

Optimization Methods for Decision Making under Uncertainty

Kota Matsui¹

¹Department of Computer Science and Mathematical Informatics

January 23, 2017

Supervisor : Professor Takafumi Kanamori

Contents

Abstract	4
1 Introduction	5
1.1 Decision Making as Optimization Problem	5
1.2 Multi-Objective Optimization	6
1.3 Main Contribution	8
1.4 Outline	10
1.4.1 Part I : Multi-Objective Optimization for Decision Making under Un- certainty	10
1.4.2 Part II : A Novel Algorithm for Optimization with Uncertain Information	10
I Multi-Objective Optimization for Decision Making under Uncertainty	11
2 Various Formulation of Multi-Objective Optimization under Uncertainty	11
2.1 Multi-Objective Optimization under Uncertainty	11
2.2 Hierarchical Multi-Objective Optimization under Uncertainty	12
3 Hierarchical Multi-objective Stochastic Linear Optimization Based on the Fuzzy Decision	13
3.1 Problem Formulation and Corresponding Pareto Optimality	13
3.2 Deriving a Satisfactory Solution based on the Fuzzy Decision	14
3.3 An Interactive Algorithm	19
3.4 A Numerical Experiment	20
4 Satisfaction maximization model for Hierarchical Multi-objective Fuzzy Random Linear Programming Problems	22
4.1 Problem Formulation	22
4.2 Fractile Optimization Model and Corresponding Pareto Optimality	23
4.3 An Interactive Algorithm	27
4.4 A Numerical Experiment	29
5 Random Fuzzy Multi-objective Linear Programming with Variance Covariance Matrices	30
5.1 Problem Formulation	30
5.2 An Interactive Algorithm	36
5.3 A Numerical Example	38
5.3.1 Simulation with Agricultural Planning in Mind	38
5.3.2 Comparison with Probability Maximization Model	40
6 Summary of Part I	41
II A Novel Algorithm for Optimization with Uncertain Information	42

7	Derivative-Free Methods for General Single-Objective Optimization	42
8	Preliminaries	45
9	Main Results	45
9.1	Algorithm	45
9.2	Convergence Properties of our Algorithm under Deterministic Oracle	47
9.3	Query Complexity	52
9.4	Generalization to Stochastic Pairwise Comparison Oracle	54
10	Numerical Experiments	57
10.1	Existing Methods and Preliminary Experiments	57
10.2	Numerical Experiments of Parallel Computation	58
11	Summary of Part II	61
12	Concluding Remarks	63
	References	64

Acknowledgements

First of all, I would like to express my deep appreciation to my supervisor, professor Takafumi Kanamori. The research activities at Kanamori Lab constitute the foundation on my current research. I have learned many things from professor Kanamori such as not only the topic of research but also research attitude.

I would also like to thank to my collaborator, assistant professor Wataru Kumagai in Kanagawa University. I have continued to collaborate with him since I was a PhD student. He always shows me interesting ideas and deep suggestions.

Furthermore, I would like to thank professor Hitoshi Yano who is my advisor from undergraduate to master's course at Nagoya City University. I have learned about decision making and optimization problems from professor Yano, and we wrote papers with respect to decision making problem under uncertainty conditions.

Finally but not least, I would like to thank my parents and my wife Rio for their support.

Abstract

Decision making is one of the most familiar problems that everyone always faces, although there are differences in degree. Various decision making problems that we face in real world can very often be formulated as mathematical optimization problem. In addition, such problems often include various ambiguity such as stochastic uncertainty (e.g. weather, stock prices, ...) or subjective uncertainty (e.g. What is sunny day?, What is high stock price?, ...). Therefore, when formulating actual decision-making problems as mathematical optimization problems, unless these uncertainties are dealt with appropriately, there is a danger of adopting poor options.

This dissertation describes novel frameworks for the design and analysis of mathematical optimization problems for decision making under uncertainty. First, we propose an end-to-end framework for each decision making problem. That is the procedure that includes problem formulation, algorithm construction, optimization, and finally selection of an action. In problem formulation step, we employ the multi-objective optimization with randomness (which represents stochastic uncertainty) and fuzziness (which represents subjective uncertainty), as a decision making model. Although, in general, such optimization problems can not be directly optimize due to the uncertainty, we will show that we can use some conventional optimization techniques by reasonable transformation of the problem to deterministic optimization. Then, in algorithm construction step, optimization algorithms are designed for the deterministic optimization problems transformed from the original uncertainty optimization problems. In optimization step, the proposed algorithms run with the interaction to the decision maker and derive a solution that is following to the decision maker's preferences. As the theoretical contribution, we also proved that the optimal solution of the transformed problem has certain Pareto optimality. Furthermore, we demonstrate how the interactive process works through the numerical experiments.

The framework we proposed incorporates the preference structure of the decision maker explicitly into problem and algorithm design. However, in reality in many cases, the decision maker can not explicitly express preferences, that is, preference information can not be used (or can be used only incompletely) in problem formulation and optimization algorithms. Therefore, we also consider a methodology for implementing general optimization under incomplete information. Here, the methodology proposed in this dissertation is not limited to use for the decision making problem, but it can be used for general function optimization. Specifically, we challenge the task of solving the function minimization problem using only the pairwise comparisons (that is, which one of the two points has lower function value) in the function domain. We propose an optimization algorithm of the block coordinate descent method type for the optimization problem based on noisy pairwise comparison. As a theoretical result, the upper bound of the convergence rate of the proposed method is provided under certain conditions. Under certain settings, this upper bound can be shown to be mini-max optimal with respect to the number of pairwise comparison queries. Moreover, as practice, we conduct the numerical experiments to show the actual performance of our proposal.

1 Introduction

1.1 Decision Making as Optimization Problem

Decision making problems that we face in the real world has a very diverse and complicated structure, and the approach to this has been studied in a variety of ways. However, especially in the engineering approach, the decision making can essentially be regarded as the behavior that trying to solve the problem. That is the end-to-end framework that consists of the following procedures

Step 1 Formulating the problem.

Step 2 Searching possible actions.

Step 3 Evaluating actions.

Step 4 Choice of an action.

In this dissertation, we formulate this procedure as mathematical optimization problem. When we say mathematical optimization, there are cases where we refer to the formulation of the problem itself or the case where we refer to the methodology to solve the formulated problem, but we call the framework that includes them the mathematical optimization. The mathematical optimization problem is, in a nutshell, a maximization (or minimization) problem of a function, and first what we must do is design the objective function to be optimized (this is corresponding to Step 1 of the above framework). The objective function needs to be carefully designed so as not to become too complicated while reflecting the characteristics of the actual decision making problem. This is because if the objective function is designed complicated, there is a high possibility that it becomes a problem that can not be solved practically from various viewpoints such as feasibility and calculation cost. In Step 1, there is another important task, that is, to specify the information that can be used to optimize the objective function. Specifically, we have to determine the information that is used in the solution update. For instance, if we can access the Hessian matrix (and its inverse) of the objective function, we can apply the Newton's method, or if we can access the gradient, we can apply some gradient descent type algorithm. As described above, the task in Step 1 is to properly design the objective function and the information that can be used to optimize the objective function. If the problem is formulated, next it is the stage of applying an algorithm to solve it. In the algorithm, the search in Step 2 and the evaluation in Step 3 are executed alternately and repeatedly. Finally, in Step 4, the optimal solution presented by the algorithm is selected as the actual action.

On the one hand, along with the diversification and complexity of society, actual decision-making problems can be formulated as multi-objective decision-making that attempts to derive solutions reflecting the preference structure of decision makers based on multiple evaluation criteria from among multiple alternatives. Such a problem can be formulated mathematically as the multi-objective optimization. As will be described in detail in the next section, the largest difference between the multi-objective case and the single-objective case is in the Step 4, the selection of an action. In the case of the single objective, the algorithm's output can be executed as it is as the next action. However, in the case of multi-objective, generally the output of the algorithm is not unique (that is, there are many candidates of the next action that have same value). Therefore, in Step 4, the decision maker is required to select an action from among

many candidates based on his/her own belief or preference. Therefore, merely formulating the problem is inadequate, and we must reflect the characteristics of the decision maker in problems or algorithms.

1.2 Multi-Objective Optimization

Multi-objective optimization (MOO) problem [41] is formally defined as follows.

Definition 1 (Multi-objective optimization).

$$\min_{x \in X} F(x) = (f_1(x), \dots, f_K(x)) \quad (1)$$

where $X \subset \mathbb{R}^d$ denotes a feasible region of the problem and $f_k : \mathbb{R}^d \rightarrow \mathbb{R}$, $k = 1, \dots, K$ is the objective function.

Multi-objective optimization can be interpreted as a natural generalization of the single objective mathematical optimization problem. However, unlike usual optimization problems with a single objective function, multi-objective optimization is the vector optimization problem and the objective functions, in general, compete with each other. To such problems, it is generally difficult to derive an optimal solution that simultaneously achieves all objectives (which is called as complete optimal). Instead, the concept of Pareto optimality (non-inferior to other feasible solutions) is introduced.

Definition 2 (Pareto optimality). $\mathbf{x}^* \in X$ is said to be a Pareto optimal solution to MOO (1) if and only if there is no $\mathbf{x} \in X$ such that $f_k(\mathbf{x}) \leq f_k(\mathbf{x}^*)$, $k = 1, \dots, K$ with strict inequality holding for at least one k .

By definition, there are in general innumerable Pareto optimal solutions. Figure 1 shows an example of the Pareto optimal and non Pareto optimal solution in the objective function space. It is obvious that if one tries to reduce the value of f_1 , the value of f_2 increases and vice versa. Pareto solutions can thus be interpreted as optimality criteria that balances among objective functions. The curve on which the Pareto solution lies is called Pareto frontier.

From among the infinitely many Pareto solutions, the decision maker has to select one of the most preferable Pareto solution. Introducing the concept of utility function, this problem can be formulated as the following utility maximization problem.

Definition 3 (Utility maximization).

$$\max_{x \in X} U(\mathbf{x}; F) = U(f_1(\mathbf{x}), \dots, f_K(\mathbf{x})) \quad (2)$$

where the utility function (or the scalarization function) $U : \mathbb{R}^K \rightarrow \mathbb{R}$ represents the preference of the decision maker with respect to feasible solution. That is, if $U(\mathbf{x}; F) \geq U(\mathbf{y}; F)$ for some $\mathbf{x}, \mathbf{y} \in X$, \mathbf{x} is more preferable than \mathbf{y} for the decision maker.

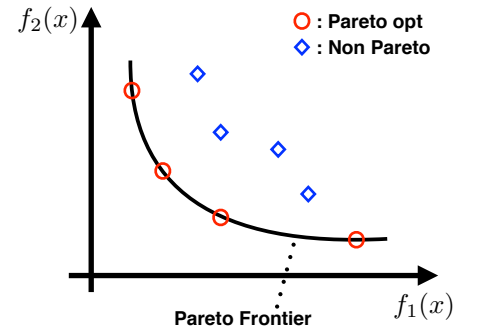


Figure 1: Pareto / non-Pareto points in the objective function space (two-objective case)

	a priori	a posteriori
preference expression	before optimization	after optimization
output	one Pareto solution	a set of Pareto solutions
advantages	single-objective optimization	Pareto solutions can be compared
main issues	eliciting preference	computational cost
general approach	scalarization	genetic algorithm, Bayesian optimization

Table 1: Two categories of the methods that treat the utility maximization problem (2)

Various methods to the MOO problem can be roughly divided into the following three categories according to the approach to the maximization problem of the utility function.

1. A priori preference articulation,
2. A posteriori preference articulation,
3. Interactive method.

A priori preference articulation : If the utility function of the decision maker is fully known or can be modeled in some way, we can directly solve (2). The approaches that formulate the optimization problem based on the preference structure that is pre-expressed by the decision maker and derive only the Pareto solution that satisfies that preference is called *a priori preference articulation*. The main advantage of this approach is that the problem can be reduced to a single-objective optimization. It mean that we can use various method for usual mathematical optimization such as gradient descent, Newton method. Some typical models of the utility function are introduced below.

Example 1 (Weighted sum). *Introducing the weight parameter w_k , $k = 1, \dots, K$ such that $\sum_{k=1}^K w_k = 1$ which represent the importance of each objective and define the utility function as*

$$U(\mathbf{x}; F) = \sum_{k=1}^K w_k f_k(\mathbf{x}). \quad (3)$$

Suppose that f_1, \dots, f_K are convex functions. Then $\mathbf{x}^ \in X$ is the unique optimal solution of (2) if and only if \mathbf{x}^* is a Pareto optimal solution of (1).*

Example 2 (Weighted exponential sum). *Based on the weighted sum utility, we additionally introduce a compensation parameter $p > 0$ and define U as*

$$U(\mathbf{x}; F) = \sum_{k=1}^K w_k f_k(\mathbf{x})^p. \quad (4)$$

If p is sufficiently large, we can claim (without convexity of the objective functions) that $\mathbf{x}^ \in X$ is the unique optimal solution of (2) if and only if \mathbf{x}^* is a Pareto optimal solution of (1).*

Example 3 (Tchebycheff method). *Define U as*

$$U(\mathbf{x}; F) = \left(\sum_{k=1}^K w_k |f_k(\mathbf{x}) - z_k^*|^p \right)^{\frac{1}{p}}. \quad (5)$$

where $z_k^* = \min_{\mathbf{x} \in X} f_k(\mathbf{x})$ is called the ideal point. This utility function is a generalization of the above functions : if $z_k^* = 0$, $k = 1, \dots, K$, (5) coincides (4) and moreover if $p = 1$, it coincides (3). We can claim that if $\mathbf{x}^* \in X$ is Pareto optimal solution then \mathbf{x}^* is an optimal solution of (5).

A posteriori preference articulation : It is sometimes difficult for decision maker to express preferences in advance (i.e. the utility function can not be explicitly written), but it is possible to choose from a set of Pareto optimal solutions. Hence, this approach aims to estimate a set of Pareto optimal solutions (or Pareto frontier). This problem is widely studied not only in the optimization community but also machine learning community and several methods are proposed.

Example 4 (Evolutionary algorithms). *One of the most popular class of the evolutionary algorithm is the genetic algorithm. Genetic algorithm is a multi-point search method and can derive sets of Pareto optimal solutions. Several variation of the genetic algorithm such as VEGA proposed by Schaffer et al. [52], Goldberg's ranking method [20] and MOGA proposed by Fonseca et al. [15] are proposed. A characteristic of this approach is that do not use models for the objective functions. Hence, a large number of function evaluations is required for precise estimation of Pareto sets.*

Example 5 (Bayesian optimization). *Bayesian optimization aims to optimize an (possibly unknown) objective function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ at the number of function evaluation as little as possible. In Bayesian optimization, the objective function is modeled as a sample from a prior stochastic process such as Gaussian process. Then an acquisition function that expresses the trade off between exploration and exploitation is maximized to select the next evaluation point. Finally, the distribution is updated based on the information of the evaluated point. The advantage of this approach is the flexibility in representation of the prior knowledge as well as the ability to guide that where to sample next.*

Bayesian optimization for multi-objective case has been actively studied in the machine learning community very recently. These studies start from the motivation that in the field such as material science, the evaluation of the objective functions requires very high computational cost and one wants to reduce it as well as possible.

Table 1 summarizes characteristics of a priori approach and a posteriori approach.

Interactive methods : Interactive approach is, in a sense, a compromise method of the previous two approaches. A typical method is to parametrically model the utility function and iteratively solve it with varying the parameters, then a set of Pareto solutions is obtained. An example of the interactive method using weighted sum utility function is shown in Algorithm 1. Starting with the proposal of the Geoffrion-Dyer-Feinberg (GDF) method [18], various methods are being studied as interactive method. GDF method employs the approximate marginal rate of substitution derived by the interaction with the decision maker as search direction of more preferred solution.

1.3 Main Contribution

In this dissertation we introduce some frameworks for the design and analysis of decision making problems as mathematical optimization. Our framework includes various algorithms, but they are roughly classified into two major categories according to the decision maker's ability to express his / her preferences.

Algorithm 1 Basic algorithm of interactive method using weighted sum utility

Input: initial model parameter $w_1^{(0)}, \dots, w_K^{(0)}$ such that $\sum_{k=1}^K w_k^{(0)} = 1$. Set $P \leftarrow \{ \}$ that stores Pareto solutions.

for $t = 0, 1, \dots, T$ **do**

Step1 solve the following utility maximization problem

$$\mathbf{x}_t^* = \arg \max_{\mathbf{x} \in X} \sum_{k=1}^K w_k^{(t)} f_k(\mathbf{x})$$

Step2 set $P \leftarrow P \cup \{\mathbf{x}^*\}$ and update $w_k^{(t)}$ to $w_k^{(t+1)}$, $k = 1, \dots, K$ using decision maker's preference.

end for

Output a set of Pareto solutions P .

Case 1 The decision maker can clearly express his / her preference in some way.

- We formulated the decision making problem under uncertainty as
 1. Hierarchical multi-objective stochastic programming,
 2. Hierarchical multi-objective fuzzy random programming,
 3. Multi-objective random fuzzy programmingand proposed an algorithm for each problem that derives a satisfactory solution based on the decision maker's preferences.
- We proved that the solution obtained by the proposed algorithm satisfies Pareto optimality under appropriate conditions.

This problem setting will be discussed in detail in Part I.

Case 2 The decision maker can not express his / her preferences explicitly, but it is possible to answer which of the two proposed solutions is more preferable.

If the decision maker can not explicitly express his / her own preferences, the problem of maximizing utility is formulated as an optimization problem of unknown function. We considered the case that the information feedback from the decision maker is given as the pairwise comparison between two alternatives. Under such a problem formulation,

- We propose a block coordinate descent algorithm based on the pairwise comparison, and point out that the algorithm is easily parallelized.
- We derive an upper bound of the convergence rate in terms of the number of pairwise comparison of function values, i.e., query complexity. Moreover, we show that the query complexity of our algorithm is optimal for specific cases.
- We show that our proposal is more efficient than the state-of-the-art methods using pairwise comparison through numerical experiments.

This problem setting will be discussed in detail in Part II.

1.4 Outline

The dissertation is divided into two main parts, titled “Multi-Objective Optimization for Decision Making under Uncertainty” and “A Novel Algorithm for Optimization with Uncertain Information”. In each part, there are several chapters. Each of the part includes a detailed review of previous work relevant to the specific contents described in the part.

1.4.1 Part I : Multi-Objective Optimization for Decision Making under Uncertainty

In Part I, we formulate various types of multi-objective optimization problems. Then for each of the problems we propose an interactive algorithm to derive a satisfactory solution based on decision maker’s preferences. First, in Chapter 2 we introduce multi-objective optimization under uncertainty and explain about related works. In Chapter 3, we describe about the hierarchical multi-objective stochastic linear programming problem that is, several decision makers try to optimize multiple objective functions respectively. We propose an algorithm to derive solutions that all decision makers obtain a certain amount of satisfaction while reflecting the priority among decision makers. In Chapter 4, we treat the hierarchical multi-objective fuzzy random linear programming problem that includes fuzzy random variable as coefficients of the objective functions. Fuzzy random variable is the function-valued random variable introduced to simultaneously handle stochastic uncertainty and subjective uncertainty. We convert this problem into a deterministic single-objective optimization problem and derive a satisfactory solution for all decision makers under certain priority by an interactive algorithm. In Chapter 5, we discuss about random fuzzy multi-objective linear programming problem which includes random fuzzy variable as coefficients of the objective functions. Although it differs from the general definition, the random fuzzy variable dealt with in this chapter is a random variable whose mean parameter is a function (it is different from fuzzy random variable). We attempt to convert a formally defined random fuzzy multi-objective optimization problem (unfortunately it is ill-defined) into a well-defined problem and to derive a satisfactory solution of the decision maker by an interactive algorithm. Finally in Chapter 6, we summarize Part I.

1.4.2 Part II : A Novel Algorithm for Optimization with Uncertain Information

In Part II, we focus on the general single-objective black-box optimization under the uncertain information feedback. Here, the black box reflects the situation that the utility function of the decision maker is unknown and it is difficult to directly perform operations such as calculation of function values and gradients. First in Chapter 7, we describe the derivative-free optimization, a general formulation one of the black-box optimization, and summarize the related works. In Chapter 8, we briefly explain the problem formulation and some notation that is used in the following chapter and later. In Chapter 9, we propose a block-coordinate descent type algorithm, namely BlockCD, for both deterministic and stochastic pairwise comparison oracle. For the BlockCD algorithm, we provide the theoretical analysis and derive an upper bound of the convergence rate. Chapter 10 examines the empirical performance of the proposed method through numerical experiments. In the experiment, BlockCD algorithm is compared with other state-of-the-art algorithms. Chapter 11 is the summary of Part II.

Part I

Multi-Objective Optimization for Decision Making under Uncertainty

2 Various Formulation of Multi-Objective Optimization under Uncertainty

2.1 Multi-Objective Optimization under Uncertainty

In formulating a multi-objective optimization problem that reflects the actual decision making situation, uncertainties are often included in parameters included in objective functions and constraint expressions. It is important to properly handle such uncertainty and the subjective judgment of the decision makers who implement optimization. Figure 2 shows the relation among some formulation of optimization problem treated in the Part I. From this point of view, two approaches to multi-objective optimization problems under uncertainty have been considered, that is, stochastic programming [5] approach and fuzzy programming approach.

Stochastic programming approach : For the multi-objective stochastic optimization problem in which the parameters included in the objective function and the constraint equation are random variables, Contini [11] firstly proposed a method combining the goal programming and the stochastic programming techniques. Besides this research, methods for deriving minimum risk solutions, interactive methods for deriving compromise solutions, and fuzzy satisfaction methods for deriving solutions maximizing the satisfaction of decision makers have been proposed. In general, in order to deal with a multi-objective optimization problem including random variable coefficients, the decision maker must appropriately set various parameters in advance. For example, when applying a probability maximization model which is one of stochastic programming methods, it is necessary to set a permissible levels for each objective function in advance. Also in another model, namely the maximization of the satisfactory criterion, the tolerance value of the probability that the objective function satisfies a certain criterion must be specified. These parameters have a serious effect on the finally obtained solution, but in this part we will not discuss how to set them appropriately (it will be discussed in the next part).

Fuzzy programming approach : The fuzzy approach to multi-objective optimization stems from the formulation as a fuzzy programming problem by Zimmermann [61]. He proposed a method to derive a satisfactory solution for multi-objective fuzzy programming problems by linear programming based on the fuzzy decisions, under the assumption that decision makers have fuzzy goals for each objective function. Thereafter, Sakawa et al. [50] proposed interactive fuzzy approaches that incorporate the features of the interactive method and the fuzzy programming method at the same time.

Some real data have a mixture of stochastic uncertainty and uncertainty attributed to human subjectivity. To handle two kinds of uncertainty simultaneously, extended random variable such as fuzzy random variable and random fuzzy variable are introduced. Fuzzy random variable

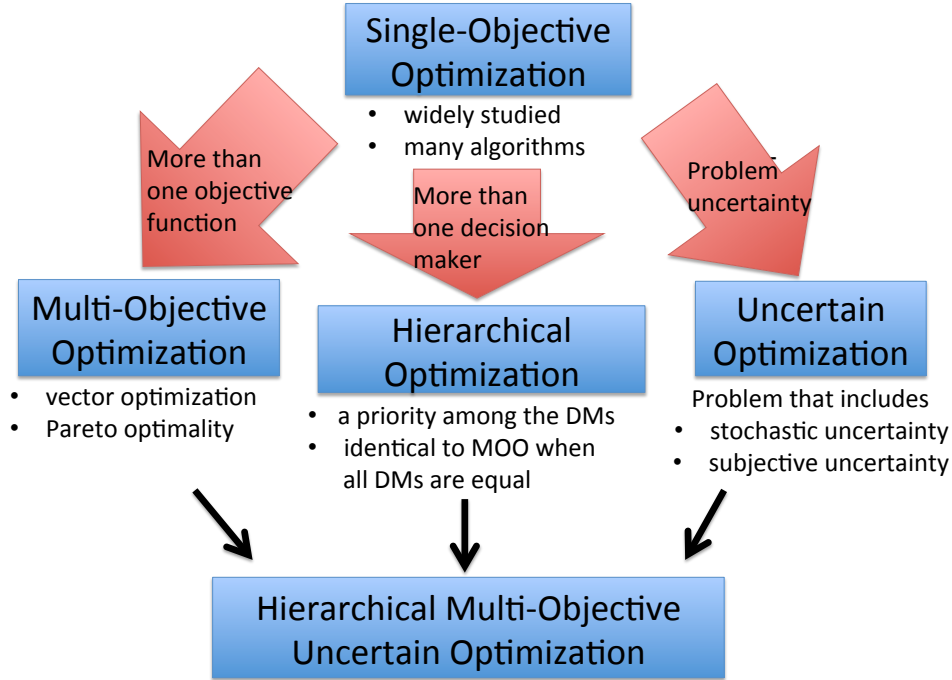


Figure 2: Relation among the classes of optimization problem treated in the Part I

were introduced by Kwakernaak [34] and given mathematical basis such as asymptotic normality by Puri and Ralescu [45]. Intuitively, it can be interpreted as the random variable whose realization is a fuzzy set (mathematically it is a function space-valued random variable). On the other hand, random fuzzy variable [38, 39] is, for example, a random variable whose mean is represented as a fuzzy number. For multi-objective optimization problems where fuzzy random variables are included in the objective functions and constraints, as in the case of random variable coefficients, interactive type methods based on probability maximization model [30, 29] or satisfaction maximization model [31] have been proposed. These methods employ a strategy of converting the fuzzy random multi-objective optimization problem into a normal optimization problem under several assumptions, and deriving an optimal solution. For multi-objective random fuzzy optimization, some interactive algorithm are proposed [32, 28] based on the concept of possibility measure [13].

2.2 Hierarchical Multi-Objective Optimization under Uncertainty

In the real world, the overall decision-making system is often composed of multiple partial systems in a hierarchical structure. Decision makers in each partial system independently act to achieve their goals, but their actions may have some influence on the action of other decision makers. Optimization of such systems is formulated as hierarchical multi-objective programming (or multilevel programming). A typical example of hierarchical multi-objective programming is the Stackelberg game [1, 54]. In this problem, decision makers are uncooperative with each other regarding their decision making, and the goal is to derive an equilibrium solution (Stackelberg equilibrium solution) for all decision makers. On the other hand, in many cases,

decision makers at higher level of the hierarchy aim at optimization considering the satisfaction of lower level decision makers. Liu [36], Shih et al [53] introduced the concept of decision power that represents the priority of higher-level decision makers for such problem setting, and proposed algorithms to derive a solution satisfied by multiple decision makers.

3 Hierarchical Multi-objective Stochastic Linear Optimization Based on the Fuzzy Decision

3.1 Problem Formulation and Corresponding Pareto Optimality

In this section, we consider the following hierarchical multiobjective stochastic linear programming problem (HMOSLP), where each of the decision makers ($DM_r, r = 1, \dots, q$) has his / her own multiple objective linear functions together with common linear constraints, and random variable coefficients are involved in each objective function.

[HMOSLP]

first level decision maker : DM_1

$$\min_{\mathbf{x} \in X} \bar{z}_1(\mathbf{x}) = (\bar{z}_{11}(\mathbf{x}), \dots, \bar{z}_{1k_1}(\mathbf{x})),$$

\vdots

q -th level decision maker : DM_q

$$\min_{\mathbf{x} \in X} \bar{z}_q(\mathbf{x}) = (\bar{z}_{q1}(\mathbf{x}), \dots, \bar{z}_{qk_q}(\mathbf{x})).$$

where

$$X = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

and A is $m \times n$ coefficient matrix. Let us assume that each objective function of $DM_r, r = 1, \dots, q$ is defined by

$$\bar{z}_{r\ell}(\mathbf{x}) = \bar{\mathbf{c}}_{r\ell}^\top \mathbf{x} + \bar{\alpha}_{r\ell}, \ell = 1, \dots, k_r, \quad (6)$$

$$\bar{\mathbf{c}}_{r\ell} = \mathbf{c}_{r\ell}^1 + \bar{t}_{r\ell} \mathbf{c}_{r\ell}^2, \quad (7)$$

$$\bar{\alpha}_{r\ell} = \alpha_{r\ell}^1 + \bar{t}_{r\ell} \alpha_{r\ell}^2, \quad (8)$$

where $\bar{\mathbf{c}}_{r\ell}^1, \bar{\mathbf{c}}_{r\ell}^2 \in \mathbb{R}^n, \ell = 1, \dots, k_r, \bar{\alpha}_{r\ell}^1, \bar{\alpha}_{r\ell}^2 \in \mathbb{R}, \ell = 1, \dots, k_r$ and $\bar{t}_{r\ell}$ is a random variable whose cumulative distribution function $T_{r\ell}(\cdot)$ is assumed to be strictly monotone increasing and continuous.

Similar to the formulations of multilevel linear programming problems proposed by Lee and Shih [37], it is assumed that the upper level decision makers make their decisions with consideration of the overall benefits for the hierarchical organization, although they can take priority for their objective functions over the lower level decision makers.

In order to deal with HMOSLP, we adopt stochastic linear programming techniques for HMOSLP. For the objectives in HMOSLP, we substitute the minimization of $\bar{z}_{r\ell}(\mathbf{x})$ for the maximization of the probability that $\bar{z}_{r\ell}(\mathbf{x})$ is less than or equal to a certain permissible objective level $f_{r\ell}$. Such a probability $p_{r\ell}(\mathbf{x}, f_{r\ell})$ can be defined as follows.

$$p_{r\ell}(\mathbf{x}, f_{r\ell}) = \Pr(\omega \mid \bar{z}_{r\ell}(\mathbf{x}, \omega) \leq f_{r\ell}), r = 1, \dots, q, \ell = 1, \dots, k_r \quad (9)$$

where $\Pr(\cdot)$ denotes a probability measure, ω is an event, and $z_{r\ell}(\mathbf{x}, \omega)$ is a realization of the random objective function $\bar{z}_{r\ell}(\mathbf{x})$ under the occurrence of each elementary event ω . Each of the decision makers ($\text{DM}_r, r = 1, \dots, q$) subjectively specifies certain permissible objective levels:

$$\mathbf{f}_r = (f_{r1}, \dots, f_{rk_r}), r = 1, \dots, q, \quad (10)$$

$$\mathbf{f} = (f_1, \dots, f_q). \quad (11)$$

Then, HMOSLP can be transformed to the following probability maximization model called HMOSLP1(\mathbf{f}).

[HMOSLP1(\mathbf{f})]

first level decision maker : DM_1

$$\max_{\mathbf{x} \in X} p_1(\mathbf{x}, \mathbf{f}_1) = (p_{11}(\mathbf{x}, f_{11}), \dots, p_{1k_1}(\mathbf{x}, f_{1k_1}))$$

\vdots

q -th level decision maker: DM_q

$$\max_{\mathbf{x} \in X} p_q(\mathbf{x}, \mathbf{f}_q) = (p_{q1}(\mathbf{x}, f_{q1}), \dots, p_{qk_q}(\mathbf{x}, f_{qk_q}))$$

Under the assumption that $(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2 > 0, r = 1, \dots, q, \ell = 1, \dots, k_r$, using cumulative distribution function $T_{r\ell}(\cdot)$, the objective function $p_{r\ell}(\mathbf{x}, f_{r\ell})$ in HMOSLP1(\mathbf{f}) is expressed as follows.

$$\begin{aligned} p_{r\ell}(\mathbf{x}, f_{r\ell}) &= \Pr(\omega \mid z_{r\ell}(\mathbf{x}, \omega) \leq f_{r\ell}) \\ &= \Pr(\omega \mid \mathbf{c}_{r\ell}(\omega)^\top \mathbf{x} + \alpha_{r\ell}(\omega) \leq f_{r\ell}) \\ &= \Pr\left(\omega \mid t_{r\ell}(\omega) \leq \frac{f_{r\ell} - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\}}{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2}\right) \\ &= T_{r\ell}\left(\frac{f_{r\ell} - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\}}{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2}\right) \end{aligned}$$

In order to deal with HMOSLP1(\mathbf{f}), we introduce Pareto optimal solution concept called P-Pareto optimal solution.

Definition 4 (P-Pareto optimality). $\mathbf{x}^* \in X$ is said to be a P-Pareto optimal solution to HMOSLP1(\mathbf{f}), if and only if there is no $\mathbf{x} \in X$ such that $p_{r\ell}(\mathbf{x}, f_{r\ell}) \geq p_{r\ell}(\mathbf{x}^*, f_{r\ell}), r = 1, \dots, q, \ell = 1, \dots, k_r$, with strict inequality holding for at least one r and ℓ .

3.2 Deriving a Satisfactory Solution based on the Fuzzy Decision

Sakawa et al. [51] formulated a probability maximization model for multi-objective stochastic programming problems, and proposed an interactive method to obtain a satisfactory solution of the decision maker. In their interactive method, after the decision maker specifies permissible objective levels for each objective function, the candidate of the satisfactory solution is obtained from among M-Pareto optimal solution set. However, in general, the decision maker seems to prefer not only the less value of permissible objective level, but also the larger value of the corresponding probability function. Since these values conflict with each other, the less values of permissible objective level results in the less value of the corresponding probability function. In order to circumvent such difficulties, Yano and Matsui [55] proposed fuzzy approaches

for multi-objective stochastic linear programming problems. From a similar point of view, in this paper, we propose a fuzzy approach for HMOSLP. Considering conflicts between permissible objective levels and the values of the corresponding probability functions in a probability maximization model, the following multi-objective programming problem can be regarded as a natural extension of HMOSLP1(f).

[HMOSLP2]

first level decision maker : \mathbf{DM}_1

$$\max_{\mathbf{x} \in X, f_1 \in \mathbb{R}^{k_1}} (p_{11}(\mathbf{x}, f_{11}), \dots, p_{1k_1}(\mathbf{x}, f_{1k_1}), -f_{11}, \dots, -f_{1k_1})$$

\vdots

q -th level decision maker: \mathbf{DM}_q

$$\max_{\mathbf{x} \in X, f_q \in \mathbb{R}^{k_q}} (p_{q1}(\mathbf{x}, f_{q1}), \dots, p_{qk_q}(\mathbf{x}, f_{qk_q}) - f_{q1}, \dots, -f_{qk_q})$$

It should be noted here that a permissible objective level $f_{r\ell}$ is not a constant value but a decision variable.

Considering the imprecise nature of the decision maker's judgment, it is natural to assume that the decision maker have a fuzzy goal for each objective function in HMOSLP2. In this section, it is assumed that such fuzzy goals can be quantified by eliciting the corresponding membership functions. Let us denote a membership function of probability function $p_{r\ell}(\mathbf{x}, f_{r\ell})$ as $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$, and a membership function of permissible objective level $f_{r\ell}$ as $\mu_{f_{r\ell}}(f_{r\ell})$ respectively. Then, HMOSLP2 can be transformed to the following multiobjective programming problem.

[HMOSLP3]

first level decision maker : \mathbf{DM}_1

$$\max_{\mathbf{x} \in X, f_1 \in \mathbb{R}^{k_1}} (\mu_{p_{11}}(p_{11}(\mathbf{x}, f_{11})), \dots, \mu_{p_{1k_1}}(p_{1k_1}(\mathbf{x}, f_{1k_1})), \mu_{f_{11}}(f_{11}), \dots, \mu_{f_{1k_1}}(f_{1k_1}))$$

\vdots

q -th level decision maker: \mathbf{DM}_q

$$\max_{\mathbf{x} \in X, f_q \in \mathbb{R}^{k_q}} (\mu_{p_{q1}}(p_{q1}(\mathbf{x}, f_{q1})), \dots, \mu_{p_{qk_q}}(p_{qk_q}(\mathbf{x}, f_{qk_q})), \mu_{f_{q1}}(f_{q1}), \dots, \mu_{f_{qk_q}}(f_{qk_q}))$$

Throughout this section, we make the assumptions that $\mu_{f_{r\ell}}(f_{r\ell})$, $r = 1, \dots, q$, $\ell = 1, \dots, k_r$ are strictly monotone decreasing and continuous with respect to $f_{r\ell}$, and $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$, $r = 1, \dots, q$, $\ell = 1, \dots, k_r$ are strictly monotone increasing and continuous with respect to $p_{r\ell}(\mathbf{x}, f_{r\ell})$.

For example, we can define the interval for $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$ as follows. Considering the individual minimum and maximum of the expectation, $\mathbb{E}(\bar{z}_{r\ell}(\mathbf{x}))$, the decision maker (\mathbf{DM}_r) subjectively specifies a sufficiently satisfactory maximum value $f_{r\ell\min}$ and an unacceptable minimum value $f_{r\ell\max}$. Then, the interval for $\mu_{f_{r\ell}}(f_{r\ell})$ is defined as:

$$F_{r\ell} = [f_{r\ell}^{\min}, f_{r\ell}^{\max}]. \quad (12)$$

Corresponding to the interval $F_{r\ell}$, denote the interval for $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$ as:

$$P_{r\ell}(F_{r\ell}) = [p_{r\ell}^{\min}, p_{r\ell}^{\max}], \quad (13)$$

where $p_{r\ell}^{\min}$ is an unacceptable maximum value and $p_{r\ell}^{\max}$ is a sufficiently satisfactory minimum value.

$p_{r\ell}^{\max}$ can be obtained by solving the following optimization problem.

$$p_{r\ell}^{\max} = \max_{\mathbf{x} \in X} p_{r\ell}(\mathbf{x}, f_{r\ell}^{\max}). \quad (14)$$

It should be noted here that the above problem is equivalent to the following linear fractional programming problem [8] because distribution function $T(\cdot)$ is strictly monotone increasing and continuous.

$$\max_{\mathbf{x} \in X} \left(\frac{f_{r\ell}^{\max} - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\}}{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2} \right). \quad (15)$$

On the other hand, in order to obtain $p_{r\ell}^{\min}$, we first solve

$$\max_{\mathbf{x} \in X} p_{r\ell}(\mathbf{x}, f_{r\ell}^{\min}), \quad (16)$$

and denote the corresponding optimal solution as $\mathbf{x}_{r\ell}$. Using the optimal solutions $\mathbf{x}_{r\ell}$, $r = 1, \dots, q$, $\ell = 1, \dots, k_r$, $p_{r\ell}^{\min}$ can be obtained as follows.

$$p_{r\ell}^{\min} = \min_{s=1, \dots, k_s, s \neq \ell} p_{r\ell}(\mathbf{x}_{rs}, f_{r\ell}^{\min}). \quad (17)$$

In order to integrate the membership functions in HMOSLP3, let us assume that the decision makers adopt the fuzzy decision as an aggregation operator. That is, the integrated membership function is defined as

$$\min \{ \mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})), \mu_{f_{r\ell}}(f_{r\ell}) \}. \quad (18)$$

Then a satisfactory solution is obtained by solving the following max-min problem.

[MAXMIN1]

$$\max_{\mathbf{x} \in X, f_{r\ell} \in F_{r\ell}, \lambda \in [0,1]} \lambda \quad (19)$$

subject to

$$\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})) \geq \lambda, \quad r = 1, \dots, q, \ell = 1, \dots, k_r \quad (20)$$

$$\mu_{f_{r\ell}}(f_{r\ell}) \geq \lambda, \quad r = 1, \dots, q, \ell = 1, \dots, k_r \quad (21)$$

It should be noted here that, in general, the optimal solution of MAXMIN1 does not reflect the hierarchical structure between q decision makers where the upper level decision maker can take priority for his / her cumulative distribution functions over the lower level decision makers. In order to cope with such a hierarchical preference structure between q decision makers in MAXMIN1, we introduce the concept of the decision power [36] :

$$\mathbf{w} = (w_1, \dots, w_q) \quad (22)$$

for the inequality constraints (20) and (21) in MAXMIN1, where the r -th level decision maker (DM_r) can specify the decision power w_{r+1} in his / her subjective manner and the last decision maker (DM_q) has no decision power. In order to reflect the hierarchical preference structure

between multiple decision makers, the decision powers $\mathbf{w} = (w_1, \dots, w_q)^T$ have to satisfy the following inequality condition.

$$w_1 = 1 \geq w_2 \geq \dots \geq w_{q-1} \geq w_q > 0 \quad (23)$$

Then, the corresponding modified MAXMIN1 is reformulated as follows.

[MAXMIN2(\mathbf{w})]

$$\max_{\mathbf{x} \in X, f_{r\ell} \in F_{r\ell}, \lambda \in [0,1]} \lambda \quad (24)$$

subject to

$$\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})) \geq \lambda w_r, \quad r = 1, \dots, q, \ell = 1, \dots, k_r \quad (25)$$

$$\mu_{f_{r\ell}}(f_{r\ell}) \geq \lambda w_r, \quad r = 1, \dots, q, \ell = 1, \dots, k_r \quad (26)$$

Since $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$ is strictly monotone increasing and continuous and $\mathbf{c}_{r\ell}^2 \mathbf{x} + \alpha_{r\ell}^2 > 0$, the constraints (25) can be transformed as follows.

$$\begin{aligned} & \mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})) \geq \lambda w_r, \\ \Leftrightarrow & p_{r\ell}(\mathbf{x}, f_{r\ell}) \geq \mu_{p_{r\ell}}^{-1}(\lambda w_r), \\ \Leftrightarrow & T_{r\ell} \left(\frac{f_{r\ell} - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\}}{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2} \right) \geq \mu_{p_{r\ell}}^{-1}(\lambda w_r), \\ \Leftrightarrow & f_{r\ell} - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\} \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2\}, \end{aligned} \quad (27)$$

where $\mu_{p_{r\ell}}^{-1}(\cdot)$ and $T_{r\ell}^{-1}(\cdot)$ are pseudo-inverse functions with respect to $\mu_{p_{r\ell}}(\cdot)$ and $T_{r\ell}(\cdot)$ respectively. Moreover, it holds that $f_{r\ell} \leq \mu_{f_{r\ell}}^{-1}(\lambda w_r)$, because $\mu_{f_{r\ell}}(f_{r\ell})$ is strictly monotone decreasing and continuous and the constraints (26). As a result, the constraint (27) can be reduced to the following inequality where a permissible objective level $f_{r\ell}$ is removed.

$$\mu_{f_{r\ell}}^{-1}(\lambda w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\} \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2\} \quad (28)$$

Then, MAXMIN2(\mathbf{w}) is equivalently transformed to the following problem.

[MAXMIN3(\mathbf{w})]

$$\max_{\mathbf{x} \in X, \lambda \in [0,1]} \lambda \quad (29)$$

subject to

$$\begin{aligned} \mu_{f_{r\ell}}^{-1}(\lambda w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\} & \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2\}, \\ & r = 1, \dots, q, \ell = 1, \dots, k_r \end{aligned} \quad (30)$$

It should be noted here that the constraints (30) can be reduced to a set of linear inequalities for some fixed value $\lambda \in [0, 1]$. This means that an optimal solution $(\mathbf{x}^*, \lambda^*)$ of MAXMIN3(\mathbf{w}) is obtained by combined use of the bisection method with respect to λ and the first-phase of the two-phase simplex method of linear programming.

The relationship between the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MAXMIN3(\mathbf{w}) and P-Pareto optimal solutions can be characterized by the following main theorem.

Theorem 1 (Yano and Matsui [56]). *If $(\mathbf{x}^*, \lambda^*)$ is a unique optimal solution of MAXMIN3(\mathbf{w}), then $\mathbf{x}^* \in X$ is a P-Pareto optimal solution to HMOSLP1(\mathbf{f}^*), where*

$$\mathbf{f}^* = (\mu_{f_{11}}^{-1}(\lambda^* w_1), \dots, \mu_{f_{1k_1}}^{-1}(\lambda^* w_1), \dots, \mu_{f_{q1}}^{-1}(\lambda^* w_q), \dots, \mu_{f_{qk_q}}^{-1}(\lambda^* w_q)).$$

Proof. Since an optimal solution $(\mathbf{x}^*, \lambda^*)$ satisfies the constraints (30), it holds that

$$\begin{aligned} & \mu_{f_{r\ell}}^{-1}(\lambda^* w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x}^* + \alpha_{r\ell}^1\} \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda^* w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x}^* + \alpha_{r\ell}^2\}, \\ \Leftrightarrow & T_{r\ell} \left(\frac{\mu_{f_{r\ell}}^{-1}(\lambda^* w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x}^* + \alpha_{r\ell}^1\}}{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x}^* + \alpha_{r\ell}^2} \right) = p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r)) \\ & \geq \mu_{p_{r\ell}}^{-1}(\lambda^* w_r). \end{aligned}$$

Assume that $\mathbf{x}^* \in X$ is not a P-Pareto optimal solution to HMOSLP1(\mathbf{f}^*), where

$$\mathbf{f}^* = (\mu_{f_{11}}^{-1}(\lambda^* w_1), \dots, \mu_{f_{1k_1}}^{-1}(\lambda^* w_1), \dots, \mu_{f_{q1}}^{-1}(\lambda^* w_q), \dots, \mu_{f_{qk_q}}^{-1}(\lambda^* w_q)),$$

then there exists $\mathbf{x} \in X$ such that

$$p_{r\ell}(\mathbf{x}, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r)) \geq p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r)) \geq \mu_{p_{r\ell}}^{-1}(\lambda^* w_r),$$

with strict inequality holding for at least one r and ℓ . Then there exists $\mathbf{x} \in X$ such that

$$\mu_{f_{r\ell}}^{-1}(\lambda^* w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\} \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda^* w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2\},$$

which contradicts the fact that $(\mathbf{x}^*, \lambda^*)$ is a unique optimal solution of MAXMIN3(\mathbf{w}). \square

After calculating the optimal value λ^* of MAXMIN3(\mathbf{w}) on the basis of linear programming, one of the corresponding optimal solutions \mathbf{x}^* can be obtained by solving the following linear fractional programming problem.

[LFP(\mathbf{w}, λ^*)]

$$\min_{\mathbf{x} \in X} - \left(\frac{\mu_{f_{11}}^{-1}(\lambda^* w_1) - \{(\mathbf{c}_{11}^1)^\top \mathbf{x} + \alpha_{11}^1\}}{(\mathbf{c}_{11}^2)^\top \mathbf{x} + \alpha_{11}^2} \right)$$

subject to

$$\begin{aligned} \mu_{f_{r\ell}}^{-1}(\lambda^* w_r) - \{(\mathbf{c}_{r\ell}^1)^\top \mathbf{x} + \alpha_{r\ell}^1\} & \geq T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda^* w_r)) \times \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{x} + \alpha_{r\ell}^2\}, \\ r &= 1, \dots, q, \ell = 1, \dots, k_r, \\ (r, \ell) &\neq (1, 1) \end{aligned} \tag{31}$$

Let us consider the transformation of variables called the Charnes-Cooper transformation [8] as follows.

$$v = \frac{1}{(\mathbf{c}_{11}^2)^\top \mathbf{x} + \alpha_{11}^2}, \quad \mathbf{y} = v\mathbf{x}, \quad v > 0. \tag{32}$$

Then, LFP(\mathbf{w}, λ^*) can be transformed to the linear programming problem.

[LP(\mathbf{w}, λ^*)]

$$\min_{\mathbf{y} \geq \mathbf{0}, v > \delta} (\mathbf{c}_{11}^1)^\top \mathbf{y} + (\alpha_{11}^1 - \mu_{f_{11}}^{-1}(\lambda^* w_1))v$$

subject to

Algorithm 2 Interactive algorithm for HMOSLP

Initialize: Set the initial decision powers as $w_r = 1, r = 1, \dots, q$ and $t = 0$.

repeat

Step 1: Solve the following individual optimization for each objective.

$$\begin{aligned} \mathbf{x}_{\min}^* &= \arg \min_{\mathbf{x} \in X} \mathbb{E}(\bar{z}_{r\ell}(\mathbf{x})), \\ \mathbf{x}_{\max}^* &= \arg \max_{\mathbf{x} \in X} \mathbb{E}(\bar{z}_{r\ell}(\mathbf{x})). \end{aligned}$$

Then, each decision maker (DM_r) subjectively specifies a sufficiently satisfactory maximum value $f_{r\ell}^{\min}$ and an unacceptable minimum value $f_{r\ell}^{\max}$ and set his / her membership functions $\mu_{f_{r\ell}}(f_{r\ell}), \ell = 1, \dots, k_r$ on the interval $F_{r\ell} = [f_{r\ell}^{\min}, f_{r\ell}^{\max}]$.

Step 2: Compute an unacceptable maximum value $p_{r\ell}^{\min}$ and a sufficiently satisfactory minimum value $p_{r\ell}^{\max}$ by solving (14) and (17). Then, DM_r sets his / her membership functions $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})), \ell = 1, \dots, k_r$ on the interval $P_{r\ell}(F_{r\ell}) = [p_{r\ell}^{\min}, p_{r\ell}^{\max}]$,

Step 3: Solve MAXMIN3(\mathbf{w}) using the bisection method and the phase one of the two-phase simplex method to obtain the optimal value λ^* .

Step 4: Solve LP(\mathbf{w}, λ^*) and obtain the corresponding optimal solution (\mathbf{y}^*, v^*) .

Step 5: Let s be the index that DM_s be the uppermost of the decision makers who are not satisfied with the current values of his / her membership functions. Then, DM_s updates the decision power w_{s+1} according to the following Rule 1, and return to Step 3.

until $\text{DM}_r, r = 1, \dots, q - 1$ is satisfied with the values of membership functions

Output: \mathbf{x}_t

$$\begin{aligned} T_{r\ell}^{-1}(\mu_{p_{r\ell}}^{-1}(\lambda^* w_r)) \cdot \{(\mathbf{c}_{r\ell}^2)^\top \mathbf{y} + v \alpha_{r\ell}^2\} + (\mathbf{c}_{r\ell}^1)^\top \mathbf{y} + (\alpha_{r\ell}^1 - \mu_{f_{r\ell}}^{-1}(\lambda^* w_r))v &\leq 0, \\ r = 1, \dots, q, \ell = 1, \dots, k_r, (r, \ell) &\neq (1, 1) \\ A\mathbf{y} - v\mathbf{b} &\leq \mathbf{0} \end{aligned}$$

where $\delta > 0$ is a sufficiently small and positive constant number.

Strictly speaking, in order to guarantee that the optimal solution (\mathbf{y}^*, s^*) to LP(\mathbf{w}, λ^*) is unique, Pareto optimality test [51] should be done.

3.3 An Interactive Algorithm

After obtaining the optimal solution $(\mathbf{x}^*, \lambda^*)$ by solving MAXMIN3(\mathbf{w}) on the basis of linear programming, each decision maker (DM_r) must either be satisfied with the current values of his / her membership functions

$$\begin{aligned} \mu_{f_{r\ell}}(\mu_{f_{r\ell}}^{-1}(\lambda^* w_r)), \\ \mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r))), \end{aligned}$$

or update his / her decision power w_{r+1} .

The proposed interactive algorithm to derive the satisfactory solution of multiple decision makers ($\text{DM}_r, r = 1, \dots, q$) in a hierarchical organization from among P-Pareto optimal solution set for HMOSLP1(\mathbf{f}) is shown in Algorithm 2.

Rule 1 (Update the decision power). *When the decision maker (DM_s) updates his / her decision power w_{s+1} , $w_{s+1} \leq w_s$ must be satisfied in order to guarantee the inequality conditions (23). After updating w_{s+1} , if there exists some index $t > s + 1$ such that $w_{s+1} < w_t$, then the corresponding decision power w_t is set as $w_t \leftarrow w_{s+1}$.*

It should be noted here that, when the decision maker (DM_s) updates his / her decision power w_{s+1} according to the above rule at Step 6, the membership functions $\mu_{f_{r\ell}}(\mu_{f_{r\ell}}^{-1}(\lambda^* w_r))$ and $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r)))$ will be improved by the less value of the decision power w_{s+1} at the expense of the other membership functions $\mu_{f_{r\ell}}(\mu_{f_{r\ell}}^{-1}(\lambda^* w_r))$ and $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r)))$.

3.4 A Numerical Experiment

In order to demonstrate the proposed method and the interactive processes, we consider the following hierarchical two-objective stochastic linear programming problem under three hypothetical decision makers.

[HMOSLP]

r -th level decision maker : DM_r , $r = 1, 2, 3$.

$$\min \bar{z}_{r1}(\mathbf{x}) = (\mathbf{c}_{r1}^1 + \bar{t}_{r1} \mathbf{c}_{r1}^2)^\top \mathbf{x} + (\alpha_{r1}^1 + \bar{t}_{r1} \alpha_{r1}^2)$$

$$\min \bar{z}_{r2}(\mathbf{x}) = (\mathbf{c}_{r2}^1 + \bar{t}_{r2} \mathbf{c}_{r2}^2)^\top \mathbf{x} + (\alpha_{r2}^1 + \bar{t}_{r2} \alpha_{r2}^2)$$

subject to $\mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{10} \mid \mathbf{a}_i \mathbf{x} \leq b_i, i = 1, \dots, 7, \mathbf{x} \geq \mathbf{0}\}$

In the above HMOSLP, $\mathbf{x} = (x_1, x_2, \dots, x_{10})^\top$ is the decision vector, \mathbf{a}_i , $i = 1, \dots, 7$, $\mathbf{c}_{r\ell}^1$, $\mathbf{c}_{r\ell}^2$, $r = 1, 2, 3$, $\ell = 1, 2$ are the constant coefficient vectors which are shown in Table 1, and $\alpha_{11}^1 = -18$, $\alpha_{11}^2 = 5$, $\alpha_{12}^1 = -27$, $\alpha_{12}^2 = 6$, $\alpha_{21}^1 = -12$, $\alpha_{21}^2 = 3$, $\alpha_{22}^1 = -15$, $\alpha_{22}^2 = 4$, $\alpha_{31}^1 = -10$, $\alpha_{31}^2 = 4$, $\alpha_{32}^1 = -27$, $\alpha_{32}^2 = 6$. The right side of the constraints are $b_1 = 140$, $b_2 = -220$, $b_3 = -190$, $b_4 = 75$, $b_5 = -160$, $b_6 = 130$, $b_7 = 90$. And $t_{r\ell}(\omega)$, $r = 1, 2, 3$, $\ell = 1, 2$ are Gaussian random variables defined as $\bar{t}_{11} \sim N(4, 2^2)$, $\bar{t}_{12} \sim N(3, 3^2)$, $\bar{t}_{21} \sim N(3, 1^2)$, $\bar{t}_{22} \sim N(3, 2^2)$, $\bar{t}_{31} \sim N(3, 2^2)$, $\bar{t}_{32} \sim N(3, 3^2)$.

According to the proposed interactive algorithm, the initial decision powers are fixed as $(w_1, w_2, w_3) = (1, 1, 1)$.

At Step 1, the individual minimum and maximum of $\mathbb{E}(\bar{z}_{r\ell}(\mathbf{x}))$, $r = 1, 2, 3$, $\ell = 1, 2$ are calculated. Considering such values, each decision maker (DM_r) specifies the intervals $F_{r\ell} = [f_{r\ell}^{\min}, f_{r\ell}^{\max}]$, as $F_{11} = [2000, 2200]$, $F_{12} = [400, 700]$, $F_{21} = [800, 1000]$, $F_{22} = [650, 800]$, $F_{31} = [-1050, -950]$, $F_{32} = [-200, -50]$. On these intervals, each hypothetical decision maker sets his / her membership functions $\mu_{f_{r\ell}}(f_{r\ell})$ as:

$$\mu_{f_{r\ell}}(f_{r\ell}) = \frac{f_{r\ell} - f_{r\ell}^{\max}}{f_{r\ell}^{\min} - f_{r\ell}^{\max}},$$

where $f_{r\ell} \in F_{r\ell}$.

At Step 2, Corresponding to the interval $F_{r\ell}$, the intervals $P_{r\ell} = P_{r\ell}(F_{r\ell}) = [p_{r\ell}^{\min}, p_{r\ell}^{\max}]$ are computed as $P_{11} = [0.023, 0.959]$, $P_{12} = [0.015, 0.993]$, $P_{21} = [0.001, 0.999]$, $P_{22} = [0.259, 0.995]$, $P_{31} = [0.136, 0.859]$, $P_{32} = [0.001, 0.987]$. On these intervals, each hypothetical decision maker sets his / her membership functions $\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell}))$ as :

$$\mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}, f_{r\ell})) = \frac{p_{r\ell}^{\min} - p_{r\ell}(\mathbf{x}, f_{r\ell})}{p_{r\ell}^{\min} - p_{r\ell}^{\max}}, \quad (33)$$

Table 1. Constant coefficients

\mathbf{x}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
\mathbf{c}_{11}^1	19	48	21	10	18	35	46	11	24	33
\mathbf{c}_{11}^2	3	2	2	1	4	3	1	2	4	2
\mathbf{c}_{12}^1	12	-46	-23	-38	-33	-48	12	8	19	20
\mathbf{c}_{12}^2	1	2	4	2	2	1	2	1	2	1
\mathbf{c}_{21}^1	12	38	-23	33	-33	45	12	-9	19	20
\mathbf{c}_{21}^2	1	2	4	2	2	1	2	1	2	1
\mathbf{c}_{22}^1	12	-36	27	-30	-33	45	-11	12	19	-8
\mathbf{c}_{22}^2	1	2	4	2	2	1	2	1	2	1
\mathbf{c}_{31}^1	-18	-26	-22	-28	-15	-29	-10	-19	-17	-28
\mathbf{c}_{31}^2	2	1	3	2	1	2	3	3	2	1
\mathbf{c}_{32}^1	-8	31	28	29	25	36	-8	-7	-13	-15
\mathbf{c}_{32}^2	1	2	3	2	2	1	2	1	2	1
\mathbf{a}_1	12	-2	4	-7	13	-1	-6	6	11	-8
\mathbf{a}_2	-2	5	3	16	6	-12	12	4	-7	-10
\mathbf{a}_3	3	-16	-4	-8	-8	2	-12	-12	4	-3
\mathbf{a}_4	-11	6	-5	9	-1	8	-4	6	-9	6
\mathbf{a}_5	-4	7	-6	-5	13	6	-2	-5	14	-6
\mathbf{a}_6	5	-3	14	-3	-9	-7	4	-4	-5	9
\mathbf{a}_7	-3	-4	-6	9	6	18	11	-9	-4	7

where $p_{r\ell}(\mathbf{x}, f_{r\ell}) \in P_{r\ell}(F_{r\ell})$.

At Step 3 and 4, solve MAXMIN3(\mathbf{w}), and obtain the optimal solution as :

$$\mu_{f_{r\ell}}(\mu_{f_{r\ell}}^{-1}(\lambda^* w_r)) = \mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r))) = 0.545, \quad r = 1, 2, 3, \quad \ell = 1, 2.$$

For the optimal solution $(\mathbf{x}^*, \lambda^*)$, DM₁ updates his / her decision power as $w_2 = 0.8$ in order to improve his / her own membership functions, and go back to Step 3. Then, the optimal solution is obtained as :

$$\mu_{f_{1\ell}}(\mu_{f_{1\ell}}^{-1}(\lambda^* w_1)) = \mu_{p_{1\ell}}(p_{1\ell}(\mathbf{x}^*, \mu_{f_{1\ell}}^{-1}(\lambda^* w_1))) = 0.621, \quad \ell = 1, 2,$$

$$\mu_{f_{r\ell}}(\mu_{f_{r\ell}}^{-1}(\lambda^* w_r)) = \mu_{p_{r\ell}}(p_{r\ell}(\mathbf{x}^*, \mu_{f_{r\ell}}^{-1}(\lambda^* w_r))) = 0.497, \quad r = 2, 3, \quad \ell = 1, 2.$$

For this optimal solution, DM₁ is satisfied with the current value of the cumulative membership functions, but DM₂ is not satisfied with the current values. Therefore, DM₂ updates his / her decision power as $w_3 = 0.6$ and the optimal solution is obtained as :

$$\mu_{f_{1\ell}}(\mu_{f_{1\ell}}^{-1}(\lambda^* w_1)) = \mu_{p_{1\ell}}(p_{1\ell}(\mathbf{x}^*, \mu_{f_{1\ell}}^{-1}(\lambda^* w_1))) = 0.715, \quad \ell = 1, 2,$$

$$\mu_{f_{2\ell}}(\mu_{f_{2\ell}}^{-1}(\lambda^* w_2)) = \mu_{p_{2\ell}}(p_{2\ell}(\mathbf{x}^*, \mu_{f_{2\ell}}^{-1}(\lambda^* w_2))) = 0.572, \quad \ell = 1, 2,$$

$$\mu_{f_{3\ell}}(\mu_{f_{3\ell}}^{-1}(\lambda^* w_3)) = \mu_{p_{3\ell}}(p_{3\ell}(\mathbf{x}^*, \mu_{f_{3\ell}}^{-1}(\lambda^* w_3))) = 0.429, \quad \ell = 1, 2.$$

Since hypothetical decision makers (DM₁ and DM₂) are satisfied with current values of the above membership functions, stop the interactive processes.

4 Satisfaction maximization model for Hierarchical Multi-objective Fuzzy Random Linear Programming Problems

4.1 Problem Formulation

In this section, we focus on a hierarchical multi-objective programming problem involving fuzzy random variable coefficients in objective functions, which is called a hierarchical multi-objective fuzzy random linear programming problem (HMOFRLP).

[HMOFRLP]

first level decision maker : DM₁

$$\min_{\mathbf{x} \in X} \tilde{\mathbf{C}}\mathbf{x} = \left(\tilde{\mathbf{c}}_{11}^\top \mathbf{x}, \dots, \tilde{\mathbf{c}}_{1k_1}^\top \mathbf{x} \right) \quad (34)$$

⋮

***q*-th level decision maker : DM_{*q*}**

$$\min_{\mathbf{x} \in X} \tilde{\mathbf{C}}\mathbf{x} = \left(\tilde{\mathbf{c}}_{q1}^\top \mathbf{x}, \dots, \tilde{\mathbf{c}}_{qk_q}^\top \mathbf{x} \right)$$

where $X = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, $\mathbf{b} \in \mathbb{R}^m$ and A is an $m \times n$ coefficient matrix. $\tilde{\mathbf{c}}_{ri} = (\tilde{c}_{ri1}, \dots, \tilde{c}_{rin})$, $i = 1, \dots, k_r$, $r = 1, \dots, q$, are coefficient vectors of objective function $\tilde{\mathbf{c}}_{ri}^\top \mathbf{x}$, whose elements are fuzzy random variables [34, 45, 51], and the symbols "-" and "~" mean randomness and fuzziness respectively.

In order to deal with the objective functions $\tilde{\mathbf{c}}_{ri}\mathbf{x}$, $i = 1, \dots, k_r$, $r = 1, \dots, q$, Katagiri et al. [26, 30], proposed an LR-type fuzzy random variable which can be regarded as a special version of a fuzzy random variable. Under the occurrence of each elementary event ω , $\tilde{c}_{rij}(\omega)$ is a realization of an LR-type fuzzy random variable \tilde{c}_{rij} , which is an LR fuzzy number [13] whose membership function is defined as follows.

$$\mu_{\tilde{c}_{rij}(\omega)}(s) = \begin{cases} L\left(\frac{\bar{d}_{rij}(\omega) - s}{\bar{\alpha}_{rij}(\omega)}\right) & (s \leq \bar{d}_{rij}(\omega) \forall \omega), \\ R\left(\frac{s - \bar{d}_{rij}(\omega)}{\bar{\beta}_{rij}(\omega)}\right) & (s > \bar{d}_{rij}(\omega) \forall \omega), \end{cases}$$

where the function $L(t) = \max\{0, l(t)\}$ is a real-valued continuous function from $[0, \infty)$ to $[0, 1]$, and $l(t)$ is a strictly decreasing continuous function satisfying $l(0) = 1$. Also, $R(t) = \max\{0, r(t)\}$ satisfies the same conditions. $\bar{d}_{rij}, \bar{\alpha}_{rij}, \bar{\beta}_{rij}$ are random variables expressed by $\bar{d}_{rij} = d_{rij}^1 + \bar{t}_{ri}d_{rij}^2$, $\bar{\alpha}_{rij} = \alpha_{rij}^1 + \bar{t}_{ri}\alpha_{rij}^2$ and $\bar{\beta}_{rij} = \beta_{rij}^1 + \bar{t}_{ri}\beta_{rij}^2$. \bar{t}_{ri} is a random variable whose distribution function is denoted by $T_{ri}(\cdot)$ which is strictly increasing and continuous, and $d_{rij}^1, d_{rij}^2, \alpha_{rij}^1, \alpha_{rij}^2, \beta_{rij}^1, \beta_{rij}^2$ are constants.

Similar to Katagiri et al. [26, 30], HMOFRLP can be transformed into a hierarchical multi-objective stochastic programming problem (HMOSP) by using a concept of a possibility measure [13]. As shown in [30], the realizations $\tilde{\mathbf{c}}_{ri}(\omega)\mathbf{x}$ becomes an LR fuzzy number characterized by the following membership functions on the basis of the extension principle [13].

$$\mu_{\tilde{\mathbf{c}}_{ri}(\omega)^\top \mathbf{x}}(y) = \begin{cases} L\left(\frac{\bar{d}_{ri}(\omega)^\top \mathbf{x} - y}{\bar{\alpha}_{ri}(\omega)^\top \mathbf{x}}\right) & y \leq \bar{d}_{ri}(\omega)^\top \mathbf{x} \\ R\left(\frac{y - \bar{d}_{ri}(\omega)^\top \mathbf{x}}{\bar{\beta}_{ri}(\omega)^\top \mathbf{x}}\right) & y > \bar{d}_{ri}(\omega)^\top \mathbf{x} \end{cases}$$

For the realizations $\tilde{\mathbf{c}}_{ri}(\omega)^\top \mathbf{x}$, $i = 1, \dots, k_r$, $r = 1, \dots, q$, it is assumed that the decision maker has fuzzy goals \tilde{G}_{ri} , $i = 1, \dots, k_r$, $r = 1, \dots, q$ [50], whose membership functions $\mu_{\tilde{G}_{ri}}(y)$,

$i = 1, \dots, k_r$, $r = 1, \dots, q$ are continuous and strictly decreasing for minimization problems. By using a concept of a possibility measure [13], a degree of possibility [25] that the objective function value $\tilde{c}_{ri}x$ satisfies the fuzzy goal \tilde{G}_{ri} is expressed as follows.

$$\Pi_{\tilde{c}_{ri}^\top x}(\tilde{G}_{ri}) = \sup_y \min \{ \mu_{\tilde{c}_{ri}^\top x}(y), \mu_{\tilde{G}_{ri}}(y) \} \quad (35)$$

Using a possibility measure, HMOFRLP can be transformed into the following hierarchical multiobjective stochastic programming problem (HMOSP).

[HMOSP]

first level decision maker : DM₁

$$\max_{x \in X} \left(\Pi_{\tilde{c}_{11}^\top x}(\tilde{G}_{11}), \dots, \Pi_{\tilde{c}_{1k_1}^\top x}(\tilde{G}_{1k_1}) \right)$$

\vdots

q -th level decision maker: DM _{q}

$$\max_{x \in X} \left(\Pi_{\tilde{c}_{q1}^\top x}(\tilde{G}_{q1}), \dots, \Pi_{\tilde{c}_{qk_q}^\top x}(\tilde{G}_{qk_q}) \right)$$

4.2 Fractile Optimization Model and Corresponding Pareto Optimality

If the decision makers adopt a fractile optimization model for the objective functions of HMOSP, HMOSP can be converted to the following hierarchical multi-objective programming problem, where the decision makers specify permissible probability levels \hat{p}_{ri} , $i = 1, \dots, k_r$, $r = 1, \dots, q$ in their subjective manner [26].

[HMOP1(\hat{p})]

first level decision maker : DM₁

$$\max_{x \in X, h_{1i} \in [0,1]} (h_{11}, \dots, h_{1k_1})$$

\vdots

q -th level decision maker: DM _{q}

$$\max_{x \in X, h_{qi} \in [0,1]} (h_{q1}, \dots, h_{qk_q})$$

subject to

$$\Pr(\omega \mid \Pi_{\tilde{c}_{ri}(\omega)^\top x}(\tilde{G}_{ri}) \geq h_{ri}) \geq \hat{p}_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (36)$$

where $\hat{p}_r = (\hat{p}_{r1}, \dots, \hat{p}_{rk_r})$ is a vector of permissible probability levels. Since a distribution function $T_{ri}(\cdot)$ is continuous and strictly increasing, the constraints (36) can be transformed to the following form.

$$\begin{aligned} \hat{p}_{ri} &\leq \Pr(\omega \mid \Pi_{\tilde{c}_{ri}(\omega)^\top x}(\tilde{G}_{ri}) \geq h_{ri}) = T_{ri} \left(\frac{\mu_{\tilde{G}_{ri}}^{-1}(h_{ri}) - \{(\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}\}}{(\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}} \right) \\ &\Leftrightarrow \mu_{\tilde{G}_{ri}}^{-1}(h_{ri}) \geq \{(\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}\} + T_{ri}^{-1}(\hat{p}_{ri}) \{(\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}\} \end{aligned} \quad (37)$$

Let us define the right-hand side of the inequality (37) as follows.

$$f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri}) = \{(\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}\} + T_{ri}^{-1}(\hat{p}_{ri}) \{(\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}\} \quad (38)$$

Then, HMOP1(\hat{p}) can be equivalently transformed into the following form.

[HMOP2(\hat{p})]

r-th level decision maker : DM_r

$$\max_{\mathbf{x} \in X, h_{ri} \in [0,1]} (h_{r1}, \dots, h_{rk_r})$$

subject to

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) \geq h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (39)$$

In HMOP2(\hat{p}), let us pay attention to the inequalities (39). $f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})$ is continuous and strictly increasing with respect to h_{ri} for any $\mathbf{x} \in X$. This means that the left-hand-side of (39) is continuous and strictly decreasing with respect to h_{ri} for any $\mathbf{x} \in X$. Since the right-hand-side of (39) is continuous and strictly increasing with respect to h_{ri} , the inequalities (39) must always satisfy the active condition, that is, $\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, i = 1, \dots, k_r, r = 1, \dots, q$ at the optimal solution. From such a point of view, HMOP2(\hat{p}) is equivalently expressed as the following form.

[HMOP3(\hat{p})]

r-th level decision maker : DM_r

$$\max_{\mathbf{x} \in X, h_{ri} \in [0,1]} (\mu_{\tilde{G}_{r1}}(f_{r1}(\mathbf{x}, h_{r1}, \hat{p}_{r1})), \dots, \mu_{\tilde{G}_{rk_r}}(f_{rk_r}(\mathbf{x}, h_{rk_r}, \hat{p}_{rk_r})))$$

subject to

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (40)$$

In order to deal with HMOP3(\hat{p}), the decision makers must specify permissible probability levels \hat{p} in advance. However, in general, the decision makers seem to prefer not only the larger value of a permissible probability level but also the larger value of the corresponding membership functions $\mu_{\tilde{G}_{ri}}(\cdot)$. From such a point of view, we consider the following multi-objective programming problem which can be regarded as a natural extension of HMOP3(\hat{p}).

[HMOP4]

r-th level decision maker : DM_r

$$\max_{\mathbf{x} \in X, h_{ri} \in [0,1], \hat{p}_{ri} \in (0,1)} (\mu_{\tilde{G}_{r1}}(f_{r1}(\mathbf{x}, h_{r1}, \hat{p}_{r1})), \dots, \mu_{\tilde{G}_{rk_r}}(f_{rk_r}(\mathbf{x}, h_{rk_r}, \hat{p}_{rk_r})), \hat{p}_{r1}, \dots, \hat{p}_{rk_r})$$

subject to

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (41)$$

It should be noted that in HMOP4 permissible probability levels are not the fixed values but the decision variables.

Considering the imprecise nature of the decision maker's judgment, we assume that each decision maker (DM_r) has a fuzzy goal for each permissible probability level. Such a fuzzy goal can be quantified by eliciting the corresponding membership function. Let us denote a membership function of a permissible probability level \hat{p}_{ri} as $\mu_{\hat{p}_{ri}}(\hat{p}_{ri}), i = 1, \dots, k_r$. Then, HMOP4 can be transformed as the following hierarchical multi-objective programming problem.

[HMOP5]

r-th level decision maker : DM_r

$$\max_{\mathbf{x} \in X, h_{ri} \in [0,1], \hat{p}_{ri} \in (0,1)} (\mu_{\tilde{G}_{r1}}(f_{r1}(\mathbf{x}, h_{r1}, \hat{p}_{r1})), \dots, \mu_{\tilde{G}_{rk_r}}(f_{rk_r}(\mathbf{x}, h_{rk_r}, \hat{p}_{rk_r})), \mu_{\hat{p}_{r1}}(\hat{p}_{r1}), \dots, \mu_{\hat{p}_{rk_r}}(\hat{p}_{rk_r}))$$

subject to

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (42)$$

In order to elicit the membership functions appropriately, we suggest the following procedures. First of all, each decision maker (DM_r) sets the intervals $P_{ri} = [p_{ri}^{\min}, p_{ri}^{\max}]$, $i = 1, \dots, k_r$, where p_{ri}^{\min} is an unacceptable maximum value of \hat{p}_{ri} and p_{ri}^{\max} is a sufficiently satisfactory minimum value of \hat{p}_{ri} . Throughout this section, we make the following assumption.

Assumption 1. $\mu_{\hat{p}_{ri}}(\hat{p}_{ri})$, $i = 1, \dots, k_r$, $r = 1, \dots, q$ are strictly increasing and continuous with respect to $\hat{p}_{ri} \in P_{ri}$, and $\mu_{\hat{p}_{ri}}(p_{ri}^{\min}) = 0$, $\mu_{\hat{p}_{ri}}(p_{ri}^{\max}) = 1$.

It should be noted here that, $\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri}))$ is strictly decreasing with respect to \hat{p}_{ri} . If the decision makers adopt the fuzzy decision [50] to integrate $\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri}))$ and $\mu_{\hat{p}_{ri}}(\hat{p}_{ri})$, HMOP5 can be transformed into the following form.

[HMOP6]

r-th level decision maker : DM_r

$$\max_{\mathbf{x} \in X, \hat{p}_{ri} \in P_{ri}, h_{ri} \in [0,1]} (\mu_{D_{G_{r1}}}(\mathbf{x}, h_{r1}, \hat{p}_{r1}), \dots, \mu_{D_{G_{rk_r}}}(\mathbf{x}, h_{rk_r}, \hat{p}_{rk_r}))$$

subject to

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (43)$$

where

$$\mu_{D_{G_{ri}}}(\mathbf{x}, h_{ri}, \hat{p}_{ri}) = \min \{ \mu_{\hat{p}_{ri}}(\hat{p}_{ri}), \mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) \} \quad (44)$$

In order to deal with HMOP6, we introduce a D_G -Pareto optimal solution concept.

Definition 5. $\mathbf{x}^* \in X$, $\hat{p}_{ri}^* \in P_{ri}$, $h_{ri}^* \in [0, 1]$ is said to be a D_G -Pareto optimal solution to HMOP6, if and only if there does not exist another $\mathbf{x} \in X$, $\hat{p}_{ri} \in P_{ri}$, $h_{ri} \in [0, 1]$ such that $\mu_{D_{G_{ri}}}(\mathbf{x}, h_{ri}, \hat{p}_{ri}) \geq \mu_{D_{G_{ri}}}(\mathbf{x}^*, h_{ri}^*, \hat{p}_{ri}^*)$ with strict inequality holding for at least one i , where $\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}^*, h_{ri}^*, \hat{p}_{ri}^*)) = h_{ri}^*$, $\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}$.

For generating a candidate of a satisfactory solution which is also D_G -Pareto optimal, each decision maker is asked to specify the reference membership values [50]. Once the reference membership values $\hat{\mu} = (\hat{\mu}_{r1}, \dots, \hat{\mu}_{rk_r})$ are specified, the corresponding D_G -Pareto optimal solution is obtained by solving the following minmax problem.

[MINMAX1($\hat{\mu}$)]

$$\min_{\mathbf{x} \in X, \hat{p}_{ri} \in P_{ri}, h_{ri} \in [0,1], \lambda \in \Lambda} \lambda$$

subject to

$$\hat{\mu}_{ri} - \mu_{\hat{p}_{ri}}(\hat{p}_{ri}) \leq \lambda, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (45)$$

$$\hat{\mu}_{ri} - h_{ri} \leq \lambda, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (46)$$

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (47)$$

where $\Lambda = [\max_{i=1, \dots, k_r, r=1, \dots, q} \hat{\mu}_{ri} - 1, \min_{i=1, \dots, k_r, r=1, \dots, q} \hat{\mu}_{ri}]$.

It should be noted here that, in general, the optimal solution of MINMAX1($\hat{\mu}$) does not reflect the hierarchical structure between q decision makers where the upper level decision maker can take priority over the lower level decision makers. In order to cope with such a hierarchical preference structure between q decision makers in MINMAX1($\hat{\mu}$), we introduce the concept of the decision powers $\mathbf{w} = (w_1, \dots, w_q)$ for the membership functions (44), where the r -th level decision maker (DM_r) can specify the decision power w_{r+1} in a subjective manner and the last decision maker (DM_q) has no decision power. In order to reflect the hierarchical

preference structure between multiple decision makers, the decision powers $\mathbf{w} = (w_1, \dots, w_q)$ have to satisfy the following inequality condition.

$$w_1 = 1 \geq w_2 \geq \dots \geq w_{q-1} \geq w_q > 0 \quad (48)$$

Then, the corresponding modified MINMAX1($\hat{\mu}$) is reformulated as follows.

[MINMAX2($\hat{\mu}, \mathbf{w}$)]

$$\min_{\mathbf{x} \in X, \hat{p}_{ri} \in P_{ri}, \lambda \in \Lambda, h_{ri} \in [0,1]} \lambda$$

subject to

$$\hat{\mu}_{ri} - \mu_{\hat{p}_{ri}}(\hat{p}_{ri}) \leq \lambda/w_r, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (49)$$

$$\hat{\mu}_{ri} - h_{ri} \leq \lambda/w_r, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (50)$$

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = h_{ri}, \quad i = 1, \dots, k_r, \quad r = 1, \dots, q \quad (51)$$

where $\Lambda = [\max_{i=1, \dots, k_r, r=1, \dots, q} w_i(\hat{\mu}_{ri} - 1), \min_{i=1, \dots, k_r, r=1, \dots, q} w_i \hat{\mu}_{ri}]$.

In the constraints (50) and (51), it holds that

$$\begin{aligned} h_{ri} &= \mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) \geq \hat{\mu}_{ri} - \lambda/w_r, \\ \Leftrightarrow \mu_{\tilde{G}_{ri}}^{-1}(h_{ri}) &= f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri}) \leq \mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \\ \Leftrightarrow \mu_{\tilde{G}_{ri}}^{-1}(h_{ri}) &= \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} \right\} \\ &\quad + T_{ri}^{-1}(\hat{p}_{ri}) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\} \leq \mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r). \end{aligned} \quad (52)$$

In the right hand side of (52), because of $L^{-1}(h_{ri}) \leq L^{-1}(\hat{\mu}_{ri} - \lambda/w_r)$ and $(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} > 0$, it holds that

$$\begin{aligned} &\left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} \right\} + T_{ri}^{-1}(\hat{p}_{ri}) \cdot \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(h_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\} \\ &= \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\mathbf{d}_{ri}^2)^\top \mathbf{x} \right\} - L^{-1}(h_{ri}) \left\{ (\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\} \\ &\geq \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\mathbf{d}_{ri}^2)^\top \mathbf{x} \right\} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \left\{ (\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\}. \end{aligned} \quad (53)$$

Using (52) and (53), it holds that

$$\begin{aligned} &\mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \\ &\geq \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\mathbf{d}_{ri}^2)^\top \mathbf{x} \right\} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \left\{ (\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} + T_{ri}^{-1}(\hat{p}_{ri})(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\} \\ &= \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \boldsymbol{\alpha}_{ri}^1 \right\} + T_{ri}^{-1}(\hat{p}_{ri}) \cdot \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \boldsymbol{\alpha}_{ri}^2 \right\}. \end{aligned} \quad (54)$$

Moreover, because of $\hat{p}_{ri} \geq \mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r)$, (54) can be transformed into the following form.

$$\begin{aligned} &T_{ri} \left(\frac{\mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r) - \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \boldsymbol{\alpha}_{ri}^1 \right\}}{(\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r) \boldsymbol{\alpha}_{ri}^2} \right) \geq \hat{p}_{ri} \geq \mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r), \\ \Leftrightarrow \mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda) &\geq \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda) \boldsymbol{\alpha}_{ri}^1 \right\} \\ &\quad + T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda)) \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda) \boldsymbol{\alpha}_{ri}^2 \right\}. \end{aligned} \quad (55)$$

Therefore, MINMAX2($\hat{\mu}, \mathbf{w}$) can be reduced to the following minmax problem.

[MINMAX3($\hat{\mu}, \mathbf{w}$)]

$$\min_{\mathbf{x} \in X, \lambda \in \Lambda} \lambda$$

subject to

$$\begin{aligned} \mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r) &\geq \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r)(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} \right\} \\ &\quad + T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda/w_r)) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda/w_r)(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\}, \end{aligned} \quad (56)$$

$$i = 1, \dots, k_r, r = 1, \dots, q$$

It should be noted here that MINMAX3($\hat{\mu}, \mathbf{w}$) is equivalent to MINMAX3($\hat{\mu}, \mathbf{w}$). The relationships between the optimal solution (\mathbf{x}^*, λ^*) of MINMAX3($\hat{\mu}, \mathbf{w}$) and D_G -Pareto optimal solutions can be characterized by the following main theorem.

Theorem 2 (Yano and Matsui [57]). *1. If $\mathbf{x}^* \in X$, $\lambda^* \in \Lambda$ is a unique optimal solution of MINMAX3($\hat{\mu}, \mathbf{w}$), then $\mathbf{x}^* \in X$, $\hat{p}_{ri}^* = \mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r) \in P_{ri}$, $h_{ri}^* = \hat{\mu}_{ri} - \lambda^*/w_r \in [0, 1]$ is a D_G -Pareto optimal solution.*

2. If $\mathbf{x}^ \in X$, $\hat{p}_{ri}^* \in P_{ri}$, $h_{ri}^* \in [0, 1]$ is a D_G -Pareto optimal solution, then $\mathbf{x}^* \in X$, $\lambda^* = w_r(\hat{\mu}_{ri} - \mu_{\hat{p}_{ri}}(\hat{p}_{ri}^*))$ is an optimal solution of MINMAX3($\hat{\mu}, \mathbf{w}$) for certain reference membership values $\hat{\mu} = (\hat{\mu}_{r1}, \dots, \hat{\mu}_{rk_r})$.*

4.3 An Interactive Algorithm

In this section, we propose an interactive algorithm to obtain a satisfactory solution from among a D_G -Pareto optimal solution set. From Theorem 2, it is not guaranteed that the optimal solution (\mathbf{x}^*, λ^*) of MINMAX3($\hat{\mu}, \mathbf{w}$) is D_G -Pareto optimal, if it is not unique. In order to guarantee the D_G -Pareto optimality, we first assume that all of the constraints (57) of MINMAX3($\hat{\mu}, \mathbf{w}$) are active at the optimal solution (\mathbf{x}^*, λ^*), i.e.

$$\begin{aligned} &\mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r) - \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}^* \right\} \\ &= T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}^* \right\}, \end{aligned} \quad (57)$$

$$i = 1, \dots, k_r, r = 1, \dots, q.$$

If some constraint of (57) is inactive, i.e.

$$\mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r) - \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}^* \right\} \quad (58)$$

$$> T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}^* \right\},$$

$$\Leftrightarrow \mu_{\tilde{G}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r) > f_{ri}(\mathbf{x}^*, \hat{\mu}_{ri} - \lambda^*/w_r, \mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)), \quad (59)$$

we can convert the inactive constraint (59) into the active one by applying the bisection method for the reference membership value $\hat{\mu}_{ri} \in [\lambda^*/w_r, \lambda^*/w_r + 1]$.

For the optimal solution (\mathbf{x}^*, λ^*) of MINMAX3($\hat{\mu}, \mathbf{w}$), where the active conditions (87) are satisfied, we solve the D_G -Pareto optimality test problem defined as follows.

[D_G -Pareto optimality test problem]

$$\max_{\mathbf{x} \in X, \epsilon_{ri} \geq 0, i=1, \dots, k_r, r=1, \dots, q} w = \sum_{r=1}^q \sum_{i=1}^{k_r} \epsilon_{ri} \quad (60)$$

Algorithm 3 Interactive algorithm for HMOFRLP

Initialize: Set the initial decision powers as $w_r = 1, r = 1, \dots, q$ and $t = 0$ and the initial reference membership values as $\hat{\mu}_{ri} = 1, i = 1, \dots, k_r, r = 1, \dots, q$.

repeat

Step 1: $DM_r, r = 1, \dots, q$ set the membership functions $\mu_{\tilde{G}_{ri}}(y), i = 1, \dots, k_r$ for the fuzzy goals of the objective functions in HMOFRLP.

Step 2: DM_r sets the intervals

$$P_{ri} = [p_{ri}^{\min}, p_{ri}^{\max}], i = 1, \dots, k_r.$$

Then according to Assumption 1, DM_r sets the membership function $\mu_{\hat{p}_{ri}}(\hat{p}_{ri})$.

Step 3: Solve MINMAX3($\hat{\mu}, \mathbf{w}$) by combined use of the bisection method with respect to $\lambda \in \Lambda$ and the first-phase of the two-phase simplex method of linear programming. For the optimal solution $(\mathbf{x}^*, \lambda^*)$, solve the corresponding D_G -Pareto optimality test problem.

Step 4: Let DM_s be the uppermost of the decision makers who are not satisfied with the current values of the membership functions. Then DM_s updates the decision power w_{s+1} and/or the reference membership values $\hat{\mu}_{si}$ according to the following two rules (Rule 2, 3), and return to Step 3.

until $DM_r, r = 1, \dots, q - 1$ is satisfied with the values of the D_G -Pareto optimal solution

subject to

$$\begin{aligned} & T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x} \right\} \\ & \quad + \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x} - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x} \right\} + \epsilon_{ri} \\ & = T_{ri}^{-1}(\mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)) \times \left\{ (\mathbf{d}_{ri}^2)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^2)^\top \mathbf{x}^* \right\} \\ & \quad + \left\{ (\mathbf{d}_{ri}^1)^\top \mathbf{x}^* - L^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r)(\boldsymbol{\alpha}_{ri}^1)^\top \mathbf{x}^* \right\}, i = 1, \dots, k_r, r = 1, \dots, q. \end{aligned} \quad (61)$$

For the optimal solution of D_G -Pareto optimality test problem, the following theorem holds.

Theorem 3 (Yano and Matsui [57]). *For the optimal solution $\check{\mathbf{x}}, \check{\epsilon}_{ri}, i = 1, \dots, k_r, r = 1, \dots, q$ of D_G -Pareto optimality test problem, if $w = 0$ (equivalently, $\check{\epsilon}_{ri} = 0, i = 1, \dots, k_r, r = 1, \dots, q$), $\mathbf{x}^* \in X, \mu_{\hat{p}_{ri}}^{-1}(\hat{\mu}_{ri} - \lambda^*/w_r) \in P_{ri}, \hat{\mu}_{ri} - \lambda^*/w_r \in [0, 1], i = 1, \dots, k_r, r = 1, \dots, q$ is a D_G -Pareto optimal solution.*

Now, following the above discussions, we can present the interactive algorithm in order to derive a satisfactory solution from among a D_G -Pareto optimal solution set.

Rule 2 (Update the decision power). *In order to guarantee the inequality conditions (48), $w_{s+1} \leq w_s$ must be satisfied. After updating w_{s+1} , if there exists some index $t > s + 1$ such that $w_{s+1} < w_t$, then the corresponding decision power w_t is set as $w_t \leftarrow w_{s+1}$.*

Rule 3 (Update the reference membership values). *At first, the reference membership values of DM_r must be set as the current values of the membership functions, i.e.*

$$\hat{\mu}_{ri} = \mu_{D_{G_{ri}}}(\mathbf{x}^*, h_{ri}^*, \hat{p}_{ri}^*).$$

After that, DM_s updates the reference membership values $\hat{\mu}_{si}, i = 1, \dots, k_s$. Here, it should be stressed for DM_s that any improvement of one membership function can be achieved only at the expense of at least one of the other membership functions.

d_{11}^1	19	48	21	10	18	35	46	11	24	33
d_{11}^2	3	2	2	1	4	3	1	2	4	2
d_{12}^1	12	46	23	38	33	48	12	8	19	20
d_{12}^2	1	2	4	2	2	1	2	1	2	1
d_{21}^1	-12	-38	-23	-33	-33	-45	-12	-9	-19	-20
d_{21}^2	1	2	4	2	2	1	2	1	2	1
d_{22}^1	-12	-36	-27	-30	-33	-45	-11	-12	-19	-8
d_{22}^2	1	2	4	2	2	1	2	1	2	1
α_{11}^1	0.312	0.759	0.225	0.990	0.248	0.951	0.643	0.984	0.340	0.465
α_{11}^2	0.098	0.042	0.074	0.052	0.045	0.024	0.06	0.036	0.082	0.035
α_{12}^1	0.756	0.704	0.295	0.508	0.216	0.859	0.692	0.741	0.641	0.569
α_{12}^2	0.042	0.058	0.06	0.095	0.081	0.068	0.096	0.091	0.011	0.09
α_{21}^1	0.605	0.107	0.564	0.885	0.652	0.957	0.464	0.733	0.230	0.416
α_{21}^2	0.083	0.074	0.058	0.06	0.032	0.096	0.042	0.076	0.081	0.083
α_{22}^1	0.272	0.313	0.442	0.583	0.341	0.641	0.892	0.432	0.611	0.147
α_{22}^2	0.085	0.056	0.065	0.096	0.066	0.043	0.026	0.091	0.052	0.035
a_1	12	-2	4	-7	13	-1	-6	6	11	-8
a_2	-2	5	3	16	6	-12	12	4	-7	-10
a_3	3	-16	-4	-8	-8	2	-12	-12	4	-3
a_4	-11	6	-5	9	-1	8	-4	6	-9	6
a_5	-4	7	-6	-5	13	6	-2	-5	14	-6
a_6	5	-3	14	-3	-9	-7	4	-4	-5	9
a_7	-3	-4	-6	9	6	18	11	-9	-4	7

Table 2: Parameters of objective functions

4.4 A Numerical Experiment

In order to demonstrate the proposed method and the interactive processes, we consider the following toy example of hierarchical two-objective linear programming problem with fuzzy random variable coefficients under two hypothetical decision makers.

[HMOFRLP]

r-th level decision maker : DM_r

$$\min_{\mathbf{x} \in X} \quad \tilde{\mathbf{c}}_r \mathbf{x} = \sum_{j=1}^{10} \tilde{c}_{rj} x_j$$

where $X = \{(x_1, \dots, x_{10}) \geq 0 \mid \sum_{j=1}^{10} a_{kj} x_j \leq b_k, 1 \leq k \leq 7\}$, and it is assumed that the realization of an LR-type fuzzy random variable \tilde{c}_{rj} is an LR fuzzy number whose membership function is defined as

$$\mu_{\tilde{c}_{rj}(\omega)}(s) = \begin{cases} L\left(\frac{d_{rj}^1 + \tilde{t}_r(\omega)d_{rj}^2 - s}{\alpha_{rj}^1 + \tilde{t}_r(\omega)\alpha_{rj}^2}\right) & s \leq \bar{d}_{rj}(\omega), \\ R\left(\frac{s - d_{rj}^1 + \tilde{t}_r(\omega)d_{rj}^2}{\beta_{rj}^1 + \tilde{t}_r(\omega)\beta_{rj}^2}\right) & s > \bar{d}_{rj}(\omega), \end{cases}$$

where $L(t) = R(t) = \max\{0, 1 - t\}$, and $d_{rj}^1, d_{rj}^2, \alpha_{rj}^1, \alpha_{rj}^2, \beta_{rj}^1, \beta_{rj}^2$ are constant parameters. Here we focus on the L function and summarize its parameters in Table 2. Similarly, the parameters of the left-hand-side of the constraints are shown in Table 2 and the right-hand-side of the

constraints are $b_1 = 140$, $b_2 = -220$, $b_3 = -190$, $b_4 = 75$, $b_5 = -160$, $b_6 = 130$, $b_7 = 90$ respectively. For simplicity, we assume that \tilde{t}_{ri} , $r = 1, 2$, $i = 1, 2$ are standard Gaussian random variables defined as $\tilde{t}_{ri} \sim N(0, 1)$.

In HMOFRLP, let us assume that the hypothetical decision makers set their membership functions for the objective functions as follows (Step 1).

$$\mu_{\tilde{G}_{ri}}(f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri})) = \frac{f_{ri}(\mathbf{x}, h_{ri}, \hat{p}_{ri}) - f_{ri}^{\max}}{f_{ri}^{\min} - f_{ri}^{\max}}$$

where $[f_{11}^{\min}, f_{11}^{\max}] = [2400, 2600]$, $[f_{12}^{\min}, f_{12}^{\max}] = [1400, 1500]$, $[f_{21}^{\min}, f_{21}^{\max}] = [-1000, -800]$, $[f_{22}^{\min}, f_{22}^{\max}] = [-700, -500]$.

We also assume that the hypothetical decision makers set their membership functions for the permissible probability levels as follows (Step 2).

$$\mu_{\hat{p}_{ri}}(\hat{p}_{ri}) = \frac{p_{ri}^{\min} - \hat{p}_{ri}}{p_{ri}^{\min} - p_{ri}^{\max}}$$

where $[p_{11}^{\min}, p_{11}^{\max}] = [0.3, 0.9]$, $[p_{12}^{\min}, p_{12}^{\max}] = [0.2, 0.85]$, $[p_{21}^{\min}, p_{21}^{\max}] = [0.3, 0.87]$, $[p_{22}^{\min}, p_{22}^{\max}] = [0.4, 0.91]$. At Step 3, solve MINMAX3($\hat{\mu}, \mathbf{w}$) to obtain the corresponding D_G -Pareto optimal solution $(\mathbf{x}^*, \lambda^*)$. For the current value of the D_G -Pareto optimal solution $(\mathbf{x}^*, \lambda^*)$, DM₁ updates the decision power as $w_2 = 0.7$ in order to improve the membership functions $\mu_{D_{G_{1i}}}(\cdot)$, $i = 1, 2$ at the expense of DM₂'s membership functions $\mu_{D_{G_{2i}}}(\cdot)$, $i = 1, 2$ slightly (Step 4). Then, return to Step 3, and the corresponding D_G -Pareto optimal solution is obtained. At this D_G -Pareto optimal solution, in order to improve $\mu_{D_{G_{11}}}(\cdot)$ at the expense of $\mu_{D_{G_{12}}}(\cdot)$, DM₁ sets the reference membership values as $(\hat{\mu}_{11}, \hat{\mu}_{12}) = (0.838, 0.838)$ according to Rule 2 (Step 4). Then, DM₁ is satisfied with the current value of the membership functions, but DM₂ is not satisfied with the current values. Therefore, in order to improve $\mu_{D_{G_{22}}}(\cdot)$ at the expense of $\mu_{D_{G_{21}}}(\cdot)$, DM₂ updates the reference membership values as $(\hat{\mu}_{21}, \hat{\mu}_{22}) = (0.730, 0.790)$ according to Rule 2 (Step 4). Then, return to Step 3, and the corresponding D_G -Pareto optimal solution is obtained. Since two hypothetical decision makers (DM₁ and DM₂) are satisfied with current values of the above membership functions, stop the interactive processes. The interactive processes under the hypothetical decision makers are summarized in Table 3. In Table 3, it should be noted here that the proper valance between permissible probability levels and the corresponding objective functions is attained at any D_G -Pareto optimal solution in each iteration.

5 Random Fuzzy Multi-objective Linear Programming with Variance Covariance Matrices

5.1 Problem Formulation

In this section, we focus on random fuzzy multi-objective linear programming (denoted as RFMOLP-VC) in which random variable coefficients are involved in objective functions.

[RFMOLP-VC]

$$\min_{\mathbf{x} \in X} \tilde{\bar{C}}\mathbf{x} = (\tilde{\bar{c}}_1^\top \mathbf{x}, \dots, \tilde{\bar{c}}_k^\top \mathbf{x})$$

where X is a feasible region consists of some linear constraints, $\tilde{\bar{c}}_i = (\tilde{\bar{c}}_{i1}, \dots, \tilde{\bar{c}}_{in})$ are coefficient vectors of the objective function whose elements are random fuzzy variables [38], and the symbols "-" and "~" mean randomness and fuzziness respectively.

	iteration1	iteration2	iteration3
w_1	1	1	1
w_2	1	0.7	0.7
$\hat{\mu}_{11}$	1	1	0.838
$\hat{\mu}_{12}$	1	1	0.838
$\hat{\mu}_{21}$	1	1	0.730
$\hat{\mu}_{22}$	1	1	0.790
$\mu_{\tilde{G}_{11}}(f_{11}(\mathbf{x}, h_{11}, \hat{p}_{11}))$	0.804	0.838	0.852
$\mu_{\tilde{G}_{12}}(f_{12}(\mathbf{x}, h_{12}, \hat{p}_{12}))$	0.804	0.838	0.852
$\mu_{\tilde{G}_{21}}(f_{21}(\mathbf{x}, h_{21}, \hat{p}_{21}))$	0.804	0.769	0.750
$\mu_{\tilde{G}_{22}}(f_{22}(\mathbf{x}, h_{22}, \hat{p}_{22}))$	0.804	0.769	0.810
$\mu_{\hat{p}_{11}}(\hat{p}_{11})$	0.804	0.838	0.852
$\mu_{\hat{p}_{12}}(\hat{p}_{12})$	0.804	0.838	0.852
$\mu_{\hat{p}_{21}}(\hat{p}_{21})$	0.804	0.769	0.750
$\mu_{\hat{p}_{22}}(\hat{p}_{22})$	0.804	0.769	0.810
$f_{11}(\mathbf{x}, h_{11}, \hat{p}_{11})$	2439.165	2432.242	2429.407
$f_{12}(\mathbf{x}, h_{12}, \hat{p}_{12})$	1419.582	1416.121	1414.703
$f_{21}(\mathbf{x}, h_{21}, \hat{p}_{21})$	-960.835	-953.940	-950.050
$f_{22}(\mathbf{x}, h_{22}, \hat{p}_{22})$	-660.35	-653.940	-662.050
\hat{p}_{11}	0.782	0.803	0.811
\hat{p}_{12}	0.722	0.745	0.754
\hat{p}_{21}	0.758	0.738	0.727
\hat{p}_{22}	0.810	0.792	0.813

Table 3: Interactive Process

In this section, we assume that a random fuzzy variable \tilde{c}_{ij} is normally distributed with the fuzzy number \tilde{M}_{ij} as mean, and the positive-definite variance covariance matrices V_i between random fuzzy variables \tilde{c}_{ij_1} and \tilde{c}_{ij_2} , $j_1, j_2 = 1, \dots, n$ are given as:

$$V_i = \begin{pmatrix} \sigma_{i11} & \sigma_{i12} & \cdots & \sigma_{i1n} \\ \sigma_{i21} & \sigma_{i22} & \cdots & \sigma_{i2n} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{in1} & \sigma_{in2} & \cdots & \sigma_{inn} \end{pmatrix}, \quad i = 1, \dots, k.$$

As a result, we assume that a probability density function $f_{ij}(y)$ for a random fuzzy variable \tilde{c}_{ij} is formally represented with the following form.

$$f_{ij}(y) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(y-\tilde{M}_{ij})^2}{2\sigma_{ij}^2}}, \quad (62)$$

where \tilde{M}_{ij} is an L-R fuzzy number characterized by the following membership function.

$$\mu_{\tilde{M}_{ij}}(t) = \begin{cases} L\left(\frac{m_{ij}-t}{\alpha_{ij}}\right) & \text{if } m_{ij} \geq t \\ R\left(\frac{t-m_{ij}}{\beta_{ij}}\right) & \text{if } m_{ij} \leq t \end{cases} \quad (63)$$

L and R are called reference functions, m_{ij} is the mean value, and α_{ij} , β_{ij} are spread parameters.

Then, a random fuzzy variable \tilde{c}_{ij} can be characterized by the following membership function.

$$\mu_{\tilde{c}_{ij}}(\tilde{\gamma}_{ij}) = \sup_{s_{ij}} \left\{ \mu_{\tilde{M}_{ij}}(s_{ij}) \mid \tilde{\gamma}_{ij} \sim N(s_{ij}, \sigma_{ij}^2) \right\} \quad (64)$$

where $N(s_{ij}, \sigma_{ij}^2)$ means a normal distribution with mean s_{ij} and standard deviation σ_{ij} . Moreover, using Zadeh's extension principle [61], the objective function $\tilde{c}_i \mathbf{x}$ become a random fuzzy variable characterized by the following membership function.

$$\mu_{\tilde{c}_i^\top \mathbf{x}}(\tilde{u}_i) = \sup_{(s_{i1}, \dots, s_{in}) \in \mathbb{R}^n} \left\{ \min_{1 \leq j \leq n} \mu_{\tilde{M}_{ij}}(s_{ij}) \mid \tilde{u}_i \sim N\left(\sum_{j=1}^n s_{ij} x_j, \mathbf{x}^\top V_i \mathbf{x}\right) \right\} \quad (65)$$

Unfortunately, RFMOLP-VC is ill-defined problem. Katagiri et al. [27, 32] formulated RFMOLP-VC using permissible objective levels of a probability maximization model and the possibility measure. For permissible objective levels f_i , $i = 1, \dots, k$ specified by the decision maker, a probability maximization model for RFMOLP-VC can be formulated as follows.

[MOP1(f)]

$$\max_{\mathbf{x} \in X} (\Pr(\omega | \tilde{c}_1(\omega)^\top \mathbf{x} \leq f_1), \dots, \Pr(\omega | \tilde{c}_k(\omega)^\top \mathbf{x} \leq f_k))$$

It should be noted here that the each objective function :

$$\tilde{P}_i(\mathbf{x}, f_i) = \Pr(\omega | \tilde{c}_i(\omega)^\top \mathbf{x} \leq f_i), \quad (66)$$

becomes a fuzzy set and the corresponding membership function is defined as follows.

$$\mu_{\tilde{P}_i(\mathbf{x}, f_i)}(p_i) = \sup_{\tilde{u}_i} \left\{ \mu_{\tilde{c}_i^\top \mathbf{x}}(\tilde{u}_i) \mid p_i = \Pr(\omega | \tilde{u}_i(\omega) \leq f_i), \tilde{u}_i \sim N\left(\sum_{j=1}^n s_{ij} x_j, \mathbf{x}^\top V_i \mathbf{x}\right) \right\} \quad (67)$$

Katagiri et al. [27, 32] showed that, from (65), the membership function (67) can be transformed as follows.

$$\mu_{\tilde{P}_i(\mathbf{x}, f_i)}(p_i) = \sup_{(s_{i1}, \dots, s_{in}) \in \mathbb{R}^n} \min_{1 \leq j \leq n} \left\{ \mu_{\tilde{M}_{ij}}(s_{ij}) \mid p_i = \Pr(\omega | \tilde{u}_i(\omega) \leq f_i), \tilde{u}_i \sim N\left(\sum_{j=1}^n s_{ij} x_j, \mathbf{x}^\top V_i \mathbf{x}\right) \right\} \quad (68)$$

Using (66), MOP1(f) can be transformed as follows.

[MOP2(f)]

$$\max_{\mathbf{x} \in X} (\tilde{P}_1(\mathbf{x}, f_1), \dots, \tilde{P}_k(\mathbf{x}, f_k))$$

MOP2(f) is ill-defined yet, because objective functions of MOP2(f) are fuzzy sets depending on permissible objective level f_i , $i = 1, \dots, k$. In order to deal with MOP2(f), let us assume that the decision maker has a fuzzy goal \tilde{G}_i for each objective function $\tilde{P}_i(\mathbf{x}, f_i)$, which is expressed in words such as " $\tilde{P}_i(\mathbf{x}, f_i)$ should be substantially less than p_i ". For the corresponding membership function $\mu_{\tilde{G}_i}(p_i)$, we make the following assumption.

Assumption 2. $\mu_{\tilde{G}_i}(p_i)$, $i = 1, \dots, k$ are strictly increasing and continuous with respect to $p_i \in [p_i^{\min}, p_i^{\max}]$, and $\mu_{p_i}(p_i^{\min}) = 0$, $\mu_{p_i}(p_i^{\max}) = 1$, where $0.5 < p_i^{\min}$ is a maximum value of an unacceptable levels and $p_i^{\max} < 1$ is a minimum value of a sufficiently satisfactory levels.

Using possibility measure [13],

$$\Pi_{\tilde{P}_i(x, f_i)}(\tilde{G}_i) = \sup_{p_i} \min \{ \mu_{\tilde{P}_i(x, f_i)}(p_i), \mu_{\tilde{G}_i}(p_i) \}, \quad (69)$$

Katagiri et al. [27, 32] transformed MOP2(f) into the following well-defined multiobjective programming problem.

[MOP3(f)]

$$\max_{x \in X} \left(\Pi_{\tilde{P}_1(x, f_1)}(\tilde{G}_1), \dots, \Pi_{\tilde{P}_k(x, f_k)}(\tilde{G}_k) \right)$$

Unfortunately, in MOP3(f), the decision maker must specify permissible objective levels in advance. However, it seems very difficult to specify such values because $\Pi_{\tilde{P}_i(x, f_i)}(\tilde{G}_i)$ depends on a permissible objective levels f_i . From such a point of view, in this paper, instead of MOP3(f), we consider the following extended problem where f_i are not constants but decision variables.

[MOP4]

$$\max_{x \in X, f_i \in \mathbb{R}} \left(\Pi_{\tilde{P}_1(x, f_1)}(\tilde{G}_1), \dots, \Pi_{\tilde{P}_k(x, f_k)}(\tilde{G}_k), -f_1, \dots, -f_k \right)$$

Considering the imprecise nature of the decision maker's judgment, we assume that the decision maker has a fuzzy goal for each permissible objective level. Such a fuzzy goal can be quantified by eliciting the corresponding membership function. Let us denote a membership function of a permissible objective level f_i as $\mu_{\tilde{F}_i}(f_i)$. For the membership function $\mu_{\tilde{F}_i}(f_i)$, we make the following assumption.

Assumption 3. $\mu_{\tilde{F}_i}(f_i)$, $i = 1, \dots, k$ are strictly decreasing and continuous with respect to $f_i \in [f_i^{\min}, f_i^{\max}]$, and $\mu_{\tilde{F}_i}(f_i^{\min}) = 1$, $\mu_{\tilde{F}_i}(f_i^{\max}) = 0$, where f_i^{\min} is a maximum value of a sufficiently satisfactory levels and f_i^{\max} is a minimum value of an unacceptable levels.

Then, MOP4 can be transformed as the following multi-objective programming problem.

[MOP5]

$$\max_{x \in X, f_i \in \mathbb{R}} \left(\Pi_{\tilde{P}_1(x, f_1)}(\tilde{G}_1), \dots, \Pi_{\tilde{P}_k(x, f_k)}(\tilde{G}_k), \mu_{\tilde{F}_1}(f_1), \dots, \mu_{\tilde{F}_k}(f_k) \right)$$

It should be noted here that, $\Pi_{\tilde{P}_i(x, f_i)}(\tilde{G}_i)$ is strictly increasing with respect to f_i . If the decision maker adopts the fuzzy decision [61] to integrate $\Pi_{\tilde{P}_i(x, f_i)}(\tilde{G}_i)$ and $\mu_{\tilde{F}_i}(f_i)$, MOP5 can be transformed into the following form.

[MOP6]

$$\max_{x \in X, f_i \in \mathbb{R}} (\mu_{D_1}(x, f_1), \dots, \mu_{D_k}(x, f_k))$$

where

$$\mu_{D_i}(x, f_i) = \min \{ \Pi_{\tilde{P}_i(x, f_i)}(\tilde{G}_i), \mu_{\tilde{F}_i}(f_i) \}$$

In order to deal with MOP6, we introduce a D -Pareto optimal solution concept.

Definition 6 (D -Pareto optimality). $x^* \in X, f_i^* \in \mathbb{R}$ is said to be a D -Pareto optimal solution to MOP6, if and only if there does not exist another $x \in X, f_i \in \mathbb{R}$ such that $\mu_{D_i}(x, f_i) \geq \mu_{D_i}(x^*, f_i^*)$, $i = 1, \dots, k$ with strict inequality holding for at least one i .

For generating a candidate of a satisfactory solution which is also D -Pareto optimal, the decision maker is asked to specify the reference membership values [50]. Once the reference membership values $\hat{\mu} = (\hat{\mu}_1, \dots, \hat{\mu}_k)$ are specified, the corresponding D -Pareto optimal solution is obtained by solving the following min-max problem.

[MINMAX1($\hat{\mu}$)]

$$\min_{\mathbf{x} \in X, f_i \in \mathbb{R}, \lambda \in [0,1]} \lambda \quad (70)$$

subject to

$$\hat{\mu}_i - \Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) \leq \lambda, i = 1, \dots, k \quad (71)$$

$$\hat{\mu}_i - \mu_{\tilde{F}_i}(f_i) \leq \lambda, i = 1, \dots, k \quad (72)$$

From [27, 32], each constraint of (71) can be equivalently transformed into the following form.

$$\begin{aligned} & \hat{\mu}_i - \Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) \leq \lambda \\ \Leftrightarrow & \sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda)) \sqrt{\mathbf{x}^\top V_i \mathbf{x}} \leq f_i \end{aligned} \quad (73)$$

where $\Phi(\cdot)$ is a distribution function of the standard Gaussian random variable, $\Phi^{-1}(\cdot)$ is a corresponding inverse function, and $L^{-1}(\cdot), \mu_{\tilde{G}_i}^{-1}(\cdot)$ are pseudo-inverse functions of $L(\cdot), \mu_{\tilde{G}_i}(\cdot)$ respectively. Moreover, since the inequalities (72) can be transformed into $f_i \leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda)$, MINMAX1($\hat{\mu}$) can be reduced to the following problem.

[MINMAX2($\hat{\mu}$)]

$$\min_{\mathbf{x} \in X, \lambda \in \Lambda} \lambda \quad (74)$$

subject to

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda)) \sqrt{\mathbf{x}^\top V_i \mathbf{x}} \leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda), i = 1, \dots, k \quad (75)$$

where

$$\Lambda = [\lambda_{\min}, \lambda_{\max}] = \left[\max_{i=1, \dots, k} \hat{\mu}_i - 1, \min_{i=1, \dots, k} \hat{\mu}_i \right]. \quad (76)$$

The relationships between the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$) and D -Pareto optimal solutions can be characterized by the following main theorem.

Theorem 4 (Yano and Matsui [58]). *1. If $\mathbf{x}^* \in X, \lambda^* \in \Lambda$ is a unique optimal solution of MINMAX2($\hat{\mu}$), then $\mathbf{x}^* \in X, \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*) \in \mathbb{R}, i = 1, \dots, k$ is a D -Pareto optimal solution.*

2. If $\mathbf{x}^ \in X, f_i^* \in \mathbb{R}, i = 1, \dots, k$ is a D -Pareto optimal solution, then $\mathbf{x}^* \in X$ and*

$$\lambda^* = \hat{\mu}_i - \Pi_{\tilde{P}_i(\mathbf{x}^*, f_i^*)}(\tilde{G}_i) = \hat{\mu}_i - \mu_{\tilde{F}_i}(f_i^*)$$

is an optimal solution of MINMAX2($\hat{\mu}$) for some reference membership values.

Proof. 1. Assume that $\mathbf{x}^* \in X, f_i^* = \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*)$ is not a D -Pareto optimal solution. Then, from (71), there exist $\mathbf{x} \in X, f_i \in \mathbb{R}$ such that

$$\begin{aligned}\mu_{D_i}(\mathbf{x}, f_i) &= \min\{\Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i), \mu_{\tilde{F}_i}(f_i)\} \\ &\geq \mu_{D_i}(\mathbf{x}^*, f_i^*) \\ &= \min\{\Pi_{\tilde{P}_i(\mathbf{x}^*, f_i^*)}(\tilde{G}_i), \mu_{\tilde{F}_i}(f_i^*)\} \\ &= \hat{\mu}_i - \lambda^*, i = 1, \dots, k,\end{aligned}$$

with strict inequality holding for at least one i . Then it holds that

$$\Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) \geq \hat{\mu}_i - \lambda^*, i = 1, \dots, k, \quad (77)$$

$$\mu_{\tilde{F}_i}(f_i) \geq \hat{\mu}_i - \lambda^*, i = 1, \dots, k. \quad (78)$$

From (73), (77) can be transformed as follows.

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda^*))\sqrt{\mathbf{x}^\top V_i \mathbf{x}} \leq f_i \quad (79)$$

From (78), it holds that $f_i \leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*)$. As a result, there exists $\mathbf{x} \in X$ such that

$$\begin{aligned}\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda^*))\sqrt{\mathbf{x}^\top V_i \mathbf{x}} &\leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*) \\ i &= 1, \dots, k\end{aligned} \quad (80)$$

which contradicts the fact that $\mathbf{x}^* \in X, \lambda^* \in \Lambda$ is a unique optimal solution to MINMAX2($\hat{\mu}$).

2. Assume that $\mathbf{x}^* \in X, \lambda^* \in \Lambda$ is not an optimal solution to MINMAX2($\hat{\mu}$) for any reference membership values $\hat{\mu} = (\hat{\mu}_1, \dots, \hat{\mu}_k)$, which satisfy the equalities :

$$\hat{\mu}_i - \lambda^* = \Pi_{\tilde{P}_i(\mathbf{x}^*, f_i^*)}(\tilde{G}_i) = \mu_{\tilde{F}_i}(f_i^*). \quad (81)$$

Then, there exists some $\mathbf{x} \in X, \lambda < \lambda^*$ such that

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda))\sqrt{\mathbf{x}^\top V_i \mathbf{x}} \leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda). \quad (82)$$

This means that

$$\begin{aligned}\Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) &\geq \hat{\mu}_i - \lambda > \hat{\mu}_i - \lambda^*, \\ \mu_{\tilde{F}_i}(f_i) &= \hat{\mu}_i - \lambda > \hat{\mu}_i - \lambda^*,\end{aligned}$$

where $f_i = \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda)$. From (81), there exists $\mathbf{x} \in X, f_i \in \mathbb{R}, i = 1, \dots, k$ such that

$$\mu_{D_i}(\mathbf{x}, f_i) > \mu_{D_i}(\mathbf{x}^*, f_i^*).$$

This contradicts the fact that $\mathbf{x}^* \in X, f_i^* \in \mathbb{R}, i = 1, \dots, k$ is a D -Pareto optimal solution. \square

Algorithm 4 Solver for MINMAX2($\hat{\mu}$)

Initialize: Set $\lambda_0 = \lambda_{\min}, \lambda_1 = \lambda_{\max}, \lambda \leftarrow (\lambda_0 + \lambda_1)/2$.

repeat

Step 1: Solve the following convex programming problem for the fixed value λ , and denote the optimal solution as $\mathbf{x}(\lambda)$.

$$\max_{\mathbf{x} \in X} g_j(\mathbf{x}, \lambda) \quad (85)$$

subject to

$$g_i(\mathbf{x}, \lambda) \geq 0, \quad i = 1, \dots, k, i \neq j \quad (86)$$

Step 2: If $g_\ell(\mathbf{x}(\lambda), \lambda) \geq 0, \ell = 1, \dots, k$ then set $\lambda_1 \leftarrow \lambda, \lambda \leftarrow (\lambda_0 + \lambda_1)/2$. Otherwise, set $\lambda_0 \leftarrow \lambda, \lambda \leftarrow (\lambda_0 + \lambda_1)/2$.

until $|\lambda_1 - \lambda_0| \geq \epsilon$

Step 3: Set $\lambda^* \leftarrow \lambda$ and $\mathbf{x}^* \leftarrow \mathbf{x}(\lambda)$.

Output: The optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$).

Since the constraints (75) are nonlinear, it is difficult to solve MINMAX2($\hat{\mu}$) directly. Before considering the algorithm to solve MINMAX2($\hat{\mu}$), we first define the following functions corresponding to (75).

$$g_i(\mathbf{x}, \lambda) = \mu_{\hat{F}_i}^{-1}(\hat{\mu}_i - \lambda) - \sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda)\alpha_{ij}\}x_j - \Phi^{-1}(\mu_{\hat{G}_i}^{-1}(\hat{\mu}_i - \lambda)) \sqrt{\mathbf{x}^\top V_i \mathbf{x}}. \quad (83)$$

Because of $\mu_{\hat{G}_i}^{-1}(\hat{\mu}_i - \lambda) > 0.5$ for any $\lambda \in \Lambda$, it holds that $\Phi^{-1}(\mu_{\hat{G}_i}^{-1}(\hat{\mu}_i - \lambda)) > 0$. This means that $g_i(\mathbf{x}, \lambda), i = 1, \dots, k$ are concave with respect to $\mathbf{x} \in X$ for any fixed value $\lambda \in \Lambda$. Let us define the following feasible set $X(\lambda)$ of MINMAX2($\hat{\mu}$) for some fixed value $\lambda \in \Lambda$.

$$X(\lambda) = \{\mathbf{x} \in X \mid g_i(\mathbf{x}, \lambda) \geq 0, i = 1, \dots, k\} \quad (84)$$

Then, it is clear that $X(\lambda)$ is a convex set. $X(\lambda)$ satisfies the following property.

Property 1. If $\lambda_{\min} \leq \lambda_1 \leq \lambda_2 \leq \lambda_{\max}$, then it holds that $X(\lambda_1) \subset X(\lambda_2)$.

In the following, we assume that $X(\lambda_{\min}) = \emptyset, X(\lambda_{\max}) \neq \emptyset$. From Property 1, we can obtain the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$) using the following simple algorithm which is based on the bisection method and the convex programming technique.

5.2 An Interactive Algorithm

In this section, we propose an interactive algorithm to obtain a satisfactory solution from among a D -Pareto optimal solution set. From Theorem 4, it is not guaranteed that the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$) is D -Pareto optimal, if it is not unique. In order to guarantee the D -Pareto optimality, we first assume that k constraints (75) are active at the optimal solution $(\mathbf{x}^*, \lambda^*)$, i.e.

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j^* + \Phi^{-1}(\mu_{\hat{G}_i}^{-1}(\hat{\mu}_i - \lambda^*)) \sqrt{\mathbf{x}^{*\top} V_i \mathbf{x}^*} = \mu_{\hat{F}_i}^{-1}(\hat{\mu}_i - \lambda^*). \quad (87)$$

If the ℓ -th constraint of (75) is inactive, i.e.

$$\sum_{j=1}^n \{m_{\ell j} - L^{-1}(\hat{\mu}_\ell - \lambda^*)\alpha_{\ell j}\}x_j^* + \Phi^{-1}(\mu_{\tilde{G}_\ell}^{-1}(\hat{\mu}_\ell - \lambda^*)) \sqrt{\mathbf{x}^{*\top} V_\ell \mathbf{x}^*} < \mu_{\tilde{F}_\ell}^{-1}(\hat{\mu}_\ell - \lambda^*), \quad (88)$$

we can convert the inactive constraint (88) into the active one by applying the bisection method for the reference membership value $\hat{\mu}_\ell \in [\lambda^*, \lambda^* + 1]$.

For the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$), where the active conditions (87) are satisfied, we solve the D -Pareto optimality test problem defined as follows.

[D -Pareto optimality test problem]

$$\max_{\mathbf{x} \in X, \epsilon_i \geq 0} w = \sum_{i=1}^k \epsilon_i \quad (89)$$

subject to

$$\begin{aligned} & \sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda^*)) \sqrt{\mathbf{x}^\top V_i \mathbf{x}} + \epsilon_i \\ & \leq \sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j^* + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda^*)) \sqrt{\mathbf{x}^{*\top} V_i \mathbf{x}^*}. \end{aligned} \quad (90)$$

For the optimal solution of the above test problem, we proved the following theorem.

Theorem 5 (Yano and Matsui [58]). *For the optimal solution $\check{\mathbf{x}}, \check{\epsilon}_i, i = 1, \dots, k$ of the test problem (89)-(90), if $w = 0$ (equivalently, $\check{\epsilon}_i = 0, i = 1, \dots, k$), $\mathbf{x}^* \in X, f_i^* = \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*) \in \mathbb{R}, i = 1, \dots, k$ is a D -Pareto optimal solution.*

Proof. From the active condition (87) at the optimal solution $(\mathbf{x}^*, \lambda^*)$ of MINMAX2($\hat{\mu}$), it holds that

$$\begin{aligned} \hat{\mu}_i - \lambda^* &= \Pi_{\tilde{P}_i(\mathbf{x}^*, f_i^*)}(\tilde{G}_i), \\ \hat{\mu}_i - \lambda^* &= \mu_{\tilde{F}_i}(f_i^*). \end{aligned}$$

Assume that $\mathbf{x}^* \in X, \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*), i = 1, \dots, k$ is not a D -Pareto optimal solution. Then, there exist $\mathbf{x} \in X, f_i \in \mathbb{R}, i = 1, \dots, k$ such that

$$\begin{aligned} \mu_{D_i}(\mathbf{x}, f_i) &= \min \{ \Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i), \mu_{\tilde{F}_i}(f_i) \} \\ &\geq \mu_{D_i}(\mathbf{x}^*, f_i^*) \\ &= \hat{\mu}_i - \lambda^*, \end{aligned}$$

with strict inequality holding for at least one i . This means that

$$\Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) \geq \hat{\mu}_i - \lambda^*, \quad (91)$$

$$\mu_{\tilde{F}_i}(f_i) \geq \hat{\mu}_i - \lambda^*. \quad (92)$$

From (73), (91) and (92), the following inequalities hold with strict inequality holding for at least one i .

Algorithm 5 Interactive algorithm for MORFLP

Initialize: Set the initial reference membership values as $\hat{\mu}_i = 1, i = 1, \dots, k$.

repeat

Step 1: According to Assumption2, set each of the membership functions $\mu_{\tilde{F}_i}(f_i)$ of the fuzzy goal for permissible objective level f_i .

Step 2: According to Assumption1, set each of the membership functions $\mu_{\tilde{G}_i}(p_i)$ for the probability that the objective function $\tilde{c}_i \mathbf{x}$ is less than f_i .

Step 3: Solve MINMAX2($\hat{\mu}$) by applying Algorithm 4, and obtain the optimal solution $(\mathbf{x}^*, \lambda^*)$. Then solve the corresponding D -Pareto optimality test problem (89)-(90).

Step 4: The decision maker updates the reference membership values $\hat{\mu}_i, i = 1, \dots, k$, and return to Step 3.

until the decision maker is satisfied with the current values of the D -Pareto optimal solution

Step 3: Set $\lambda^* \leftarrow \lambda$ and $\mathbf{x}^* \leftarrow \mathbf{x}(\lambda)$.

Output: The D -Pareto optimal solution (\mathbf{x}^*, f_i^*) .

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda^*)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda^*))\sqrt{\mathbf{x}'V_i\mathbf{x}} \leq \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*). \quad (93)$$

This contradicts the fact that $\xi_i = 0, i = 1, \dots, k$. \square

Now, following the above discussions, we propose the interactive algorithm in order to derive a satisfactory solution from among a set of D -Pareto optimal solution.

5.3 A Numerical Example

5.3.1 Simulation with Agricultural Planning in Mind

In order to demonstrate our proposed decision making method, we consider the following crop planning problem [19, 21, 23], in which a farmer or an agricultural manager wants to maximize the total profit (unit: 1000 yen) and minimize the working time (unit: 1 hour) by using the farmland effectively. In order to decide the planting ratio for four kinds of crops $x_j, j = 1, \dots, 5$ (unit: 1000 m²) in the farmland, we formulate the following multi-objective fuzzy random programming problem.

[RFMOLP-VC]

$$\begin{aligned} \min_{\mathbf{x} \in X} \tilde{\mathbf{c}}_1^\top \mathbf{x} &= \sum_{j=1}^5 \tilde{c}_{1j}x_j \quad (\text{profit maximization}) \\ \min_{\mathbf{x} \in X} \tilde{\mathbf{c}}_2^\top \mathbf{x} &= \sum_{j=1}^5 \tilde{c}_{2j}x_j \quad (\text{labor minimization}) \end{aligned}$$

where

$$X = \left\{ \mathbf{x} \in \mathbb{R}^5 \mid \sum_{j=1}^5 x_j \leq 200, x_1 \geq 15, x_2 \geq 35, x_3 \geq 20, x_4 \geq 25, x_5 \geq 45 \right\}$$

	m_{ij}	$\alpha_{ij}(=\beta_{ij})$	σ_{ij}^2 (the variance of \tilde{c}_{ij})
\tilde{M}_{11}	-4.2	0.5	1
\tilde{M}_{12}	-8	0.4	2
\tilde{M}_{13}	-5.6	0.3	2
\tilde{M}_{14}	-10.6	0.2	3
\tilde{M}_{15}	-9.5	0.3	1
\tilde{M}_{21}	10	0.3	2
\tilde{M}_{22}	5	0.4	3
\tilde{M}_{23}	5	0.4	1
\tilde{M}_{24}	9	0.2	2
\tilde{M}_{25}	5	0.1	1

Table 4: The parameters of the fuzzy number \tilde{M}_{ij} and the variance σ_{ij}^2 of the random fuzzy variable \tilde{c}_{ij}

represents the land constraints and x_j means the cultivation area for crop j .

\tilde{c}_{1j} are profit coefficients at the unit area for crop j , and \tilde{c}_{2j} are working time coefficients for growing crop j at the unit area, each of which is defined as a random fuzzy variable. Random fuzzy variables \tilde{c}_{ij} are normally distributed with the fuzzy number \tilde{M}_{ij} as mean and σ_{ij}^2 as variance, and a probability density function $f_{ij}(y)$ for a random fuzzy variable \tilde{c}_{ij} is formally represented with the following form.

$$f_{ij}(y) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(y-\tilde{M}_{ij})^2}{2\sigma_{ij}^2}}, 1 \leq i \leq 2, 1 \leq j \leq 5 \quad (94)$$

\tilde{M}_{ij} is an L-R fuzzy number characterized by the following membership function.

$$\mu_{\tilde{M}_{ij}}(t) = \begin{cases} L\left(\frac{m_{ij}-t}{\alpha_{ij}}\right), & m_{ij} \geq t \\ R\left(\frac{t-m_{ij}}{\beta_{ij}}\right), & m_{ij} \leq t \end{cases} \quad (95)$$

where $L(t) = R(t) = \max\{0, 1 - t\}$, and $m_{ij}, \alpha_{ij}(=\beta_{ij}), \sigma_{ij}^2$ are given in Table 4.

In RFMOLP-VC, let us assume that the hypothetical decision maker sets the membership functions $\mu_{\tilde{F}_i}(\cdot), \mu_{\tilde{G}_i}(\cdot), i = 1, 2$ as follows (Step 1, 2).

$$\begin{aligned} \mu_{\tilde{F}_1}(f_1) &= \frac{f_1 - (-1000)}{(-1650) - (-1000)} \\ \mu_{\tilde{F}_2}(f_2) &= \frac{f_2 - 1500}{1200 - 1500} \\ \mu_{\tilde{G}_1}(p_1) &= \frac{p_1 - 0.7}{0.85 - 0.7} \\ \mu_{\tilde{G}_2}(p_2) &= \frac{p_2 - 0.8}{0.9 - 0.8} \end{aligned}$$

Set the initial reference membership values as $(\hat{\mu}_1, \hat{\mu}_2) = (1, 1)$ (Step 3), and solve MINMAX2($\hat{\mu}$) to obtain the corresponding D -Pareto optimal solution $(\mathbf{x}^*, \lambda^*)$ (Step 4).

$$(\mu_{D_1}(\mathbf{x}^*, f_1^*), \mu_{D_2}(\mathbf{x}^*, f_2^*)) = (0.7965, 0.7965)$$

Iteration	1	2	3
$\hat{\mu}_1$	1	1	0.93
$\hat{\mu}_2$	1	0.8	0.8
f_1^*	-1517.78	-1597.23	-1559.20
f_2^*	1261.02	1305.01	1280.90
p_1^*	0.8194	0.8378	0.8290
p_2^*	0.8796	0.8649	0.8730
$\mu_{D_1}(\mathbf{x}^*, f_1^*)$	0.7965	0.9188	0.8603
$\mu_{D_2}(\mathbf{x}^*, f_2^*)$	0.7965	0.6499	0.7303

Table 5: Interactive processes

	proposed method	probability max.
f_1^*	-1517.78	-1550
f_2^*	1261.02	1230
p_1^*	0.8194	0.7767
p_2^*	0.8796	0.8488
$\mu_{f_1}(f_1^*)$	0.7965	0.8461
$\mu_{f_2}(f_2^*)$	0.7965	0.900
$\mu_{\tilde{G}_1}(p_1^*)$	0.7965	0.5118
$\mu_{\tilde{G}_2}(p_2^*)$	0.7965	0.4881

Table 6: Comparison between the Proposed Method and the Probability Maximization Method

where $f_i^* = \mu_{\tilde{F}_i}^{-1}(\hat{\mu}_i - \lambda^*)$, $i = 1, 2$. The hypothetical decision maker is not satisfied with the current value of the D -Pareto optimal solution (\mathbf{x}^*, f_i^*) , and, in order to improve $\mu_{D_1}(\cdot)$ at the expense of $\mu_{D_2}(\cdot)$, DM updates the reference membership values as $(\hat{\mu}_1, \hat{\mu}_2) = (1, 0.8)$ (Step 5). Then, the corresponding D -Pareto optimal solution is obtained by solving MINMAX2($\hat{\mu}$) (Step 4). The interactive processes under the hypothetical decision maker are summarized in Table 5

5.3.2 Comparison with Probability Maximization Model

In order to compare our proposed approach with one of a state-of-the-art, probability maximization model, let us consider the following multi-objective programming problem which is based on the probability maximization model for RFMOLP-VC.

[MOP2'(f)]

$$\max_{\mathbf{x} \in X} (\Pi_{\tilde{P}_1(\mathbf{x}, f_1)}(\tilde{G}_1), \Pi_{\tilde{P}_2(\mathbf{x}, f_2)}(\tilde{G}_2))$$

where f_1 and f_2 are permissible objective levels specified by the decision maker in a subjective manner. Once the reference membership values $\hat{\mu} = (\hat{\mu}_1, \hat{\mu}_2)$ are specified by the decision maker, the corresponding Pareto optimal solution is obtained by the following min-max problem.

[MINMAX3($\hat{\mu}, f$)]

$$\min_{\mathbf{x} \in X, \lambda \in \Lambda} \lambda \tag{96}$$

subject to

$$\hat{\mu}_i - \Pi_{\tilde{P}_i(\mathbf{x}, f_i)}(\tilde{G}_i) \leq \lambda, \quad i = 1, 2$$

From (73), $\text{MINMAX3}(\hat{\mu}, f)$ can be equivalently transformed to the following form.
 $[\text{MINMAX4}(\hat{\mu}, f)]$

$$\min_{x \in X, \lambda \in \Lambda} \lambda$$

subject to

$$\sum_{j=1}^n \{m_{ij} - L^{-1}(\hat{\mu}_i - \lambda)\alpha_{ij}\}x_j + \Phi^{-1}(\mu_{\tilde{G}_i}^{-1}(\hat{\mu}_i - \lambda))\sqrt{x^\top V_i x} \leq f_i, \quad i = 1, 2 \quad (97)$$

We can easily solve $\text{MINMAX4}(\hat{\mu}, f)$ by applying Algorithm 1, because the constraint set (97) are convex for any fixed $\lambda \in \Lambda$. Let us assume that the decision maker sets the reference membership values as $(\hat{\mu}_1, \hat{\mu}_2) = (1, 1)$ and permissible objective levels as $(f_1, f_2) = (-1550, 1230)$. Then, the corresponding Pareto optimal solution can be obtained as shown in Table 6, where the left side shows the D -Pareto optimal solution of the proposed method with reference membership values $(\hat{\mu}_1, \hat{\mu}_2) = (1, 1)$. In Table 6, it is clear that, in the proposed method, a proper balance between permissible probability levels and the corresponding objective functions in a probability maximization model is attained in membership space. On the other hand, in a probability maximization model, although permissible objective levels are improved in comparison with the proposed method, the corresponding probability function values was changed for the worse.

6 Summary of Part I

In Part 1, we proposed the interaction-based algorithm to derive a satisfactory solution for three classes of decision-making problems such as hierarchical multi-objective stochastic programming problem, hierarchical fuzzy random programming problem and multi-objective random fuzzy programming problem. We also proved that the solution derived by the proposed method satisfies Pareto optimality under certain conditions. The problems and algorithms discussed in this part are based on the assumption that all decision makers can express their preferences in various ways. For example, decision makers can express a goal of “desiring to make the objective function roughly less than a certain value” as a membership function of the fuzzy goal. In addition, decision makers adopts the integrated function of the membership function as the utility function, and can adaptively set the reference membership value. To such decision makers, the method proposed in this part probably work effectively to derive a satisfactory solution.

On the other hand, it is natural to think that there are many decision makers who can not explicitly express their own preferences. Or, the expressed preference may include some error. In such a case, the proposed method in this part may not be effective. It is because, for example, in comparison between two Pareto solution x_1 and x_2 , there is a possibility that the decision maker answers that x_1 is better at first but x_2 is better when hearing next. The proposed methods are not designed to deal with such noisy preference structure.

In the next part, we formulate such an uncertain and noisy utility maximization as general stochastic optimization problem with limited information feedback. In the problem setting treated in the next part, these requirements to the decision maker are relaxed i.e. the decision maker does not need to explicitly set membership functions, reference membership values, etc., and do not have to explicitly express the utility function.

Part II

A Novel Algorithm for Optimization with Uncertain Information

In the previous part, we proposed interactive methods to derive a satisfactory solution to the problem of the decision maker can express his / her preferences as a membership function and contribute to its updating. However, this assumption made to the decision maker is too strict and it is difficult to apply to directly realistic problems. Therefore, we tried to relax the requirement to the decision maker in the problem of the maximization of the utility function (see Figure 3). The basic idea is to perform the optimization by pairwise comparison of the utility function values at two points. Pairwise comparison of function values can be interpreted as a pairwise comparison of two arguments (in particular, two Pareto solutions). This corresponds to making assumptions to the extent that the decision maker tells us which of the two Pareto solutions is better.

7 Derivative-Free Methods for General Single-Objective Optimization

Recently, demand for large-scale complex optimization is increasing in computational science, engineering and many other fields. In that kind of problems, there are many difficulties caused by noise in function evaluation, many tuning parameters and high computation cost. In such cases, derivatives of the objective function are unavailable or computationally infeasible. These problems can be treated by the derivative-free optimization (DFO) methods.

DFO is the tool for optimization without derivative information of the objective function and constraints, and it has been widely studied for decades [9, 49]. There are roughly two approaches in DFO. The first one is first order methods with estimated gradient [16, 14] and the second one is zero-th order methods (or direct search methods) [2, 44]. A first order method is, for example, the usual gradient descent algorithm in which the gradient of the objective function is unbiasedly estimated. On the other hand, zero-th order methods do not use the true gradient. It uses only function values to estimate a descent direction of the objective function.

There is, however, a more restricted setting in which not only derivatives but also values of the objective function are unavailable or computationally infeasible. In such a situation, the so-called pairwise comparison oracle, that tells us an order of function values on two evaluation points, is used instead of derivatives and function evaluation [44, 24]. For example, the pairwise comparison is used in learning to rank to collect training samples to estimate the preference function of the ranking problems [43]. In decision making, finding the most preferred feasible solution from among the set of many alternatives is an important application of ranking methods using the pairwise comparison. Also, other type of information such as stochastic gradient-sign oracle has been studied [47].

Related Works: Nelder and Mead's downhill simplex method [44] (shortly, Nelder-Mead method) was proposed in an early study of algorithms based on the pairwise comparison of function values. In each iteration of the algorithm, a simplex that approximates the objective function is constructed according to a ranking of function values on sampled points. Then,

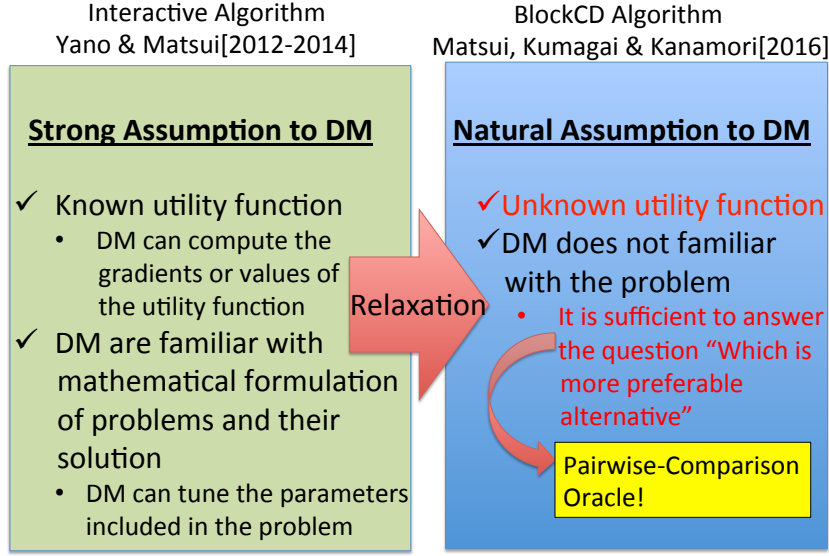


Figure 3: Relaxation of the assumption to the decision maker.

the simplex receives four operations, namely, reflection, expansion, contraction and reduction in order to get close to the optimal solution. The Nelder-Mead method is a state-of-the-art in general optimization (e.g. “optim”, a general optimization function in R language, employs the Nelder-Mead method as a standard solver). Unfortunately, the convergence of the Nelder-Mead algorithm is theoretically guaranteed only in low dimensional problems [35]. In high dimensional problems, the Nelder-Mead algorithm works poorly as shown in [17].

Mesh adaptive direct search (MADS) [4, 10], another deterministic direct search algorithm, can treat non-smooth problems and hidden constraints. Each iteration of MADS consists of three procedures namely, the search, the poll, and the updates. The search explores a finite number of trial points on the *mesh*, constructed from a finite set of scaled directions (namely simplex gradients direction). The poll computes a descent direction from the generated trial points and the current solution is updated along the direction. MADS employs the passive-aggressive strategy, that is, if the candidate point y is not “better than” current solution x (represented by $y \not\prec x$), the iteration is a failure and no update is done. Custódio et al. [12] analyze the properties of the simplex gradients that is used in some direct search methods that include MADS. The parallelization of MADS algorithm is considered in several ways e.g. parallel space decomposition [3].

Stochastic coordinate descent (SCD) algorithm is one of the most powerful algorithm for the (especially large scale) optimization problems. Richtárik and Takáč [48] provided a parallelization of SCD (PSCD) for the class of “partially separable” objective function that is, the objective function f can be represented as the sum of partial objective f_J : $f(\mathbf{x}) = \sum_{J \in \{1, \dots, n\}} f_J$. Here, the decision variable of f_J and $f_{J'}$ do not intersect if $J \cap J' = \emptyset$.

Dueling bandits (DB) [59, 60, 62] is an online optimization problem where one can use only comparisons (called duels) between two points in a set with constraints. This problem is motivated by optimization of an information retrieval system (such as web search) to provide answers of queries that maximize user utility. Yue et al. [60] proposed a gradient descent type algorithm for this problem that uses stochastic pairwise comparisons. In the algorithm, update is taken along a random direction if the candidate point wins the current point by a duel.

Algorithm	Method	Underlying setup	Required information	Parallelization
Nelder-Mead	deterministic	deterministic	pairwise comparison	
MADS	deterministic	deterministic	function value	√
PSCD	stochastic	deterministic	gradient	√
DB	stochastic	stochastic	pairwise comparison	
PCSCD	stochastic	stochastic	pairwise comparison	
Proposal	stochastic	stochastic	pairwise comparison	√

Table 7: Summary of the relationships between proposal and related works. Method : either the algorithm is deterministic or stochastic. Underlying setup : if the problem contains stochastic oracle or not. Required information : the information that is actually used in the algorithm. Parallelization : if the parallelization of the algorithm is considered or not. Our proposal has two characterizations, that is, the first is a parallelization of the PCSCD and the second is the PSCD under the restricted information.

The stochastic coordinate descent algorithm using only the noisy pairwise comparison (PC-SCD) was proposed in [24]. Lower and upper bounds of the convergence rate were also presented in terms of the number of pairwise comparison of function values, i.e., query complexity. The algorithm iteratively solves one-dimensional optimization problems like the coordinate descent method. However, practical performance of the optimization algorithm was not studied in that work.

The relationships between proposal and related works are summarized in Table 7.

In this part, we focus on optimization algorithms using the pairwise comparison oracle. In our algorithm, the convergence to the optimal solution is guaranteed, when the number of pairwise comparison tends to infinity. Our algorithm is regarded as a block coordinate descent method consisting of two steps: the direction estimate step and search step. In the direction estimate step, the search direction is determined. In the search step, the current solution is updated along the search direction with an appropriate step length. In our algorithm, the direction estimate step is easily parallelized. Therefore, it is expected that our algorithm effectively works even in large-scale optimization problems.

Let us summarize the contributions presented in this part [42].

1. We propose a block coordinate descent algorithm based on the pairwise comparison oracle, and point out that the algorithm is easily parallelized.
2. We derive an upper bound of the convergence rate in terms of the number of pairwise comparison of function values, i.e., query complexity. Moreover, we show that the query complexity of our algorithm is optimal for specific cases.
3. We show that our proposal is more efficient than the state-of-the-art methods using the same information (i.e. pairwise comparison oracle) through numerical experiments.

The rest of the part is organized as follows. In Section 8, we explain the problem setup and give some definitions. Section 9 is devoted to the main results. The convergence properties and query complexity of our algorithm are shown in the section. In Section 10, numerical examples are reported. Finally in Section 11, we conclude the part with the discussion on future works.

8 Preliminaries

In this section, we introduce the problem setup and prepare some definitions and notations used throughout the paper. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be σ -strongly convex on \mathbb{R}^n for a positive constant σ , if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the inequality

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\sigma}{2} \|\mathbf{x} - \mathbf{y}\|^2 \quad (98)$$

holds, where $\nabla f(\mathbf{x})$ and $\|\cdot\|$ denote the gradient of f at \mathbf{x} and the Euclidean norm, respectively. The function f is L -strongly smooth for a positive constant L , if $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. The gradient $\nabla f(\mathbf{x})$ of the L -strongly smooth function f is referred to as L -Lipschitz gradient. The class of σ -strongly convex and L -strongly smooth functions on \mathbb{R}^n is denoted as $\mathcal{F}_{\sigma,L}(\mathbb{R}^n)$. In the convergence analysis, we focus mainly on the optimization of objective functions in $\mathcal{F}_{\sigma,L}(\mathbb{R}^n)$.

We consider the following pairwise comparison oracle defined in [24].

Definition 7 (Pairwise comparison oracle). *The stochastic pairwise comparison (PC) oracle is a binary valued random variable $O_f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \{-1, +1\}$ defined as*

$$\Pr[O_f(\mathbf{x}, \mathbf{y}) = \text{sign}\{f(\mathbf{y}) - f(\mathbf{x})\}] \geq \frac{1}{2} + \min\{\delta_0, \mu|f(\mathbf{y}) - f(\mathbf{x})|^{\kappa-1}\}, \quad (99)$$

where $0 < \delta_0 \leq 1/2$, $\mu > 0$ and $\kappa \geq 1$. For $\kappa = 1$, without loss of generality $\mu \leq \delta_0 \leq 1/2$ is assumed.

When the equality

$$\Pr[O_f(\mathbf{x}, \mathbf{y}) = \text{sign}\{f(\mathbf{y}) - f(\mathbf{x})\}] = 1 \quad (100)$$

is satisfied for all \mathbf{x} and \mathbf{y} , we call O_f the deterministic PC oracle.

For $\kappa = 1$, the probability in (99) is not affected by the difference $|f(\mathbf{y}) - f(\mathbf{x})|$, and hence, the probability for the output of the PC oracle is not changed under any monotone transformation of f .

In [24], Jamieson et al. derived lower and upper bounds of convergence rate of an optimization algorithm using the stochastic PC oracle. The algorithm is referred to as the original PC algorithm in the present paper.

Under above preparations, our purpose is to find the minimizer \mathbf{x}^* of the objective function $f(\mathbf{x})$ in $\mathcal{F}_{\sigma,L}(\mathbb{R}^n)$ by using PC oracle. In the following section, we provide a DFO algorithm in order to solve the optimization problem and consider the convergence properties including query complexity.

9 Main Results

9.1 Algorithm

In Algorithm 6, we propose a DFO algorithm based on the PC oracle. In our algorithm, m coordinates out of n elements are updated in each iteration to efficiently cope with high dimensional problems. Algorithm 6 is referred to as BlockCD[n, m]. The original PC algorithm is recovered by setting $m = 1$. The PC oracle is used in the line search algorithm to solve one-dimensional optimization problems; the detailed line search algorithm is shown in Algorithm 7.

Algorithm 6 Block coordinate descent using PC oracle (BlockCD[n, m])

Input: initial point $\mathbf{x}_0 \in \mathbb{R}^n$, and accuracy in line search $\eta > 0$.

Initialize: set $t = 0$.

repeat

Choose m coordinates i_1, \dots, i_m out of n coordinates according to the uniform distribution.

(Direction estimate step)

Step D-1 Solve the one-dimensional optimization problems

$$\min_{\alpha \in \mathbb{R}} f(\mathbf{x}_t + \alpha \mathbf{e}_{i_k}), \quad k = 1, \dots, m, \quad (101)$$

within the accuracy $\eta/2$ using the PC-based line search algorithm shown in Algorithm 7, where \mathbf{e}_i denotes the i -th unit basis vector.

Then, obtain the numerical solutions $\alpha_{t,i_k}, k = 1, \dots, m$.

Step D-2 Set $\mathbf{d}_t = \sum_{k=1}^m \alpha_{t,i_k} \mathbf{e}_{i_k}$.

If \mathbf{d}_t is the zero vector, set $\mathbf{d}_t \leftarrow \frac{\eta}{2} \mathbf{e}_{i_1}$.

(Search step)

Step S-1 Apply Algorithm 7 to obtain a numerical solution β_t of

$$\min_{\beta} f(\mathbf{x}_t + \beta \mathbf{d}_t / \|\mathbf{d}_t\|)$$

within the accuracy η .

(Update step)

if $f(\mathbf{x}_t) \geq f(\mathbf{x}_t + \beta_t \mathbf{d}_t / \|\mathbf{d}_t\|)$ **then**

$\mathbf{x}_{t+1} = \mathbf{x}_t + \beta_t \mathbf{d}_t / \|\mathbf{d}_t\|$; $t \leftarrow t + 1$.

else

$\mathbf{x}_{t+1} = \mathbf{x}_t$; $t \leftarrow t + 1$.

end if

until A stopping criterion is satisfied.

Output: \mathbf{x}_t

The conditional statements in Update step are necessary to guarantee the monotone decrease of $f(\mathbf{x}_t)$ (see also proof of Theorem 6).

For $m = n$, the search direction \mathbf{d}_t in Algorithm 6 approximates that of a modified Newton method [40, Chap. 10], as shown below. In Step D-1 of the algorithm, one-dimensional optimization problems (101) are solved.

Let $\alpha_{t,i}^*$ be the optimal solution of (101) with $i_k = i$. Then, $\alpha_{t,i}^*$ will be close to the numerical solution $\alpha_{t,i}$. The Taylor expansion of the objective function leads to

$$f(\mathbf{x}_t + \alpha \mathbf{e}_i) = f(\mathbf{x}_t) + \alpha \mathbf{e}_i^\top \nabla f(\mathbf{x}_t) + \frac{\alpha^2}{2} \mathbf{e}_i^\top \nabla^2 f(\mathbf{x}_t) \mathbf{e}_i + o(\alpha^2),$$

where $\nabla^2 f(\mathbf{x}_t)$ is the Hessian matrix of f at \mathbf{x}_t . When the point \mathbf{x}_t is close to the optimal solution of $f(\mathbf{x})$, the optimal parameter $\alpha_{t,i}^*$ will be close to zero, implying that the higher order term $o(\alpha^2)$ in the above is negligible. Hence, $\alpha_{t,i}$ is approximated by the optimal solution of the quadratic approximation, i.e., $-(\nabla f(\mathbf{x}_t))_i / (\nabla^2 f(\mathbf{x}_t))_{ii}$. As a result, the search direction in BlockCD[n, n] approximates $-(\text{diag}(\nabla^2 f(\mathbf{x}_t)))^{-1} \nabla f(\mathbf{x}_t)$, where $\text{diag}(A)$ denotes the diagonal matrix, the diagonal elements of which are those of the square matrix A . In the modified Newton method, the

Algorithm 7 line search algorithm using PC oracle [24]

Input: current solution $\mathbf{x}_t \in \mathbb{R}^n$, search direction $\mathbf{d} \in \mathbb{R}^n$ and accuracy in line search $\eta > 0$.

Initialize: set $\alpha_0 = 0$, $\alpha_0^+ = \alpha_0 + 1$, $\alpha_0^- = \alpha_0 - 1$, $k = 0$.

[Step1]

if $O_f(\mathbf{x}_t, \mathbf{x}_t + \alpha_0^+ \mathbf{d}) > 0$ and $O_f(\mathbf{x}_t, \mathbf{x}_t + \alpha_0^- \mathbf{d}) < 0$ **then**

$\alpha_0^+ \leftarrow 0$

else if $O_f(\mathbf{x}_t, \mathbf{x}_t + \alpha_0^+ \mathbf{d}) < 0$ and $O_f(\mathbf{x}_t, \mathbf{x}_t + \alpha_0^- \mathbf{d}) > 0$ **then**

$\alpha_0^- \leftarrow 0$

end if

[Step2] (double-sign corresponds)

while $O_f(\mathbf{x}_t, \mathbf{x}_t + \alpha_k^\pm \mathbf{d}) < 0$ **do**

$\alpha_{k+1}^\pm \leftarrow 2\alpha_k^\pm$, $k \leftarrow k + 1$

end while

[Step3]

while $|\alpha_k^+ - \alpha_k^-| > \eta/2$ **do**

if $O_f(\mathbf{x}_t + \alpha_k \mathbf{d}, \mathbf{x}_t + \frac{1}{2}(\alpha_k + \alpha_k^+) \mathbf{d}) < 0$ **then**

$\alpha_{k+1} \leftarrow \frac{1}{2}(\alpha_k + \alpha_k^+)$, $\alpha_{k+1}^+ \leftarrow \alpha_k^+$, $\alpha_{k+1}^- \leftarrow \alpha_k$

else if $O_f(\mathbf{x}_t + \alpha_k \mathbf{d}, \mathbf{x}_t + \frac{1}{2}(\alpha_k + \alpha_k^-) \mathbf{d}) < 0$ **then**

$\alpha_{k+1} \leftarrow \frac{1}{2}(\alpha_k + \alpha_k^-)$, $\alpha_{k+1}^- \leftarrow \alpha_k^-$, $\alpha_{k+1}^+ \leftarrow \alpha_k$

else

(double-sign corresponds)

$\alpha_{k+1} \leftarrow \alpha_k$, $\alpha_{k+1}^\pm \leftarrow \frac{1}{2}(\alpha_k + \alpha_k^\pm)$

end if

end while

Output: α_t

Hessian matrix in the Newton method is replaced with a positive definite matrix to reduce the computation cost. Using only the diagonal part of the Hessian matrix is a popular choice in the modified Newton method.

Figure 4 demonstrates an example of the optimization process of both the original PC algorithm and our algorithm. The original PC algorithm updates the numerical solution along a randomly chosen coordinate in each iteration. Hence, many iterations are required to get close to the optimal solution. On the other hand, in our algorithm, a solution can move along a oblique direction. Therefore, our algorithm can get close to the optimal solution with less iterations than the original PC algorithm. It should be noted here that although our algorithm requires higher computational costs in one iteration compared to original PC algorithm, parallelizability of direction estimate step reduces the practical computation time and our algorithm as a whole can efficiently find the optimal solution. The detail of parallelization of BlockCD[n, m] is described in section 10.

9.2 Convergence Properties of our Algorithm under Deterministic Oracle

We now provide an upper bound of the convergence rate of our algorithm using the deterministic PC oracle (100).

Let us denote the minimizer of f as \mathbf{x}^* .

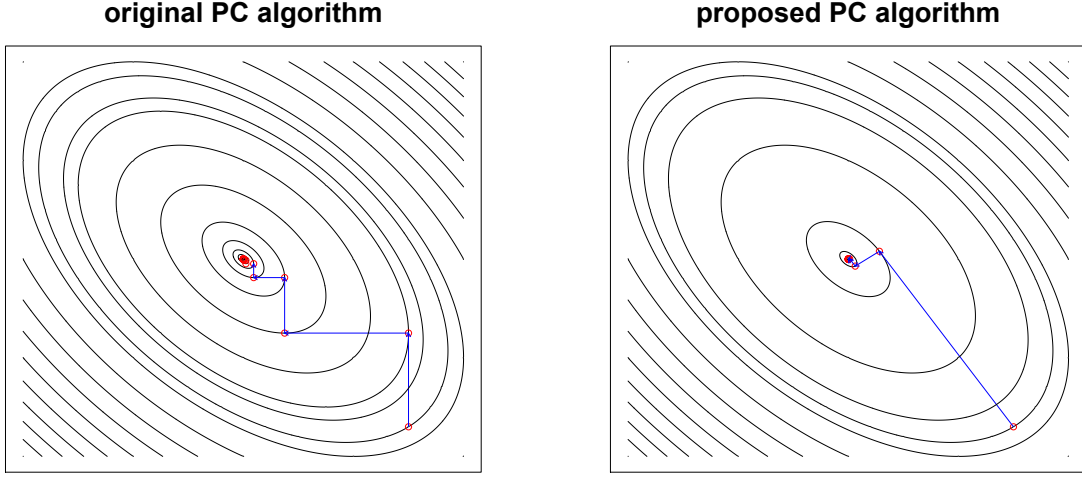


Figure 4: A behavior of the algorithms on the contour of the quadratic objective function $x_1^2 + x_2^2 + x_1x_2$ with same initialization. Left panel: Jamieson et al.'s original PC algorithm. Right panel: proposed algorithm.

Theorem 6. Suppose $f \in \mathcal{F}_{\sigma,L}(\mathbb{R}^n)$, and set $\varepsilon > 0$ arbitrarily. Define γ and η as

$$\gamma = \frac{\sigma/L}{52} \left(\frac{1 - \sqrt{1 - \sigma/L}}{1 + \sqrt{1 - \sigma/L}} \right)^2, \quad \eta = \sqrt{\frac{\sigma\varepsilon}{8nL^2(1 + n/m\gamma)}}. \quad (102)$$

Let us define T_0 be

$$T_0 = \left\lceil \frac{n}{m\gamma} \log \frac{(f(\mathbf{x}_0) - f(\mathbf{x}^*))(1 + \frac{n}{m\gamma})}{\varepsilon} \right\rceil. \quad (103)$$

For $T \geq T_0$, we have $\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \leq \varepsilon$, where the expectation is taken with respect to the random choice of coordinates i_1, \dots, i_k to be updated in BlockCD $[n, m]$ with the accuracy η .

Proof. The optimal solution of f is denoted as \mathbf{x}^* . Let us define ε' be $\varepsilon/(1 + \frac{n}{m\gamma})$. If $f(\mathbf{x}_t) - f(\mathbf{x}^*) < \varepsilon'$ holds in the algorithm, we obtain $f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*) < \varepsilon'$, since the function value is non-increasing in each iteration of the algorithm

Next, we assume $\varepsilon' \leq f(\mathbf{x}_t) - f(\mathbf{x}^*)$. In the following, we use the inequality

$$f(\mathbf{x}_t + \beta_t \mathbf{d}_t / \|\mathbf{d}_t\|) \leq f(\mathbf{x}_t) - \frac{|\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t|^2}{2L\|\mathbf{d}_t\|^2} + \frac{L}{2}\eta^2$$

that is proved in [24]. For the i -th coordinate, let us define the functions $g_{\text{low},i}(\alpha)$ and $g_{\text{up},i}(\alpha)$ as

$$g_{\text{low},i}(\alpha) = f(\mathbf{x}_t) + \frac{\partial f(\mathbf{x}_t)}{\partial x_i} \alpha + \frac{\sigma}{2} \alpha^2, \quad \text{and} \quad g_{\text{up},i}(\alpha) = f(\mathbf{x}_t) + \frac{\partial f(\mathbf{x}_t)}{\partial x_i} \alpha + \frac{L}{2} \alpha^2.$$

Then, we have

$$g_{\text{low},i}(\alpha) \leq f(\mathbf{x}_t + \alpha \mathbf{e}_i) \leq g_{\text{up},i}(\alpha).$$

Let $\alpha_{\text{up},i}$ and α_i^* be the minimum solution of $\min_{\alpha} g_{\text{up},i}(\alpha)$ and $\min_{\alpha} f(\mathbf{x}_t + \alpha \mathbf{e}_i)$, respectively. In particular, $\alpha_{\text{up},i}$ can be written explicitly. Then, we obtain

$$g_{\text{low},i}(\alpha_i^*) \leq f(\mathbf{x}_t + \alpha_i^* \mathbf{e}_i) \leq f(\mathbf{x}_t + \alpha_{\text{up},i} \mathbf{e}_i) \leq g_{\text{up},i}(\alpha_{\text{up},i}).$$

The inequality $g_{\text{low},i}(\alpha_i^*) \leq g_{\text{up},i}(\alpha_{\text{up},i})$ and the concrete form of $\alpha_{\text{up},i}$ yield that α_i^* lies between $-c_0 \frac{\partial f(\mathbf{x}_t)}{\partial x_i}$ and $-c_1 \frac{\partial f(\mathbf{x}_t)}{\partial x_i}$, where c_0 and c_1 are defined as

$$c_0 = (1 - \sqrt{1 - \sigma/L})/\sigma, \quad c_1 = (1 + \sqrt{1 - \sigma/L})/\sigma.$$

Here, $0 < c_0 \leq c_1$ holds. Each component of the search direction $\mathbf{d}_t = (d_1, \dots, d_n) \neq \mathbf{0}$ in Algorithm 6 satisfies $|d_i - \alpha_i^*| \leq \eta$ if $i = i_k$ and otherwise $d_i = 0$. For $I = \{i_1, \dots, i_m\} \subset \{1, \dots, n\}$, let $\|\mathbf{a}\|_I^2$ of the vector $\mathbf{a} \in \mathbb{R}^n$ be $\sum_{i \in I} a_i^2$. The vector $(\alpha_1^*, \dots, \alpha_n^*)$ is denoted as $\boldsymbol{\alpha}^*$. Then, the triangle inequality leads to

$$\begin{aligned} \|\mathbf{d}_t\| &\leq \|\boldsymbol{\alpha}^*\|_I + \|\mathbf{d}_t - \boldsymbol{\alpha}^*\|_I \leq c_1 \|\nabla f(\mathbf{x}_t)\|_I + \sqrt{m}\eta, \\ |\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t| &\geq \left| \sum_{i \in I} (\nabla f(\mathbf{x}_t))_i \alpha_i^* \right| - \left| \sum_{i \in I} (\nabla f(\mathbf{x}_t))_i (d_i - \alpha_i^*) \right| \\ &\geq c_0 \|\nabla f(\mathbf{x}_t)\|_I^2 - \sqrt{m}\eta \|\nabla f(\mathbf{x}_t)\|_I. \end{aligned}$$

The assumption $\varepsilon' \leq f(\mathbf{x}_t) - f(\mathbf{x}^*)$ leads to

$$2\sigma\varepsilon' \leq 2\sigma(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \|\nabla f(\mathbf{x}_t)\|^2,$$

in which the second inequality is derived from (9.9) in [6]. The above inequality and $1/4L^2 \leq c_0^2$ yield

$$\eta = \sqrt{\frac{\varepsilon'\sigma}{8L^2n}} \leq c_0 \sqrt{\frac{\sigma\varepsilon'}{2n}} \leq c_0 \frac{\|\nabla f(\mathbf{x}_t)\|}{2\sqrt{n}}.$$

Hence, we obtain

$$\begin{aligned} \|\mathbf{d}_t\| &\leq c_1 \|\nabla f(\mathbf{x}_t)\|_I + \frac{c_0}{2} \sqrt{\frac{m}{n}} \|\nabla f(\mathbf{x}_t)\|, \\ |\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t| &\geq \left[c_0 \|\nabla f(\mathbf{x}_t)\|_I^2 - \frac{c_0}{2} \sqrt{\frac{m}{n}} \|\nabla f(\mathbf{x}_t)\| \|\nabla f(\mathbf{x}_t)\|_I \right]_+, \end{aligned}$$

where $[x]_+ = \max\{0, x\}$ for $x \in \mathbb{R}$. Let $Z = \sqrt{\frac{n}{m}} \|\nabla f(\mathbf{x}_t)\|_I / \|\nabla f(\mathbf{x}_t)\|$ be a non-negative valued random variable defined from the random set I , and define the non-negative value k as $k = c_0/c_1 \leq 1$. A lower bound of the expectation of $(|\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t| / \|\mathbf{d}_t\|)^2$ with respect to the distribution of I is given as

$$\begin{aligned} \mathbb{E}_I \left[\left(\frac{|\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t|}{\|\mathbf{d}_t\|} \right)^2 \right] &\geq \mathbb{E}_I \left[\left(\frac{\left[c_0 \|\nabla f(\mathbf{x}_t)\|_I^2 - \frac{c_0}{2} \sqrt{\frac{m}{n}} \|\nabla f(\mathbf{x}_t)\| \|\nabla f(\mathbf{x}_t)\|_I \right]_+}{c_1 \|\nabla f(\mathbf{x}_t)\|_I + \frac{c_0}{2} \sqrt{\frac{m}{n}} \|\nabla f(\mathbf{x}_t)\|} \right)^2 \right] \\ &= k^2 \frac{m}{n} \|\nabla f(\mathbf{x}_t)\|^2 \mathbb{E}_I \left[Z^2 \frac{[Z - 1/2]_+^2}{(Z + k/2)^2} \right] \\ &\geq k^2 \frac{m}{n} \|\nabla f(\mathbf{x}_t)\|^2 \mathbb{E}_I \left[Z^2 \frac{[Z - 1/2]_+^2}{(Z + 1/2)^2} \right]. \end{aligned}$$

Here, we recall that $i_k \in I$ is uniformly distributed. The random variable Z is non-negative, and $\mathbb{E}_I[Z^2] = 1$ holds. Thus, Lemma 1 below leads to

$$\mathbb{E}_I \left[\left(\frac{|\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t|}{\|\mathbf{d}_t\|} \right)^2 \right] \geq \frac{k^2}{52} \frac{m}{n} \|\nabla f(\mathbf{x}_t)\|^2.$$

Combining the above inequality with the case of $f(\mathbf{x}_t) - f(\mathbf{x}^*) < \varepsilon'$, we obtain the conditional expectation of $f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*)$ for given $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{t-1}$ as follows.

$$\begin{aligned} & \mathbb{E}[f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*) | \mathbf{d}_0, \dots, \mathbf{d}_{t-1}] \\ & \leq \mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \varepsilon'] \cdot \left[f(\mathbf{x}_t) - f(\mathbf{x}^*) - \frac{k^2}{104L} \frac{m}{n} \|\nabla f(\mathbf{x}_t)\|^2 + \frac{L\eta^2}{2} \right] \\ & \quad + \mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) < \varepsilon'] \cdot \varepsilon' \\ & \leq \mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \varepsilon'] \cdot \left[\left(1 - \frac{m}{n}\gamma\right)(f(\mathbf{x}_t) - f(\mathbf{x}^*)) + \frac{L\eta^2}{2} \right] \\ & \quad + \mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) < \varepsilon'] \cdot \varepsilon'. \end{aligned} \tag{104}$$

Taking the expectation with respect to all $\mathbf{d}_0, \dots, \mathbf{d}_t$ yields

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_{t+1}) - f(\mathbf{x}^*)] & \leq \left(1 - \frac{m}{n}\gamma\right) \mathbb{E}[\mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \varepsilon'](f(\mathbf{x}_t) - f(\mathbf{x}^*))] \\ & \quad + \mathbb{E}[\mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \varepsilon']] \frac{L\eta^2}{2} + \mathbb{E}[\mathbf{1}[f(\mathbf{x}_t) - f(\mathbf{x}^*) < \varepsilon']] \varepsilon' \\ & \leq \left(1 - \frac{m}{n}\gamma\right) \mathbb{E}[f(\mathbf{x}_t) - f(\mathbf{x}^*)] + \max \left\{ \frac{L\eta^2}{2}, \varepsilon' \right\}. \end{aligned}$$

Since $0 < \gamma < 1$ and $\max\{L\eta^2/2, \varepsilon'\} = \varepsilon'$ hold, for $\Delta_T = \mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)]$ we have

$$\Delta_T - \frac{n}{m} \frac{\varepsilon'}{\gamma} \leq \left(1 - \frac{m}{n}\gamma\right) \left(\Delta_{T-1} - \frac{n}{m} \frac{\varepsilon'}{\gamma}\right) \leq \left(1 - \frac{m}{n}\gamma\right)^T \Delta_0.$$

When T is greater than T_0 in (103), we obtain $\left(1 - \frac{m}{n}\gamma\right)^T \Delta_0 \leq \varepsilon'$ and

$$\Delta_T \leq \varepsilon' \left(1 + \frac{n}{m\gamma}\right) = \varepsilon.$$

□

Remark 1. For $m = 1$, the exact evaluation of $\mathbb{E}[(\nabla f(\mathbf{x}_t)^\top \mathbf{d}_t / \|\mathbf{d}_t\|)^2]$ is possible. Hence, we do not need to introduce the threshold ε' to evaluate the perturbation of the norm $\|\mathbf{d}_t\|$ such as (104). Arbitrary small ε' is available, and $\max\{L\eta^2/2, \varepsilon'\}$ in the above proof becomes $L\eta^2/2$. As a result, the faster convergence rate shown in [24] is obtained for $m = 1$.

Lemma 1. Let Z be a non-negative random variable satisfying $\mathbb{E}[Z^2] = 1$. Then, we have

$$\mathbb{E} \left[Z^2 \frac{[Z - 1/2]_+^2}{(Z + 1/2)^2} \right] \geq \frac{1}{52}.$$

Proof. For $z \geq 0$ and $\delta \geq 0$, we have the inequality

$$\frac{[z - 1/2]_+^2}{(z + 1/2)^2} \geq \frac{\delta^2}{(1 + \delta)^2} \mathbf{1}[z \geq 1/2 + \delta].$$

Then, we get

$$\begin{aligned} \mathbb{E} \left[Z^2 \frac{[Z - 1/2]_+^2}{(Z + 1/2)^2} \right] &\geq \frac{\delta^2}{(1 + \delta)^2} \mathbb{E}[Z^2 \mathbf{1}[Z \geq 1/2 + \delta]] \\ &= \frac{\delta^2}{(1 + \delta)^2} \mathbb{E}[Z^2 (1 - \mathbf{1}[Z < 1/2 + \delta])] \\ &= \frac{\delta^2}{(1 + \delta)^2} (1 - \mathbb{E}[Z^2 \mathbf{1}[Z < 1/2 + \delta]]) \\ &\geq \frac{\delta^2}{(1 + \delta)^2} (1 - (1/2 + \delta)^2 \Pr(Z < 1/2 + \delta)) \\ &\geq \frac{\delta^2}{(1 + \delta)^2} (1 - (1/2 + \delta)^2). \end{aligned}$$

By setting δ appropriately, we obtain

$$\mathbb{E}_I \left[Z^2 \frac{[Z - 1/2]_+^2}{(Z + 1/2)^2} \right] \geq \frac{1}{52}.$$

□

Note that any monotone transformation of the objective function does not affect the output of the deterministic PC oracle. Hence, the theorem is applicable for the composite of any function f and a non-decreasing function g , $g \circ f$, if $g \circ f \in \mathcal{F}_{\sigma, L}(\mathbb{R}^n)$.

Remark 2. As shown in (103), the number of iterations to achieve the accuracy ε is of the order $T_0 = \tilde{O}(\lambda^3 \log(1/\varepsilon))$ for fixed n and m , where λ denotes the condition number of the Hessian matrix, $\lambda = L/\sigma$. On the other hand, an upper bound of the iteration number for non-linear optimization algorithms such as the steepest descent method is given by $O(\lambda \log(1/\varepsilon))$ [40, Chap. 8.6.3]. The cubic order of λ in our method comes from the worst-case analysis of the numerical error in the search direction \mathbf{d}_t caused by multiple inexact line searches. In the standard setup of non-linear optimization problems, such a large numerical error in the search direction is not taken into account.

Remark 3. Theorem 6 says that for arbitrary ε , if we take the tolerance η , of the line search as in (102), it is guaranteed that the average difference between optimal solution and current solution is smaller than ε . In contrast, if we take η arbitrary, it can be guaranteed that

$$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \leq \frac{8nL^2\eta^2(1 + \frac{n}{m\gamma})}{\sigma} \quad (105)$$

for $T \geq T_0$, where

$$T_0 = \left\lceil \frac{n}{m\gamma} \log \frac{(f(\mathbf{x}_0) - f(\mathbf{x}^*))\sigma}{8nL^2\eta^2} \right\rceil.$$

Here, when we get η small, the average difference compared with the optimal function value quadratically decreases with respect to η while the number T_0 of total queries logarithmically increases at most. Thus, it means that we can get tight bound with only slight increment of total queries if we set the tolerance of a line search small.

Remark 4. To the best of our knowledge, it is still an open problem to determine an appropriate stopping criterion of the algorithm under the pairwise comparison framework. The first order condition, $\|\nabla f(\mathbf{x}_T)\| < \text{tolerance}$, or the difference of function values, $|f(\mathbf{x}_T) - f(\mathbf{x}_{T-1})| < \text{tolerance}$, can not be employed under the pairwise comparison oracle because function values and gradients are not available. Yue et al. [60] or Jamieson et al. [24] also do not explicitly describe about the stopping criterion. In practice, we employ the norm of the difference of iterative solutions, $\|\mathbf{x}_{t+1} - \mathbf{x}_t\| < \text{tolerance}$.

Let us consider the convergence accuracy of the numerical solution \mathbf{x}_T . From the strong convexity (98), the inequality

$$\|\mathbf{x} - \mathbf{x}^*\|^2 \leq \frac{2}{\sigma}(f(\mathbf{x}) - f(\mathbf{x}^*))$$

holds. Thus, for $T \geq T_0$, we have

$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^*\|^2] \leq \mathbb{E}[\|\mathbf{x}_T - \mathbf{x}^*\|^2] \leq \frac{2}{\sigma}\varepsilon = 16n\left(\frac{L}{\sigma}\right)^2\left(1 + \frac{n}{m\gamma}\right)\eta^2.$$

9.3 Query Complexity

Let $\hat{\mathbf{x}}_Q$ be the output of BlockCD[n, m] after Q pairwise comparison queries. To solve the one-dimensional optimization problem within the accuracy $\eta/2$, the sufficient number of the call of PC-oracle is

$$K_0 = \left\lceil 2 \log_2 \frac{2^{10} L \Delta_0}{\sigma^2 \eta^2} \right\rceil = \left\lceil \frac{2}{\log 2} \log \frac{2^{13} (L/\sigma)^3 \Delta_0 (1 + \frac{n}{m\gamma})}{\varepsilon} \right\rceil, \quad (106)$$

as shown in [24] where $\Delta_0 = f(\mathbf{x}_0) - f(\mathbf{x}^*)$. Hence, if the inequality $Q \geq T_0 K_0 (m + 2)$ holds, Theorem 6 assures that the numerical solution $\hat{\mathbf{x}}_Q$ based on Q queries satisfies

$$\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon.$$

The following Corollary shows the query complexity of Algorithm 6.

Corollary 1. For any Q such that

$$\frac{n}{c_0^2} \frac{m+2}{m\gamma} \left[\log \left(2^{13.5} \left(\frac{L}{\sigma} \right)^3 n \Delta_0 \left(1 + \frac{n}{m\gamma} \right) \right) \right]_+^2 \leq Q,$$

we have

$$\mathbb{E}[f(\mathbf{x}_Q) - f(\mathbf{x}^*)] \leq \exp \left\{ -c_0 \sqrt{\gamma \times \frac{m}{m+2} \times \frac{Q}{n}} \right\} \quad (107)$$

Proof. Let us replace T_0 and K_0 with

$$T_0 = \frac{n}{m\gamma} \log \frac{\Delta_0 (1 + \frac{n}{m\gamma})}{\varepsilon} + 1 = \frac{n}{m\gamma} \log \frac{e^{m\gamma/n} \Delta_0 (1 + \frac{n}{m\gamma})}{\varepsilon}, \quad (108)$$

$$K_0 = \frac{2}{\log 2} \log \frac{2^{13} (L/\sigma)^3 n \Delta_0 (1 + \frac{n}{m\gamma})}{\varepsilon} + 1 = \frac{2}{\log 2} \log \frac{2^{13.5} (L/\sigma)^3 n \Delta_0 (1 + \frac{n}{m\gamma})}{\varepsilon}, \quad (109)$$

respectively. Note that $e^{m\gamma/n}\Delta_0(1 + \frac{n}{m\gamma}) \leq 2^{13.5}(L/\sigma)^3 n\Delta_0(1 + \frac{n}{m\gamma})$ holds because of $\sigma \leq L$, $\gamma < 1$ and $1 \leq m \leq n$. Suppose that

$$2^{13.5}(L/\sigma)^3 n\Delta_0(1 + \frac{n}{m\gamma}) \leq \frac{1}{\varepsilon} \quad (110)$$

holds. Then we have

$$T_0 \leq \frac{2n}{m\gamma} \log \frac{1}{\varepsilon}, \quad K_0 \leq \frac{4}{\log 2} \log \frac{1}{\varepsilon},$$

and thus,

$$(m+2)T_0K_0 \leq \frac{8}{\log 2} \frac{(m+2)n}{m\gamma} \left(\log \frac{1}{\varepsilon} \right)^2 =: Q_0$$

holds. Theorem 6 leads to

$$\mathbb{E}[f(\mathbf{x}_{Q_0}) - f(\mathbf{x}^*)] \leq \varepsilon = \exp \left\{ -c_0 \sqrt{\frac{m\gamma}{m+2} \times \frac{Q_0}{n}} \right\},$$

where $c_0 = \sqrt{\frac{\log 2}{8}}$. The condition (110) is expressed as

$$\frac{n}{c_0^2} \frac{m+2}{m\gamma} \left[\log \left(2^{13.5} \left(\frac{L}{\sigma} \right)^3 n\Delta_0 \left(1 + \frac{n}{m\gamma} \right) \right) \right]_+^2 \leq Q_0,$$

where $[a]_+ = \max\{a, 0\}$. Note that the left-hand side of the above inequality is determined from the problem setup (i.e. σ, L, n), initial point \mathbf{x}_0 , and the parallelization parameter m of Algorithm 6. \square

Remark 5. $\gamma = O((\sigma/L)^3)$ is regarded as the reciprocal of the condition number for the objective function.

Remark 6. It is also guaranteed that there exists a constant $C > 0$ such that for any Q (even if the assumption in Corollary 1 is not held),

$$\mathbb{E}[f(\mathbf{x}_Q) - f(\mathbf{x}^*)] \leq C \exp \left\{ -c_0 \sqrt{\gamma \times \frac{m}{m+2} \times \frac{Q}{n}} \right\} \quad (111)$$

holds.

The above bound is of the same order of the convergence rate for the original PC algorithm up to polylog factors. On the other hand, a lower bound presented in [24] is of order $e^{-cQ/n}$ with a positive constant c up to polylog factors, when the PC oracle with $\kappa = 1$ is used.

In Theorem 6, it is assumed that the objective function is strongly convex and gradient Lipschitz. In a realistic situation, we usually do not have the knowledge of the class parameters σ and L of the unknown objective function. Moreover, strong convexity and gradient Lipschitzness on the whole space \mathbb{R}^n is too strong.

In the following corollary, we relax the assumption in Theorem 6 and prove the convergence property of our algorithm without explicitly imposing strong convexity and strong smoothness.

Corollary 2. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a twice continuously differentiable convex function with non-degenerate Hessian on \mathbb{R}^n and have the minimizer \mathbf{x}^* . Then, there is a constant c and sufficiently small η^1 , such that the output $\hat{\mathbf{x}}_Q$ of BlockCD[n, m] satisfies (107).*

Proof. For the output $\hat{\mathbf{x}}_Q$ of BlockCD[n, m], $f(\hat{\mathbf{x}}_Q) \geq f(\hat{\mathbf{x}}_{Q+1})$ holds, and thus, the sequence $\{\hat{\mathbf{x}}_Q\}_{Q \in \mathbb{N}}$ is included in

$$C(\mathbf{x}_0) := \{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) \leq f(\mathbf{x}_0)\}.$$

Since f is convex and continuous, $C(\mathbf{x}_0)$ is convex and closed. Moreover, since f is convex and it has non-degenerate Hessian, the Hessian is positive definite, and thus, f is strictly convex. Then $C(\mathbf{x}_0)$ is bounded as follows. We set the minimal directional derivative along the radial direction from \mathbf{x}^* over the unit sphere around \mathbf{x}^* as

$$b := \min_{\|u\|=1} \nabla f(\mathbf{x}^* + u) \cdot u.$$

Then, b is strictly positive and the following holds for any $\mathbf{x} \in C(\mathbf{x}_0)$ such that $\|\mathbf{x} - \mathbf{x}^*\| \geq 1$,

$$b\|\mathbf{x} - \mathbf{x}^*\| + (f(\mathbf{x}^*) - b) \leq f(\mathbf{x}) \leq f(\mathbf{x}_0).$$

Thus we have

$$C(\mathbf{x}_0) \subset \left\{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{x}^*\| \leq 1 + \frac{f(\mathbf{x}_0) - f(\mathbf{x}^*)}{b} \right\}. \quad (112)$$

Since the right hand side of (112) is a bounded ball, $C(\mathbf{x}_0)$ is also bounded. Thus, $C(\mathbf{x}_0)$ is a convex compact set.

Since f is twice continuously differentiable, the Hessian matrix $\nabla^2 f(\mathbf{x})$ is continuous with respect to $\mathbf{x} \in \mathbb{R}^n$. By the positive definiteness of the Hessian matrix, the minimum and maximum eigenvalues $e_{\min}(\mathbf{x})$ and $e_{\max}(\mathbf{x})$ of $\nabla^2 f(\mathbf{x})$ are continuous and positive. Therefore, there are the positive minimum value σ of $e_{\min}(\mathbf{x})$ and maximum value L of $e_{\max}(\mathbf{x})$ on the compact set $C(\mathbf{x}_0)$. It means that f is σ -strongly convex and L -Lipschitz on $C(\mathbf{x}_0)$. Thus, the same argument to obtain (107) can be applied for f .

□

9.4 Generalization to Stochastic Pairwise Comparison Oracle

In this subsection, we generalize the results described in the previous section to the stochastic setting. Our strategy is to come down to the deterministic situation since for the deterministic PC oracle, the monotone decreasing of function value is guaranteed. As such, in Algorithm 1, the decrease of function value is guaranteed in high probability even if the deterministic PC oracle is replaced with the stochastic PC oracle. More specifically, for stochastic PC oracle, we derive the number of queries that is required to obtain the correct PC (i.e. deterministic PC) in high probability. Then, the total number of query for stochastic PC oracle is computed by the total number of query for deterministic PC oracle times the above number of queries.

In Algorithm 8, the query $O_f(\mathbf{x}, \mathbf{y})$ is repeated under the stochastic PC oracle. The reliability of line search algorithm based on stochastic PC oracle (99) was investigated by [24, 33].

¹“sufficiently” means that η is smaller than or equal to the quantity of the right-hand-side of (102). Although the quantity can not be explicitly computed (since L and σ are unknown), we can achieve the order optimal by taking η smaller and smaller.

Algorithm 8 Repeated querying subroutine ([24, 33])

Input: $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\delta > 0$.
Initialize: set $n_0 = 1$ and draw a stochastic PC oracle $O_f(\mathbf{x}, \mathbf{y})$.
for $k = 0, 1, \dots$ **do**
 p_k = frequency of +1 in all draw of $O_f(\mathbf{x}, \mathbf{y})$ so far
 $I_k = \left[p_k - \sqrt{\frac{(k+1)\log(2/\delta)}{2^k}}, p_k + \sqrt{\frac{(k+1)\log(2/\delta)}{2^k}} \right]$
 if $\frac{1}{2} \notin I_k$ **then**
 break
 else
 draw $O_f(\mathbf{x}, \mathbf{y})$ n_k more times, and set $n_{k+1} = 2n_k$.
 end if
end for
if $p_k + \sqrt{\frac{(k+1)\log(2/\delta)}{2^k}} \leq \frac{1}{2}$ **then**
 return -1
else
 return +1
end if

Lemma 2 ([24, 33]). *For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with $p = \Pr[O_f(\mathbf{x}, \mathbf{y}) = \text{sign}\{f(\mathbf{y}) - f(\mathbf{x})\}]$, the repeated querying subroutine in Algorithm 8 correctly identifies the sign of $\mathbb{E}[O_f(\mathbf{x}, \mathbf{y})]$ with probability $1 - \delta$, and requests no more than*

$$\frac{\log 2/\delta}{4|1/2 - p|^2} \log_2 \left(\frac{\log 2/\delta}{4|1/2 - p|^2} \right) \quad (113)$$

queries.

It should be noted here that, in this paper, $\text{sign}\{\mathbb{E}[O_f(\mathbf{x}, \mathbf{y})]\} = \text{sign}\{f(\mathbf{y}) - f(\mathbf{x})\}$ always holds because $p > 1/2$ from (99). In [24], a modified line search algorithm using a ternary search instead of bisection search was proposed so that $|1/2 - p|$ in Lemma 2 is lower bounded by $\mu \left(\frac{\sigma\eta^2}{18} \right)^{\kappa-1}$ for arbitrary $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ where η is an accuracy of line search computed by (102) and μ and κ are oracle parameters defined in (99). Then one can find that, the total number of queries required to estimate the correct sign of $\mathbb{E}[O_f(\mathbf{x}, \mathbf{y})]$ for arbitrary $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with probability more than $1 - \delta$, is at most

$$\begin{aligned}
 Q(\delta) &= \frac{\log 2/\delta}{4 \left(\mu (\sigma\eta^2/18)^{\kappa-1} \right)^2} \log_2 \left(\frac{\log 2/\delta}{4 \left(\mu (\sigma\eta^2/18)^{\kappa-1} \right)^2} \right) \\
 &= O \left(\log \frac{1}{\delta} \cdot \log \log \frac{1}{\delta} \right).
 \end{aligned}$$

Intuitively, $Q(\delta)$ is the number of queries that is required in order to treat the stochastic PC oracle as the deterministic PC oracle. Let δ' be a probability parameter such that we want to guarantee $\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon$ with probability more than $1 - \delta'$ under stochastic PC oracle. Note that if we call $J_0 Q(\delta')$ stochastic oracles, we can guarantee $\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon$ with probability $(1 - \delta')^{J_0}$, where

$$J_0 = T_0 K_0 (m + 2) \quad (114)$$

and T_0 and K_0 are defined in (108). This implies that if we call $J_0 Q(\delta'/J_0)$ stochastic oracle, we can guarantee $\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon$ with probability $(1 - \delta'/J_0)^{J_0}$ that is greater than $1 - \delta'$. That is, if we call

$$J_0 Q\left(\frac{\delta'}{J_0}\right) \approx O\left(J_0 \log \frac{J_0}{\delta'}\right) = \tilde{O}\left(J_0 \log \frac{1}{\delta'}\right) \quad (115)$$

stochastic oracle, we can guarantee $\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon$ with probability more than $1 - \delta'$. Hence, we have to take δ in Lemma 2 such that

$$\delta \leq \frac{\delta'}{J_0} \quad (116)$$

to achieve the accuracy ε .

Then, if we take δ which satisfies (116), it is sufficient to call

$$Q \geq J_0 Q(\delta) \quad (117)$$

stochastic oracle for guaranteeing $\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \varepsilon$ with probability $1 - \delta'$.

Using the relation (102) between η and ε and solve the inequality (117) with respect to ε , we have the following upper bounds for stochastic setting:

$$\mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}^*)] \leq \begin{cases} \exp\left\{-\frac{c_1}{\log n} \sqrt{\frac{Q}{n}}\right\}, & \kappa = 1, \\ c_2 \frac{n^2}{m} \left(\frac{n}{Q}\right)^{1/(2\kappa-2)}, & \kappa > 1, \end{cases} \quad (118)$$

where c_1 and c_2 are constant depending on oracle parameters, L/σ and $f(\mathbf{x}_0) - f(\mathbf{x}^*)$ as well as $1/\delta$ poly-logarithmically. If m and n are of the same order in the case of $\kappa > 1$, the bound (118) coincides with that shown in Theorem 2 of [24].

Note that our upper bound in (118) is optimal with respect to the number of queries Q when $\kappa > 1$. This result comes from the lower bound derived by Theorem 7 [24].

Theorem 7 (Theorem 1 in [24]). *For $n \geq 8$ and sufficiently large Q ,*

$$\inf_{\hat{\mathbf{x}}_Q} \sup_{f \in \mathcal{F}_{\sigma, L}(\mathbb{R}^n)} \mathbb{E}[f(\hat{\mathbf{x}}_Q) - f(\mathbf{x}_f^*)] \geq \begin{cases} C_1 \exp\{-C_2 \frac{Q}{n}\} & \text{if } \kappa = 1 \\ C_3 \left(\frac{n}{Q}\right)^{1/(2\kappa-2)} & \text{if } \kappa > 1 \end{cases} \quad (119)$$

where the constants C_1, C_2, C_3 depend on the oracle and function class parameters.

Although Jamieson et al. also achieved the optimal rate with respect to the number of queries by PCSCD algorithm when $\kappa > 1$ (Theorem 2 in [24]), our proposal, BlockCD algorithm, practically outperforms PCSCD algorithm by an effect of parallelization. We explain this in section 4.2 in detail.

Remark 7. *In the case of $\kappa > 1$, the upper bound of the query complexity (118) with $m = 1$ is of the order $O(n^2(n/Q)^{1/(2\kappa-2)})$, which is larger than the upper bound $O(n(n/Q)^{1/(2\kappa-2)})$ presented in Theorem 2 of [24]. In our algorithm, not only the step length β but also the search direction \mathbf{d}_t is affected by numerical errors in the line search to the directions of \mathbf{e}_{i_k} , $k = 1, \dots, m$. Though (118) provides the upper bound for any $m = 1, \dots, n$, that is not necessarily tight for a small m . A technical point for $m = 1$ is explained in Remark 1 in Appendix ???. On the other hand, (118) with $m = O(n)$ is $O(n(n/Q)^{1/(2\kappa-2)})$, which is of the same order of the original PC algorithm. Thus, the parallel implementation of the multiple line searches will be of great help to reducing the computation time, as will be shown in Section 10.*

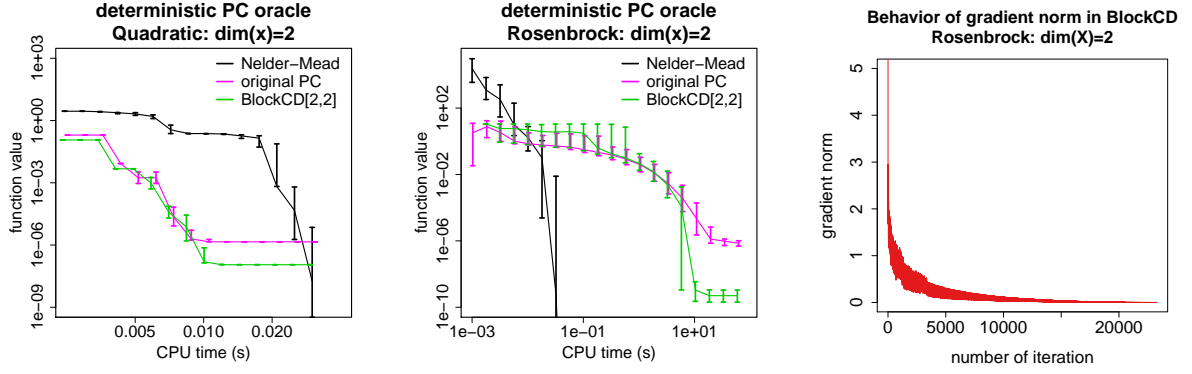


Figure 5: Left panel: two-dimensional quadratic function. Mid panel: two-dimensional Rosenbrock function. The Nelder-Mead method, BlockCD[2, 1], and BlockCD[2, 2] are compared. For each algorithm, the median of the function value is depicted to the CPU time (s). The vertical bar shows the percentile 30% to 70%. Right panel: The average behavior of the gradient norm of two-dimensional Rosenbrock function of in stochastic PC-oracle setting. Although Rosenbrock function is a non-convex function, we can see that the BlockCD algorithm practically converges to a stationary point.

10 Numerical Experiments

We present numerical experiments in which the proposed method in Algorithm 6 was compared with the Nelder-Mead algorithm [44], dueling bandit gradient descent method [60], and original PC algorithm [24], i.e., BlockCD[n , 1] of Algorithm 6. Here, the PC oracle was used in all the optimization algorithms.

10.1 Existing Methods and Preliminary Experiments

Nelder-Mead Algorithm : It is well-known that the Nelder-Mead method efficiently works in low dimensional problems. Indeed, in our preliminary experiments for two-dimensional optimization problems, the Nelder-Mead method showed a good convergence property compared to the other methods such as BlockCD[2, m] with $m = 1, 2$ (see Figure 5). We tested optimization methods on the quadratic function $f(x) = x^T A x$ with a random positive definite matrix A , and two-dimensional Rosenbrock function, $f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$. Rosenbrock function is a famous non-convex test function. The right panel in Figure 5 shows the average behavior of the true gradient norm of two-dimensional Rosenbrock function in stochastic PC-oracle setting. We randomly generate initial points from normal distribution with mean 0 and variance 2, and conduct 30-trials. We can see that in practice, our proposal, BlockCD algorithm, can achieve the first-order necessary condition about local minimum even when the objective function is non-convex.

Dueling Bandit Gradient Descent : The dueling bandit gradient descent (DBGD) is a simple online optimization algorithm [60]. In each round, the DBGD firstly sample an unit vector u_t uniformly. Then, a duel (or a pairwise comparison) between current solution x_t and a feasible solution $x'_t = x_t + \delta u_t$ is performed. If x_t wins then the solution is not updated, otherwise the solution is updated along the direction u_t by step size γ . In [60], a duel is performed stochastic

way. Specifically, \mathbf{x}_t wins \mathbf{x}'_t by a probability

$$\Pr(\mathbf{x}_t > \mathbf{x}'_t) = \ell(f(\mathbf{x}_t) - f(\mathbf{x}'_t))$$

where f is the objective function and $\ell(\cdot)$ is a link function, e.g. logistic sigmoid function. The largest difference between our proposal and the DBGD is the guarantee of convergence. As we showed in theoretical section, our proposal is guaranteed to averagely converge to the optimal objective function value. On the other hand, the DBGD is only guaranteed sublinear regret which is measured by probability defined above. The algorithmic real performance greatly reflects this difference (see Figure 6). The experiments are done with test functions described in [60]. The first one is quadratic function $v_1(\mathbf{x}) = \mathbf{x}^\top \mathbf{x}$ and the second one is the hyperbolic cosine $v_4(\mathbf{x}) = \sum_{i=1}^d \{\exp(x_i) + \exp(-x_i)\}$ (The notation v_1 and v_4 are same as [60]). In this experiment, the performance of two algorithms are evaluated by the number of queries requested in the algorithm. To perform a fair comparison, (i) we use the stochastic PC oracle (99) with repeated querying subroutine (Algorithm 8) in both algorithms, (ii) we set η (tolerance parameter of line search in the BlockCD algorithm) and γ (step size parameter in the DBGD algorithm) in the same value. Figure 6 plotted averages and variances of ten times of trials. We can see that the objective function values in the DBGD algorithm can not efficiently converge to the optimal value. This is because the probability that the correct PC information returns comes near 1/2 and hence the update which achieves decrease of function value becomes hard to occur. On the other hand, our proposal is shown to converge faster than the DBGD.

10.2 Numerical Experiments of Parallel Computation

In BlockCD[n, m] with $m \geq 2$, one can execute the line search algorithm to each axis separately. Hence, the parallel computation is directly available to find each component of the search direction. Below, we investigate the computational efficiency of the parallel implementation of our method. The numerical experiments were conducted on AMD Opteron Processor 6176 (2.3GHz) with 48 cores (4 processors each of which has 12 cores), running Cent OS Linux release 6.4. In numerical experiments, the R language [46] with snow library for parallel statistical computing was used.

In numerical experiments using the PC oracle, BlockCD[n, m] with $m \geq 2$ and its parallel implementations were compared with the Nelder-Mead algorithm, DBGD, and original PC algorithm, i.e., BlockCD[$n, 1$]. In each iteration of BlockCD[n, m] with $m \geq 2$, $m + 1$ runs of the line search were required. In the parallel implementation, line-search tasks except the search step in Algorithm 6 were almost equally assigned to each processor. In the following, the parallel implementation of BlockCD[n, m] is referred to as parallel-BlockCD[n, m]. When c processors are used in parallel-BlockCD[n, m] such that n and m are much greater than c , the parallel computation will be approximately $(m + 1)/(m/c + 1) \approx c$ times more efficient than the serial processing. Practically, however, the communication overhead among processors may cancel the effect of the parallel computation, especially in small-scale problems.

We examined optimization methods on two n -dimensional optimization problems: the quadratic function $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x}$, and Rosenbrock function, $f(\mathbf{x}) = \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$. The n by n matrix \mathbf{A} was defined as $\mathbf{A} = \mathbf{B}^\top \mathbf{B}$, and each element of the n by n matrix \mathbf{B} was randomly generated from the standard normal distribution. The quadratic function satisfies the assumptions in Theorem 6. The Rosenbrock function is not convex, and thus, the assumption is violated. We examine whether the proposed method efficiently works even when the theoretical

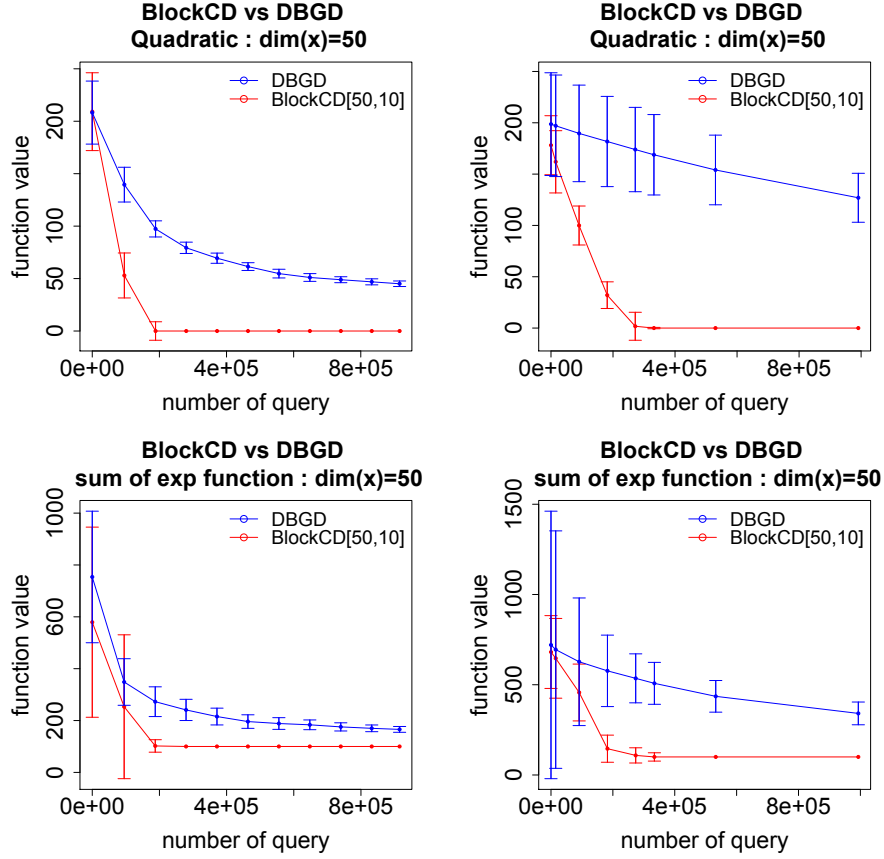


Figure 6: Comparison between the BlockCD and the DBGD with test functions described in [60]. Top two panels: 50-dimensional quadratic function (v_1 in § 5.1 of [60]). Bottom two panels: summation of 50-dimensional exponential functions (v_4 in § 5.1 of [60]). For each test function, left panel shows the results at $\eta = 10^{-1} = \gamma$, where η is tolerance parameter of line search in the BlockCD and γ is step size parameter in the DBGD. Similarly, right panel shows the results at $\eta = 10^{-2} = \gamma$. Although the DBGD is guaranteed sublinear regret, it can not efficiently decrease the objective function value compared to the BlockCD.

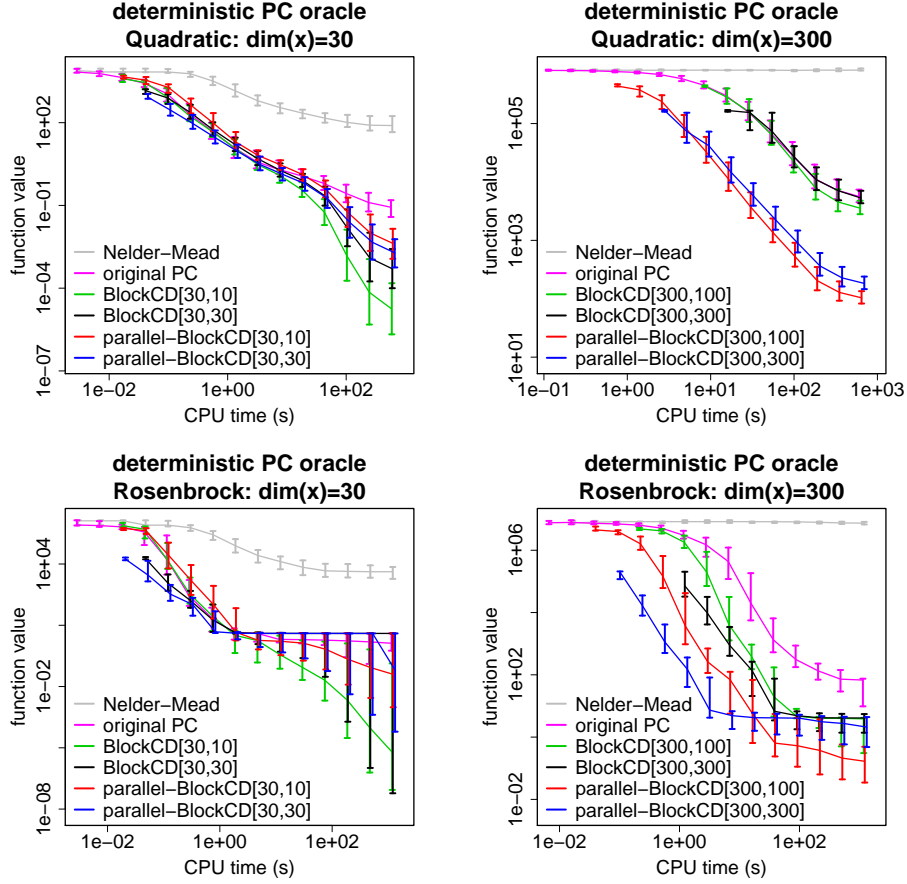


Figure 7: Deterministic PC oracle is used in optimization algorithms. Top panels: results in optimization of quadratic function. Bottom panels: results in optimization of Rosenbrock function. Nelder-Mead algorithm, original PC algorithm, BlockCD[n, m] with $m = n$ and $m = n/3$, and parallel-BlockCD[n, m] with $m = n$ and $m = n/3$, are compared for $n = 30$ and $n = 300$. Median of function values is presented to the CPU time (s). The vertical bar shows the percentile 30% to 70%.

assumptions are not necessarily satisfied. In each objective function, the dimension was set to $n = 30$ or 300 . In all problems, the minimum value is zero. For each algorithm, the optimization was repeated 10 times using randomly chosen initial points generated from $N(0, 3^2)$. According to [17], we examined some tuning parameters for the Nelder-Mead algorithm, and we verified that initial points did not significantly affect the numerical results in the present experiments. Hence, the standard parameter setting of the Nelder-Mead algorithm recommended in [17] was used throughout the present experiments.

Numerical results using the deterministic PC oracle are presented in Figure 7. For each algorithm, the median of function values in optimization process is depicted as the solid line with 30 and 70 percentiles for each CPU time. The Nelder-Mead algorithm did not efficiently work even for 30-dimensional quadratic functions. The performance of the Nelder-Mead algorithm tends to be easily degraded in high dimensional problems, as reported by several authors; see [17] and references therein. The present numerical results agree with past results. The original PC algorithm and serially executed BlockCD[n, m] were comparable. As shown in

(107), the upper bound of the query complexity is independent of m , when the deterministic PC oracle is used. As for the efficiency of the parallel computation, the parallel-BlockCD $[n, m]$ outperformed the competitors in 300 dimensional problems. In our experiments, the parallel implementation was about 15 times more efficient than the serial implementation in CPU time. For large-scale problems, the communication overhead can be canceled by the efficiency of the parallel computation. In our approach, the parallel computation is easily conducted without losing the convergence property as proved in Theorem 6. Computational efficiency of the parallel computation of our method was verified in moderate-scale optimization problems.

Also, we conducted optimization using the stochastic PC oracle. The DBGD was employed instead of the Nelder-Mead algorithm, since the Nelder-Mead algorithm even with the deterministic PC oracle did not work efficiently. The parameter in the stochastic PC oracle was set to $\kappa = 2$, $\delta_0 = 0.3$ and $\mu = 0.01$. Hence, the probability that the oracle returns the correct sign depends on the difference of function values. According to Lemma 2, queries were repeated at each point so that the probability of receiving the correct sign exceeded $1 - \delta$, where $\delta = 0.1$. The step-size parameter η in the BlockCD and DBGD was set to $\eta = 0.01$. Numerical results are presented in Figure 8. We dropped results of BlockCD $[n, n]$, since BlockCD $[n, m]$ with a large m was extremely inefficient. The convergence rate of the DBGD and original PC algorithm was relatively slow, though the computational cost of each iteration was not high. When the stochastic PC oracle was used, the parallel implementation of BlockCD $[n, m]$ achieved a faster convergence rate compared to the other algorithms in CPU time.

11 Summary of Part II

In this part, we proposed a block coordinate descent algorithm for unconstrained optimization problems using the pairwise comparison of function values. Our algorithm consists of two steps: the direction estimate step and search step. The direction estimate step can easily be parallelized. Hence, our algorithm is effectively applicable to large-scale optimization problems. Theoretically, we obtained an upper bound of the convergence rate and the query complexity, when the deterministic and stochastic pairwise comparison oracles were used. Our upper bound achieves the optimal rate with respect to the query complexity when oracle parameter $\kappa > 1$. On the other hand, in practice, our algorithm is simple and easy to implement. In addition, numerical experiments show that the parallel implementation of our algorithm outperformed the other methods using pairwise comparison. An extension of our algorithm to constrained optimization problems is an important future work. Other interesting research directions include pursuing the relation between pairwise comparison oracle and other kind of oracles such as gradient-sign oracle [47].

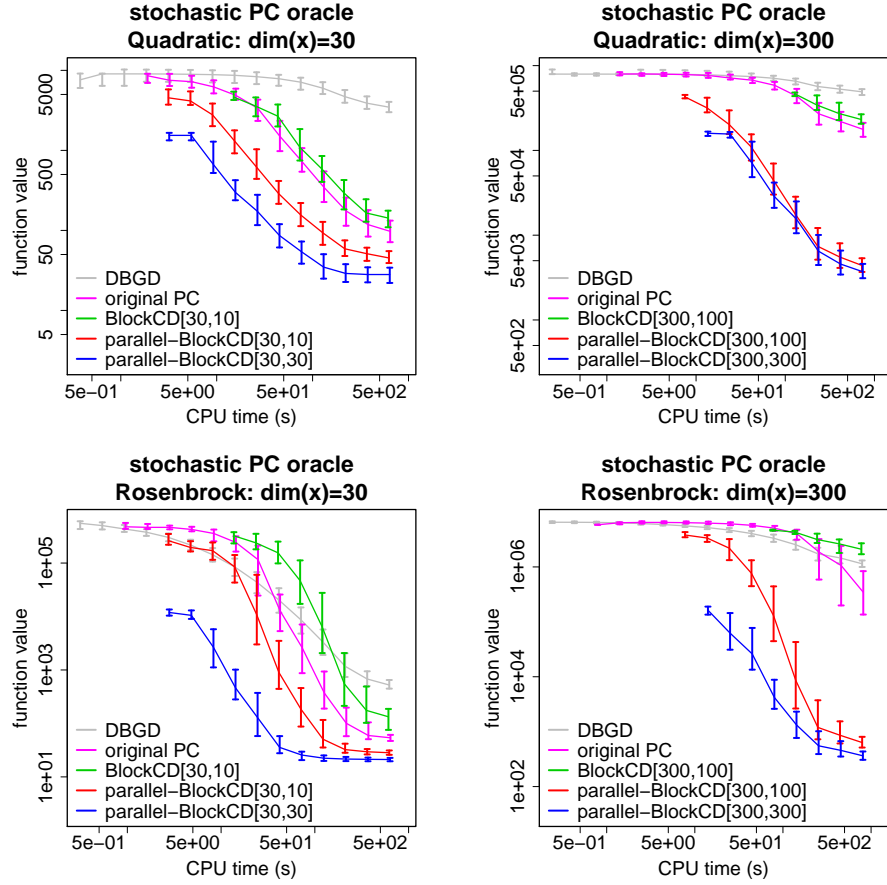


Figure 8: Stochastic PC oracle is used in optimization algorithms. Top panels: results in optimization of quadratic function. Bottom panels: results in optimization of Rosenbrock function. DBGD method, original PC algorithm, BlockCD[n, m] with $m = n/3$ and parallel-BlockCD[n, m] with $m = n$ and $m = n/3$ are compared for $n = 30$ and $n = 300$. Median of function values is shown to the CPU time (s). The vertical bar stands for the percentile 30% to 70%.

12 Concluding Remarks

In this dissertation, we presented a framework for the formulation of uncertain decision making problem as multi-objective optimization. We discussed the design of algorithms and its theoretical analysis. Although the algorithm we proposed in Part I is based on the conventional single-objective optimization algorithm, we proved that the obtained optimal solution satisfies Pareto optimality under certain conditions. The practical demonstrations of interaction with decision makers are shown through numerical experiments. In Part 2, we proposed an algorithm, BlockCD, to optimize unknown objective function based on noisy pairwise comparison oracle. We derived the convergence rate of the BlockCD algorithm and showed it achieves mini-max optimal rate under certain parameter settings. We also showed the practical performance of the BlockCD algorithm by comparison with other state-of-the-art methods through the numerical experiments.

We started from cases that the decision makers can express their preferences explicitly, and considered more general problem setting, that is optimization based on implicit (or noisy) preferences. However, the problem discussed in this dissertation is not yet completely natural setting. For example, in Part II, we proved the convergence of the BlockCD algorithm under the assumption of convexity and smoothness of the objective function. But in real decision-making problems, we can not expect that the unknown utility function of the decision makers have these properties. Although the possibility of application to non-convex optimization problems is suggested through optimization of the Rosenbrock function, it may be difficult to guarantee the convergence or to derive the convergence rate, for general non-convex functions.

An approach to non-convex utility maximization is Bayesian optimization [7] that we introduced in Section 1. Bayesian optimization is a model-based approach that models an arbitrary objective function by Gaussian process. Then deriving an optimal solution with a small number of searches by sampling points with high uncertainty like active learning. On the other hand, there is a non model-based approach which is called graduated optimization [22] (or continuation). The graduated optimization is a method that introduces a convexification parameter for the objective function and aims to escape from the local minima like simulated annealing method.

Hence, as a future work, we focus on graduated optimization and aim at designing a pairwise comparison-based optimization algorithm. The reason we focus on graduated optimization is that it is a data-driven method that does not assume models. Assuming a model for the utility function, it is necessary to evaluate the deviation from the true preference structure in some way, but it is a difficult task. In decision making problem, it perhaps practical to execute optimization based on observation data (in this case, pairwise comparison between two points). Furthermore, we are planning to proceed with applying to actual decision making problems, including the method proposed in this dissertation.

References

- [1] G Anandalingam. A mathematical programming model of decentralized multi-level systems. *Journal of the Operational Research Society*, 39(11):1021–1033, 1988.
- [2] Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.
- [3] Charles Audet, John E Dennis Jr, and Sèbastien Le Digabel. Parallel space decomposition of the mesh adaptive direct search algorithm. *SIAM Journal on Optimization*, 19(3):1150–1170, 2008.
- [4] Charles Audet, Andrea Ianni, Sèbastien Le Digabel, and Christophe Tribes. Reducing the number of function evaluations in mesh adaptive direct search algorithms. *SIAM Journal on Optimization*, 24(2):621–642, 2014.
- [5] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
- [6] Stephen Poythress Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [7] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
- [8] Abraham Charnes and William W Cooper. Programming with linear fractional functionals. *Naval Research logistics quarterly*, 9(3-4):181–186, 1962.
- [9] A Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*, volume 8. SIAM, 2009.
- [10] Andrew R Conn and Sèbastien Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. *Optimization Methods and Software*, 28(1):139–158, 2013.
- [11] Bruno Contini. A stochastic approach to goal programming. *Operations Research*, 16(3):576–586, 1968.
- [12] AL Custòdio, JE Dennis, and Lu’is Nunes Vicente. Using simplex gradients of nonsmooth functions in direct search methods. *IMA Journal of Numerical Analysis*, 28(4):770–784, 2008.
- [13] Didier J Dubois. *Fuzzy sets and systems: theory and applications*, volume 144. Academic press, 1980.
- [14] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’05, pages 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

- [15] Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization: Formulation and discussion and generalization. In *ICGA*, volume 93, pages 416–423. Citeseer, 1993.
- [16] M. C. Fu. Gradient estimation. In S. G. Henderson and B. L. Nelson, editors, *Handbooks in Operations Research and Management Science: Simulation*, chapter 19. Elsevier Amsterdam, 2006.
- [17] Fuchang Gao and Lixing Han. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.*, 51(1):259–277, 2012.
- [18] Arthur M Geoffrion, James S Dyer, and A Feinberg. An interactive approach for multicriterion optimization, with an application to the operation of an academic department. *Management science*, 19(4-part-1):357–368, 1972.
- [19] John J Glen. Mathematical models in farm planning: A survey. *Operations Research*, 35(5):641–666, 1987.
- [20] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addison wesley*, 1989:102, 1989.
- [21] Kiyotada Hayashi. Multicriteria analysis for agricultural resource management: a critical survey and future perspectives. *European Journal of Operational Research*, 122(2):486–500, 2000.
- [22] Elad Hazan, Kfir Y Levy, and Shai Shalev-Shwartz. On graduated optimization for stochastic non-convex problems. *arXiv preprint arXiv:1503.03712*, 2015.
- [23] Takeshi Itoh, Hiroaki Ishii, and Teruaki Nanseki. A model of crop planning under uncertainty in agricultural management. *International Journal of Production Economics*, 81:555–558, 2003.
- [24] K. G. Jamieson, R. D. Nowak, and B. Recht. Query complexity of derivative-free optimization. In *NIPS*, pages 2681–2689, 2012.
- [25] H Katagiri, H Ishii, and T Itoh. Fuzzy random linear programming problem. In *Proc. Second European Workshop on Fuzzy Decision Analysis and Neural Networks for Management, Planning and Optimization*, pages 107–115, 1997.
- [26] H Katagiri, M Sakawa, K Kato, and S Ohsaki. An interactive fuzzy satisficing method for fuzzy random multiobjective linear programming problems through the fractile optimization model using possibility and necessity measures. In *Proceedings of the 9th Asia Pacific Management Conference*, pages 795–802, 2003.
- [27] Hideki Katagiri, Takashi Hasuike, and Hiroaki Ishii. A random fuzzy programming models based on possibilistic programming. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 1788–1793. IEEE, 2008.
- [28] Hideki Katagiri, Kosuke Kato, and Takashi Hasuike. A random fuzzy minimum spanning tree problem through a possibility-based value at risk model. *Expert Systems with Applications*, 39(12):10639–10646, 2012.

- [29] Hideki Katagiri and Masatoshi Sakawa. Interactive multiobjective fuzzy random programming through the level set-based probability model. *Information Sciences*, 181(9):1641–1650, 2011.
- [30] Hideki Katagiri, Masatoshi Sakawa, Kosuke Kato, and Ichiro Nishizaki. Interactive multi-objective fuzzy random linear programming: Maximization of possibility and probability. *European journal of operational research*, 188(2):530–539, 2008.
- [31] Hideki Katagiri, Masatoshi Sakawa, Kosuke Kato, and Syuuji Ohsaki. An interactive fuzzy satisficing method based on the fractile optimization model using possibility and necessity measures for a fuzzy random multiobjective linear programming problem. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 88(5):20–28, 2005.
- [32] Hideki Katagiri, Masatoshi Sakawa, and Takeshi Matsui. Interactive multiobjective random fuzzy programming problems through the possibility-based fractile model. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 3144–3148. IEEE, 2011.
- [33] Matti Kääriäinen. Active learning in the non-realizable case. In *Algorithmic Learning Theory*, pages 63–77. Springer, 2006.
- [34] Huibert Kwakernaak. Fuzzy random variables—i. definitions and theorems. *Information Sciences*, 15(1):1–29, 1978.
- [35] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147, 1998.
- [36] Young-Jou Lai. Hierarchical optimization: a satisfactory solution. *Fuzzy Sets and Systems*, 77(3):321–335, 1996.
- [37] E Stanley Lee and Hsu-Shih Shih. *Fuzzy and multi-level decision making: An interactive computational approach*. Springer Science & Business Media, 2012.
- [38] Baoding Liu. Random fuzzy dependent-chance programming and its hybrid intelligent algorithm. *Information Sciences*, 141(3):259–271, 2002.
- [39] Baoding Liu. Uncertainty theory. In *Uncertainty Theory*, pages 205–234. Springer, 2007.
- [40] D. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.
- [41] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [42] Kota Matsui, Wataru Kumagai, and Takafumi Kanamori. Parallel distributed block coordinate descent methods based on pairwise comparison oracle. *Journal of Global Optimization*, doi:10.1007/s10898-016-0465-x, 2016.
- [43] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

- [44] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [45] Madan L Puri and Dan A Ralescu. Fuzzy random variables. *Journal of mathematical analysis and applications*, 114(2):409–422, 1986.
- [46] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [47] Aaditya Ramdas and Aarti Singh. Algorithmic connections between active learning and stochastic convex optimization. In *Algorithmic Learning Theory*, pages 339–353. Springer, 2013.
- [48] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.
- [49] Luis Miguel Rios and Nikolaos V Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.
- [50] Masatoshi Sakawa. *Fuzzy sets and interactive multiobjective optimization*. Springer Science & Business Media, 2013.
- [51] Masatoshi Sakawa, Ichiro Nishizaki, and Hideki Katagiri. *Fuzzy stochastic multiobjective programming*, volume 159. Springer Science & Business Media, 2011.
- [52] J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms*, pages 93–100. L. Erlbaum Associates Inc., 1985.
- [53] Hsu-Shih Shih, Young-Jou Lai, and E Stanley Lee. Fuzzy approach for multi-level programming problems. *Computers & Operations Research*, 23(1):73–91, 1996.
- [54] Ue-Pyng Wen and Shuh-Tzy Hsu. Linear bi-level programming problems—a review. *Journal of the Operational Research Society*, 42(2):125–133, 1991.
- [55] Hitoshi Yano and Kota Matsui. Two fuzzy approaches for multiobjective stochastic programming and multiobjective fuzzy random programming through a probability maximization model. *IAENG International Journal of Computer Science*, 38(3), 2011.
- [56] Hitoshi Yano and Kota Matsui. Hierarchical multiobjective stochastic linear programming problems based on the fuzzy decision. *Iaeng Transactions on Engineering Technologies*, 7:1–14, 2012.
- [57] Hitoshi Yano and Kota Matsui. Hierarchical multiobjective fuzzy random linear programming problems. *Procedia Computer Science*, 22:162–171, 2013.
- [58] Hitoshi Yano and Kota Matsui. Random fuzzy multiobjective linear programming with variance covariance matrices. In *Transactions on Engineering Technologies*, pages 391–405. Springer, 2014.

- [59] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012.
- [60] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.
- [61] H-J Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy sets and systems*, 1(1):45–55, 1978.
- [62] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the K-armed dueling bandit problem. In *ICML 2014: Proceedings of the Thirty-First International Conference on Machine Learning*, pages 10–18, June 2014.