On the Solution of Parametric Bi-Level Quadratic Programming Problem

T.I. Sultan\textsuperscript{1}, O.E. Emam\textsuperscript{2} and S.A. Nasr\textsuperscript{3}

\textsuperscript{1, 2, 3} Department of Information Systems  
Faculty of Computer Science and Information  
Helwan University, P.O. Box - 11795, Egypt

\textsuperscript{2} E-mail: emam_o_e@yahoo.com  
\textsuperscript{3} E-mail: anwer_safaa@yahoo.com

(Received: 13-8-13 / Accepted: 25-9-13)

Abstract

This paper presents a parametric bi-level quadratic programming problem with random rough coefficient in objective functions. An approach combines the convert technique of rough coefficient and Stackelberg strategy. An auxiliary problem is discussed and an algorithm as well as an example is presented.

Keywords: Rough parameter, Quadratic programming, bi-level programming, rough in objective.

1 Introduction:

Bi-level programming is a class of multi-level programming which is computationally more complex and expensive that conventional mathematical programming. Several bi-level programming problem and their solution method have been presented, such as, the hybrid extreme-point search algorithm, mixed-integer problem with complementary slackness, and the penalty function approach.
In [5] Pramanik proposed a multi-level optimization by relaxation provided each level decision maker of bounds on the decision variables. Using two fuzzy goal programming models, reach the highest degree of membership goals by minimizing negative deviational variables.

Another approach of vagueness is the rough set theory. The rough set expressed by boundary region of set which described by lower and upper approximation, and the set is considered as crisp set if the boundary region is empty, this is exactly the idea of vagueness. The approach for solving rough programming problems is to convert the objective function from rough to crisp using theorem of crisp evaluation.

In [1] Osman et al. presented rough bi-level programming problems using genetic algorithm (GA) by constructing the fitness function of the upper level programming problems based on the definition of through feasible degree.

This paper is organized as follows: it starts from section 2 formulating the problem, in section 3, 4 introduce the theorems used to turned from rough to crisp variable and to fuzzy using membership function, section 5 is providing an algorithm of the solution technique, then section 6 presents a numerical example illustrates the theory of solution. Finally, section 6 the conclusion.

2 Problem Formulation and Solution Concept

The bi-level quadratic programming problem with random rough coefficient in the objective functions may be written as:

\[ \text{[1st level]} \]
\[ \max_{x_1} f_1(x) = kx + \frac{1}{2} x^T C x, \]  \hspace{1cm} (1)

Where \( x_2 \) solves

\[ \text{[2nd level]} \]
\[ \max_{x_2} f_2(x) = kx + \frac{1}{2} x^T C x, \]  \hspace{1cm} (2)

Subject to

\[ G = \{x | Ax \leq b, x \geq 0\}. \]  \hspace{1cm} (3)

Where \( f_1, f_2 \) are the objective functions of the first level decision maker (FLDM), and second level decision maker (SLDM), \( C \) is \( n \times n \) real matrix contain random rough coefficient and \( k \) is \( (1 \times n) \) matrix, \( A \) is \( (m \times n) \) real matrix and \( b \) is \( (m \times 1) \) matrix, the vector of decision variables \( x \) is \( (n \times 1) \) matrix partitioned
between the three planners. The first-level decision maker has control over the vector $x_1 \in \mathbb{R}^{n_1}$, and the second-level decision maker has control over the vector $x_2 \in \mathbb{R}^{n_2}$, where $\mathcal{R} = n_1 + n_2$.

**Definition 2.1** [9] Let $\Lambda$ be a nonempty set, $C$ be an $\sigma$-algebra of subset of $\Lambda$, $\Delta$ be an element in $C$, and $\pi$ be a nonnegative, real-valued, additive set function. Then $(\Lambda; \Delta; C; \pi)$ is called a rough space.

**Definition 2.2** [9] A rough variable $\xi$ on the rough space $(\Lambda; \Delta; C; \pi)$ is a function from $\Lambda$ to the real line $\mathbb{R}$ such that for every Borel set $\mathcal{B}$ of $\mathbb{R}$, we have $(\lambda \in \Lambda | \xi(\lambda) \in \mathcal{B}) \in \mathcal{C}$. The lower and the upper approximations of the rough variable $\xi$ are then defined as follows: $\xi^- = \{\xi(\lambda) | \lambda \in A\}, \xi^+ = \{\xi(\lambda) | \lambda \in \Delta\}$.

**Definition 2.3** [9] Let $\hat{\xi}$ be a rough vector on the rough space $(\Lambda; \Delta; C; \pi)$, and $f_j: \mathbb{R}^n \rightarrow \mathbb{R}$ be continuous functions, $j = 1,2,\ldots,m$, then the upper trust of the rough event characterized by $f_j(\xi) \leq 0, j = 1,2,\ldots,m$, is defined by $\text{Tr}^* \{f_j(\xi) \leq 0, j = 1,2,\ldots,m\} = \frac{\pi(\lambda \in \Delta | f_j(\xi) \leq 0, j = 1,2,\ldots,m)}{\pi(\Lambda)}$.

**Definition 2.4** [9] Let $\xi$ be a rough vector on the rough space $(\Lambda; \Delta; C; \pi)$, and $f_j: \mathbb{R}^n \rightarrow \mathbb{R}$ be continuous functions, $j = 1,2,\ldots,m$. Then the lower trust of the rough event characterized by $f_j(\xi) \leq 0, j = 1,2,\ldots,m$, is defined by $\text{Tr} \{f_j(\xi) \leq 0, j = 1,2,\ldots,m\} = \frac{\pi(\lambda \in \Delta | f_j(\xi) \leq 0, j = 1,2,\ldots,m)}{\pi(\Delta)}$.

### 3 The Transformation of Random Rough Coefficient [9]

To convert the bi-level quadratic programming problem with random rough coefficient in the objective functions into the respective crisp equivalents for solving a trust probability constraints, this process is usually hard work for many cases but the transformation process is introduced in the following theorem.

**Theorem 3.1** [9] Assume that random rough variable $\bar{c}_{ij}$ is characterized by $\bar{c}_{ij}(\hat{\epsilon}) \sim N(c_{ij}(\lambda), V_{ic})$ where:

$c_{ij}(\lambda)[c_{ij}(\lambda)_{n \times 1} = c_{i1}(\lambda), c_{i2}(\lambda), \ldots, c_{in}(\lambda)^T]$ is a rough variable and $V_{ic}$ is a positive definite covariance matrix. It follows that $c_{i}(\lambda)^T x = ([a,b],[c,d])$, (where $c \leq a \leq b \leq d$) is a rough variable and characterized by the following trust measure function:

- if $d \leq t$,
- if $b \leq t \leq d$,
- if $a \leq t \leq b$,
- if $c \leq t \leq a$,
46

\[ \text{Tr}\{c_i(\lambda)^T x \geq t}\} = \begin{cases} 
0 & \text{if } t \leq c, \\
\frac{1}{2} \left( \frac{d-t}{d-c} + \frac{b-t}{b-a} \right) & \text{if } d \leq M \leq d, \\
\frac{1}{2} \left( \frac{d-t}{d-c} + 1 \right) & \text{if } a \leq M \leq b, \\
1 & \text{if } c \leq M \leq a, \\
\end{cases} \]

Then, we have \( \text{Tr} \{ \lambda \mid \text{Pr} \{ c_i(\lambda)^T x \geq f_i(x) \} \geq \gamma_i \} \geq \gamma_i \), if and only if

\[
\begin{align*}
\begin{cases}
\begin{aligned}
b + R &\leq f_i \leq d - 2\gamma_i(d - c) + R \\
a + R &\leq f_i \leq \frac{d(b-a)+b-2\gamma_i(d-c)(b-a)}{d-c+b-a} + R \\
c + R &\leq f_i \leq d - (d - c)(2\gamma_i - 1) + R \\
f_i &\leq c + R 
\end{aligned}
\end{cases}
\end{align*}
\]

Where \( M = f_i - \Phi^{-1}(1 - \delta_i) \sqrt{x^T V_i c x} \) and \( R = \Phi^{-1}(1 - \delta_i) \sqrt{x^T V_i c x} \) and \( \Phi \) is the standardized normal distribution and \( \delta_i, \gamma_i \in [0,1] \) are predetermined confidence levels.

4 Fuzzy Approach of Bi-level Quadratic Programming Problem [4]

To solve a crisp bi-level quadratic programming program, one first gets the satisfactory solution that is acceptable to FLDM, and then gives the FLDM decision variable and goal with some leeway to the SLDM for him/her to seek the optimal solution, and to arrive at the solution which is closest to the optimal solution of the FLDM. This due to, the SLDM who should not only optimize his/her objective function but also try to satisfy the FLDM’s goal and preference as much as possible.

4.1 FLDM Problem

First, the FLDM solves the following Problem:

\[
\begin{align*}
&\max h_1(x), \\
&\text{Subject to } x \in G.
\end{align*}
\]

(4)

To build membership function, goals and tolerances should be determined first. However, they could hardly be determined without meaningful supporting data. We should first find the individual best solution \( h_1^* \) and individual worst solution \( \bar{h}_1 \) of (8), where:

\[
h_1^* = \max_{x \in G} h_1(x), \quad \bar{h}_1 = \min_{x \in G} h_1(x)
\]

(5)
This data can then be formulated as the following membership function:

\[
\mu[h_1(x)] = \begin{cases} 
1 & h_1(x) > h_1^*, \\
\frac{h_1(x) - \bar{h}_1}{\bar{h}_1 - h_1} & \bar{h}_1 \leq h_1(x) \leq h_1^*, \\
0 & \bar{h}_1 \geq h_1(x).
\end{cases}
\]  

(6)

Now, we can get the solution of the FLDM problem by solving the following Tchebycheff problem:

\[
\max \lambda \\
\text{Subject to}
\]

\[
x \in G, \\
\mu[h_1(x)] \geq \lambda, \\
\lambda \in [0,1].
\]

The FLDM solution is assumed to be \([\lambda^f, x_1^f, x_2^f, h_1^f]\) where \(\lambda^f\) is satisfactory level for FLDM.

### 4.2 SLDM Problem

Second, in the same way, the SLDM independently solves:

\[
\max h_2(x), \\
\text{Subject to } x \in G.
\]

The individual best solution \(h_2^*\) and individual worst solution \(\bar{h}_2\) of (12), where

\[
h_2^* = \max_{x \in G} h_2(x), \quad \bar{h}_2 = \min_{x \in G} h_2(x)
\]

(9)

This information can then be formulated as the following membership function:

\[
\mu[h_2(x)] = \begin{cases} 
1 & h_2(x) > h_2^*, \\
\frac{h_2(x) - \bar{h}_2}{\bar{h}_2 - h_2^*} & \bar{h}_2 \leq h_2(x) \leq h_2^*, \\
0 & \bar{h}_2 \geq h_2(x).
\end{cases}
\]  

(10)

Now the solution of the SLDM can be obtained by solving the following Tchebycheff problem:
\[
\max \lambda,
\]

Subject to
\[
x \in G,
\]
\[
\mu[h_2(x)] \geq \lambda,
\]
\[
\lambda \in [0,1).
\]

The FLDM solution is assumed to be \([\lambda^S, x_1^S, x_2^S, h_2^S] \) where \(\lambda^S\) is satisfactory level for SLDM.

Now the solution of the FLDM and SLDM are disclosed. But, two solutions are usually different because of nature between two levels objective functions. The FLDM knows that using the optimal decisions \(x_1^F\) as a control factors for the SLDM, is not practical. It is more reasonable to have some tolerance that gives the SLDM an extent feasible region to search for his/her optimal solution, and reduce searching time or interactions. In this way, the range of decision variable \(x_1\) should be around \(x_1^F\) with maximum tolerance \(t_1\) and the following membership function specify \(x_1\) as:

\[
\mu(x_1) = \begin{cases} 
\frac{x_1-(x_1^F-t_1)}{t_1} & x_1^F - t_1 \leq x_1 \leq x_1^F, \\
\frac{-x_1+(x_1^F+t_1)}{t_1} & x_1^F \leq x_1 \leq x_1^F - t_1.
\end{cases}
\]

(12)

The FLDM goals may reasonably consider \(h_1 \geq h_1^F\) is absolutely acceptable and \(h_1 \leq h_1^F\) is absolutely unacceptable, and that the preference with \([h_1^F, h_1^F]\) is linearly increasing. This due to the fact that the SLDM obtained the optimum at \((x_1^S, h_1^S)\), which in turn provides the FLDM the objective function values \(h_1\), make any \(h_1 \geq h_1 = h_1 (x_1^S, x_2^S)\) unattractive in practice.

The following membership functions of the FLDM can be stated as:

\[
\hat{\mu}[h_1(x)] = \begin{cases} 
1 & h_1(x) > h_1^F, \\
\frac{h_1(x)-h_1^F}{h_1^F-h_1} & h_1^F \leq h_1(x) \leq h_1^F, \\
0 & h_1 \geq h_1(x).
\end{cases}
\]

(13)

Second, the SLDM goals may reasonably consider the \(h_2 \geq h_2^S\) is absolutely acceptable and \(h_2 \leq h_2^S\) is absolutely unacceptable, and that the preference with \([h_2^S, h_2^S]\) is linearly increasing. In this way, the SLDM has the following membership functions for his/her goal:
\[ \hat{\mu}[h_2(x)] = \begin{cases} 
1 & h_2(x) > h_2^s, \\
\frac{h_2(x) - h_2}{\hat{h}_2 - h_2} & \hat{h}_2 \leq h_2(x) \leq h_2^s, \\
0 & \hat{h}_2 \geq h_2(x). 
\end{cases} \] (14)

Finally, in order to generate the satisfactory solution, which is also a Pareto optimal solution with overall satisfaction for all decision-makers, we can solve the following Tchebycheff problem.

\[
\max \beta, \tag{15}
\]

Subject to

\[
\hat{\mu}[h_1(x)] \geq \beta, \\
\hat{\mu}[h_2(x)] \geq \beta, \\
\frac{[x_1 - (x^f_1 - t_1)]}{t_1} \geq \beta, \\
\frac{-x_1 + (x^f_1 + t_1)}{t_1} \geq \beta, \\
x \in G, \\
t_i > 0, \ \beta \in [0,1].
\]

5 Numerical Example:

To demonstrate the solution method for bi-level quadratic programming problem under random rough coefficient in objective functions (1, 2) can be written as:

[1st level]

\[
\max_{x_1} f_1(x) = 3c_1x_1^2 + x_2 + 5x_1,
\]

Where \( x_2 \) solves

[2nd level]

\[
\max_{x_2} f_2(x) = 2x_1 + 6c_2x_2^2,
\]

Subject to

\[
\text{Tr}\{\lambda|\text{Pr}\{k_{11}c_1x_1^2 + k_{12}x_2 + 5x_1 \geq f_1 \geq \delta_i \} \geq \gamma_i \} \geq \gamma_i.
\]
\[
\text{Tr}\{\lambda | \text{Pr}\{k_{21}x_1^2 + k_{22}c_2x_2 \geq f_2\} \geq \delta_i\} \geq y_i.
\]

\begin{align*}
3x_1 + 5x_2 & \leq 45, \\
2x_1 - x_2 & \leq 15, \\
3x_1 + 2x_2 & \leq 30, \\
x_1, x_2 & \geq 0.
\end{align*}

Assume that the rough parameters are defines as:

\begin{align*}
c_1 & \sim N(\rho_1, 2) \text{ With } \rho_1 \in [1,2], [1,3], \\
c_2 & \sim N(\rho_2, 3) \text{ With } \rho_2 \in [2,4], [2,5],
\end{align*}

Let \( \delta_i = y_i = 0.4 \), then \( \Phi^{-1}(1 - \delta_i) = 0.26 \).

Now by using theorem 1, the equivalent crisp problem which equivalent to bi-level quadratic programming problem under rough parameters in objective functions is:

\begin{align*}
\text{[1st level]} \\
\max_{x_1} h_1 & = 4.2x_1^2 + x_2 + 5x_1 + 0.26\sqrt{2x_1^4}, \\
\text{Where } x_2 \text{ solves } \\
\text{[2nd level]} \\
\max_{x_2} h_2 & = 2x_1 + 15.6x_2^2 + 0.26\sqrt{4x_2^4}, \\
\text{Subject to } \\
x & \in G = \{3x_1 + 5x_2 \leq 45, \\
2x_1 - x_2 & \leq 15, \\
3x_1 + 2x_2 & \leq 30, \\
x_1, x_2 & \geq 0.\}
\end{align*}

Then, calculating trust for every rough coefficients using trust measure function in theorem 1:

\begin{align*}
c_1 & = \frac{4.2}{3} = 1.4 \in lower \ [1,2], [1,3], \\
c_2 & = \frac{15.6}{6} = 2.6 \in lower \ [2,4],[2,5],
\end{align*}

To solve the equivalent crisp problem of bi-level quadratic programming problem:
First: the FLDM solves his/her problem as following:

1- Find individual optimal solution by solving (5), we get: \( (h_1^*, h_1) = (380.5858, 0) \)
2- By using (6), the FLDM build membership function \( \mu h_1(x) \) then solve the Tchebycheff problem (7) as follows:

\[
\max \lambda , \\
\text{Subject to} \\
4.2x_1^2 + x_2 + 5x_1 + 0.26 \sqrt{2x_1^4} \leq 380.5858\lambda , \\
x \in G , \\
\lambda \in [0,1].
\]

Whose solution is \( \lambda^T = 0.7 , (x_1^T, x_2^T) = (7.5,0) , h_1^T = 294.4329 \).

Second, the SLDM solves his / her problem as follows:

1- Finds individual optimal solutions by solving (9), we get: \( (h_2^*, \tilde{h}_2) = (1300.077,0) \).

2- By using (10), the SLDM build membership function \( \mu h_2(x) \) then solve the Tchebycheff problem (11) as follows:

\[
\max \lambda , \\
\text{Subject to} \\
2x_1 + 15.6x_2 + 0.26 \sqrt{4x_2^4} \leq 1300.077\lambda , \\
x \in G , \lambda \in [0,1].
\]

Whose solution is \( \lambda^S = 0.62 , (x_1^S, x_2^S) = (1.529219, 7.07315) , h_2^S = 806.0478 \).

Finally, calculating the membership function for Tchebycheff problem:

1- We assume the FLDM’S control decision \( x_1^F = 7.5 \) with the tolerance 1.8

2- By using (13) – (14) calculating membership functions \( \hat{\mu} \), then solves the Tchebycheff problem (15) as follows:

\[
\max \beta , \\
\text{Subject to} \\
x \in G , \\
4.2x_1^2 + x_2 + 5x_1 + 0.26 \sqrt{2x_1^4} - 269.03205\beta \geq 25.40085 , \\
2x_1 + 15.6x_2 + 0.26 \sqrt{4x_2^4} - 791.0478\beta \geq 15 , \\
x_1 - 1.8\hat{a} \geq 5.7 , \\
x_1 + 1.8\hat{a} \geq 9.3 , \\
\beta \in [0,1].
\]

Whose, optimal solution is: \( \beta = 0.9588 , (x_1^0, x_2^0) = (8.571429,2.142857) \).
\((h_1^0, h_2^0) = (380.5858, 90.84336)\) Overall satisfaction for both decisions makers.

6 Conclusion:

This paper presented a parametric bi-level quadratic programming problem with random rough coefficient in objective functions. An approach combines the convert technique of rough coefficient and Stackelberg strategy. An auxiliary problem is discussed and an algorithm as well as an example is presented. However, there are many other aspects, which should be explored and studied in the area of a multi-level optimization such as:

1- Large scale multi-level multi-objective non-linear programming problem with rough parameters in objective.
2- Large scale multi-level multi-objective non-linear programming problem with rough parameters in constraints.
3- Large scale multi-level multi-objective non-linear programming problem with rough parameters in both objectives and constraints.

References