} = D^1 = D$. Set an initial upper bound $U^* = \infty$.
0.2: Solve the convex relaxation programming $(RO - CRP)(D^{q(k)})$ (if (CVaR-DC) model is considered, the convex relaxation is $(C VaR - CRP)(D^{q(k)})$), and let $\hat{x}(D^{q(k)})$ denote the optimal solution, and $CRP_{q(k)}$ the minimum value. Then $U^* = \max_{\omega \in Y} (f(\hat{x}(D^{q(k)}), \omega) - g(\hat{x}(D^{q(k)}), \omega))$. If (CVaR-DC) model is considered, $U^* = C VaR_{\alpha}(f(\hat{x}(D^{q(k)}), \omega) - g(\hat{x}(D^{q(k)}), \omega))$. Set the initial lower bound $LB(k) = CRP_{q(k)}$:
0.3: If $U^* - LB(k) \leq \delta$, then stop with $\hat{x}(D^{q(k)})$ as the prescribed solution to the problem (DC).

**Step 1. (Partitioning step).** Partition $D^{q(k)}$ into two subsets $D^{q(k).1}$ and $D^{q(k).2}$. Replace $q(k)$ by these two new node indices $q(k).1, q(k).2$ in $Q_k$.

**Step 2. (Updating upper bound).** For the two subsets, updating the vertex points sets $V^{q(k).s}$ where $s = 1, 2$. Solve $(RO - CRP)(D^{q(k).s})$ or $(C VaR - CRP)(D^{q(k).s})$ respectively, and let $\hat{x}(D^{q(k).s})$ denote the optimal solution, and $CRP_{q(k).s}$ the minimum value. Then update upper bound

$$U^* = \min\{U^*, \max_{\omega \in Y} (f(\hat{x}(D^{q(k).s}), \omega) - g(\hat{x}(D^{q(k).s}), \omega))\}.$$  

If (CVaR-DC) model is considered, $U^* = \min\{U^*, C VaR_{\alpha}(f(\hat{x}(D^{q(k).s}), \omega) - g(\hat{x}(D^{q(k).s}), \omega))\}$.

**Step 3. (Fathoming step).** Fathom any non-improving nodes by setting $Q_{k+1} = Q_k - \{q \in Q_k : CRP_q \geq U^* - \delta\}$. If $Q_{k+1} = \emptyset$ then stop, and $U^*$ is the optimal value, $(x^*(\kappa))$ (where $\kappa \in V$) are the global solutions, where $V = \{\kappa : \max_{\omega \in Y} \{f(\kappa), \omega) - g(\kappa, \omega))\} = U^*\}$. If (CVaR-DC) model is considered, $V = \{\kappa : C VaR_{\alpha}(f(\kappa), \omega) - g(\kappa, \omega))\} = U^*\}$. Otherwise, $k = k + 1$.

**Step 4. (Node selection step).** Set the lower bound $LB(k) = \min\{CRP_q : q \in Q_k\}$, then select an active node $q(k) \in \arg\min\{\{LB(k)\}$ for further considering, and then return to step 1.

**Theorem 4.1. (convergence result):** The above algorithm either terminates finitely with the incumbent solution being optimal to (RDC)/(CVaR-DC), or generates an infinite sequence of iteration. Along any infinite branch of the branch and bound tree, any accumulation point of the sequence $x^*$ will be the global solution of the problem (RDC)/(CVaR-DC).

**Proof.** A sufficient condition for a global optimization to be convergent to the global minimum, stated in Horst and Tuy [7] requires that the bounding operation must be consistent and the selection operation bound must be improving. A bounding operation is called consistent if at every step any unfathomed partition can be further
refined, and if any infinitely decreasing sequence of successively refined partition elements satisfied:

$$\lim_{k \to +\infty} (U^* - LB(k)) = 0. \quad (9)$$

where $LB(k)$ is a lower bound inside some subset in stage $k$ and $U^*$ is the best upper bound at iteration $k$ not necessarily occurring inside the above same subset. In the following we will demonstrate equation (9) holds.

Since the subdivision process is the bisection and simplex division, the processes are exhaustive. Consequently, from the discussion [7] (9) holds, it means that the bounding operation is consistent.

A selection operation is called bound improving if at least one partition element where the actual lower bound is attained is selected for further partition after a finite number of refinements. Clearly, the selection operation is bound improving because the partition element where the actual lower bound is attained is selected for further partition in the immediately following iteration.

In summary, we have shown that the bounding operation is consistent and that the selection operation is bound improving, therefore according to Theorem IV.3. in Horst and Tuy [7] the employed global optimization algorithm is convergent to the global solutions.

5. **Numerical experiments.** In this section, several numerical experiments are carried out on a PC with Intel Pentium 4 2.30GHz CPU, 2048MB RAM, Matlab 2013b. In the following DC problems under uncertainty environments, we set the termination tolerances $\varepsilon = 10^{-4}$.

$$\begin{align*}
\min & \quad x^TW_1x - (x^TW_2x + x_1 + x_2 + x_3 + 1) \\
\text{s.t.} & \quad x \in D = \text{conv}\{T_1, \cdots, T_m\}
\end{align*}$$

where $W_i = A_i^TA_i$ ($i = 1, 2$) are random matrix generated by $A_i^{n \times n}$, in which the elements are followed by the uniform distribution $(0,1)$. Two different kinds of convex hull $D$ will be included: the simplex feasible set and the box feasible set. For each case, two models (RDC) and (CVaR-DC) are adopted to treat the uncertainty in the primal DC problems respectively.

(1) Simplex constraints

In the first experiment, $D$ is a simplex feasible set generated by the four vectors $a_1 = \{0,0,0\}, a_2 = \{0,0,3\}, a_3 = \{0,3,0\}, a_4 = \{3,0,0\}$. The global optimization algorithm designed in Section 4 can be applied to the CVaR model and Robust model, respectively. The results are shown in Table 1.

We first plot the optimal values of Robust model and CVaR model with increasing value $\alpha$ in Figure 3. The optimal value of Robust model is independent with the value of $\alpha$, so a line can be drawn. The optimal value of Robust model is always above the optimal value of CVaR model, which we have already proved in Proposition 3.3. When $\alpha = 0.99$, the optimal value is -1.4221, the same as the Robust one. In fact, when $\alpha \to 1$, the limit value of the CVaR model turns to be the solution of the robust cases. In the right column, we list the global solutions of the problems derived by using the solver BARON. From the Table, the global solutions by BARON are the same with the one derived by our algorithm, which shows the quality of the solutions derived by our algorithm.

(2) Box constraints
Table 1.

| α   | CPU(s) | Step | Nodes | Opt Solution      | Opt Value | Opt* Value |
|-----|--------|------|-------|-------------------|-----------|------------|
| 0.70| 356.11 | 87   | 69    | (0.0000, 1.1250, 1.8750)ᵀ | -2.2690   | -2.2689    |
| 0.75| 956.11 | 163  | 101   | (0.0000, 1.0547, 1.5703)ᵀ | -2.1214   | -2.1214    |
| 0.80| 847.73 | 163  | 106   | (0.0000, 0.7969, 1.1191)ᵀ | -1.9628   | -1.9628    |
| 0.85| 516.92 | 132  | 106   | (0.0000, 0.6211, 0.8789)ᵀ | -1.7508   | -1.7508    |
| 0.90| 518.31 | 122  | 107   | (0.0000, 0.4569, 0.6797)ᵀ | -1.5661   | -1.5663    |
| 0.95| 321.43 | 78   | 68    | (0.0000, 0.3580, 0.5326)ᵀ | -1.4453   | -1.4452    |
| 0.97| 143.17 | 31   | 27    | (0.0000, 0.3387, 0.5038)ᵀ | -1.4228   | -1.4228    |
| 0.98| 160.39 | 30   | 24    | (0.0104, 0.3437, 0.5156)ᵀ | -1.4222   | -1.4222    |
| 0.99| 167.81 | 31   | 26    | (0.0023, 0.3281, 0.5000)ᵀ | -1.4221   | -1.4221    |
| Robust | 218.18 | 52 | 51 | (0.0080, 0.3515, 0.5002)ᵀ | -1.4221   | -1.4221    |

Figure 3. Optimal value comparison of Robust Model and CVaR model.

In this case, $D$ is a box set constructed by eight vertex points, $a_1 = \{0, 0, 0\}, a_2 = \{0, 0, 1\}, a_3 = \{0, 2, 0\}, a_4 = \{3, 0, 0\}, a_5 = \{3, 0, 1\}, a_6 = \{3, 2, 1\}, a_7 = \{3, 2, 0\}, a_8 = \{3, 2, 1\}$

The results of the numerical test are shown in Table 2. The Robust model and CVaR model with box feasible set can be also solved by our branch and cut algorithm. Furthermore, compared with Table 2, we can find that the steps and the cost of CPU times are less than that of the simplex ones.

Finally, two different concave constructions are compared: our RDC and Floudas in [8]. We choose the following DC problems under uncertainty environments.

$$
\begin{align*}
\min & \quad x^T W_1 x - \exp(W_2^T x) = f(x) - g(x) \\
\text{s.t.} & \quad x \in D = \text{conv}(T_1, \ldots , T_m)
\end{align*}
$$

where $W_1 = A_1^T A_1$ are random matrix generated by $A_i^{n \times n}$, in which the elements are followed by normal distribution $(0, 1)$, and $W_2^{n \times 1}$ follows uniform distribution in $(0, 1)$. We show our results in Table 3 and Table 4 through different random seeds.

If the linear overestimate is generated by the method in [8], then the algorithms will produce more nodes and cost more CPU time than our methods. Our concave reconstruction RDC produces only 11 nodes in simplex feasible and 8 nodes in box feasible for seed 1, which are much less than Floudas. Thus, our estimate is better.
Table 2.

| $\alpha$ | CPU(s) | Step | Nodes | Opt Solution          | Opt Value | $Opt^*$ |
|---------|--------|------|-------|-----------------------|-----------|---------|
| 0.70    | 264.90 | 24   | 22    | (0.0000, 0.6719, 0.9844)$^T$ | -2.2410  | -2.2410 |
| 0.75    | 105.19 | 25   | 29    | (0.0000, 0.6641, 0.9844)$^T$ | -2.1214  | -2.1215 |
| 0.80    | 217.65 | 24   | 19    | (0.0000, 0.7187, 0.9844)$^T$ | -1.9520  | -1.9520 |
| 0.85    | 516.92 | 55   | 40    | (0.0000, 0.6250, 0.8750)$^T$ | -1.7506  | -1.7505 |
| 0.90    | 350.26 | 39   | 34    | (0.0000, 0.4531, 0.6741)$^T$ | -1.5661  | -1.5661 |
| 0.95    | 208.85 | 38   | 32    | (0.0000, 0.3580, 0.5326)$^T$ | -1.4452  | -1.4452 |
| 0.97    | 143.17 | 31   | 27    | (0.0000, 0.3437, 0.5156)$^T$ | -1.4228  | -1.4228 |
| 0.98    | 167.81 | 30   | 26    | (0.0104, 0.3437, 0.5156)$^T$ | -1.4222  | -1.4222 |
| 0.99    | 160.39 | 31   | 24    | (0.0023, 0.3281, 0.5000)$^T$ | -1.4221  | -1.4221 |
| Robust  | 216.98 | 35   | 27    | (0.0080, 0.3515, 0.5002)$^T$ | -1.4221  | -1.4221 |

Table 3. Simpex feasible

| CPU Time(s) | Nodes | SEED |
|-------------|-------|------|
| RDC         | 91    | 11   |
| Floudas     | 227   | 42   |
| RDC         | 100   | 15   |
| Floudas     | 204   | 40   |
| RDC         | 120   | 25   |
| Floudas     | 280   | 75   |

Table 4. Box feasible

| CPU Time(s) | Nodes | SEED |
|-------------|-------|------|
| RDC         | 88    | 8    |
| Floudas     | 220   | 40   |
| RDC         | 105   | 12   |
| Floudas     | 207   | 36   |
| RDC         | 117   | 20   |
| Floudas     | 281   | 68   |

than that in [8]. On average, our RDC produces less nodes and cost less time, when solving DC problems under uncertain environments.

6. Conclusions and future work. A new concave reformulation is proposed in this paper and then is applied to solve the DC programming globally under uncertain environment by constructing a concave approximation. Then, this paper proves the connection of the optimal values between the robust counterpart and the CVaR model: when $\alpha \to 1$, the optimal value of CVaR model will converge to the optimal value of the Robust counterpart. Finally, the numerical experiments are given for a kind of special DC program under uncertain environment with two kinds of constraints: simplex set and box set. Simulation results show the effectiveness of the proposed global optimization algorithm and the correctness of the proposed results. We have proved the general case of concave approximation for the Robust case. As for CVaR, we only have the result of specific concave approximation. There is still room for improvement in the suggested approach. Numerical experiments in
this paper have been conducted to solve small-scale problems. Additional research needs to be conducted to solve large-scale problems.

REFERENCES

[1] A. Ben-Tal and A. Nemirovski, Robust convex optimization, Mathematics of Operations Research, 23 (1998), 769–805.

[2] A. Ben-Tal and A. Nemirovski, Robust solutions of uncertain linear programs, Operations Research letters, 25 (1999), 1–13.

[3] A. Ben-Tal and A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data, Mathematical Programming, 88 (2000), 411–424.

[4] T. P. Dinh, The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems, Annals of Operations Research, 113 (2005), 23–46.

[5] T. P. Dinh and A. Le Thi Hoai, A DC optimization algorithm for solving the trust-region subproblem, SIAM J. Optim., 8 (1998), 476–505.

[6] A. Edward, H. F. Xu and D. L. Zhang, Confidence levels for cva risk measures and minimax limits, Business Analytics, 2014.

[7] R. Horst and H. Tuy, Global Optimization: Deterministic Approaches, Second edition. Springer-Verlag, Berlin, 1993.

[8] C. D. Maranas and C. A. Floudas, Global optimization in generalized geometric programming, Computers & Chemical Engineering, 21 (1997), 351–369.

[9] G. C. Pflug, Some remarks on the value-at-risk and the conditional value-at-risk, Probabilistic constrained optimization, 8 (2000), 272–281.

[10] H. Reiner and T. Nguyen V, Robust solutions of linear programming problems contaminated with uncertain data, Journal of Optimization Theory and Applications, 103 (1999), 1–43.

[11] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, N.J., 1970.

[12] R. T. Rockafellar and S. Uryasev, Optimization of conditional value-at-risk, Journal of Risk, 2 (2000), 21–41.

[13] R. T. Rockafellar and S. Uryasev, Conditional value-at-risk for general loss distributions, Journal of Banking & Finance, 26 (2002), 1443–1471.

[14] R. T. Rockafellar, Coherent approaches to risk in optimization under uncertainty, OR Tools and Applications: Glimpses of Future Technologies, 8 (2007), 38–61.

[15] P. P. Shen and C. F. Wang, Global optimization for sum of linear ratios problem with coefficients, Applied Mathematics and Computation, 176 (2006), 219–229.

[16] Y. J. Wang and Y. Lan, Global optimization for special reverse convex programming, Computers & Mathematics with Applications, 55 (2008), 1154–1163.

Received July 2018; 1st revision December 2018; final revision January 2019.

E-mail address: wangyj@mail.shufe.edu.cn
E-mail address: tonyshenkaiji@gmail.com