

Article

Using Genetic Algorithms and Core Values of Cooperative Games to Solve Fuzzy Multiobjective Optimization Problems

Hsien-Chung Wu 

Department of Mathematics, National Kaohsiung Normal University, Kaohsiung 82444, Taiwan; hsien.chung.wu@gmail.com or hcwu@mail.nknu.edu.tw

Abstract: A new methodology for solving the fuzzy multiobjective optimization problems is proposed in this paper by considering the fusion of cooperative game theory and genetic algorithm. The original fuzzy multiobjective optimization problem needs to be transformed into a scalar optimization problem, which is a conventional optimization problem. Usually, the assignments of suitable coefficients to the corresponding scalar optimization problem are subjectively determined by the decision makers. However, these assignments may cause some biases by their subjectivity. Therefore, this paper proposes a mechanical procedure to avoid this subjective biases. We are going to formulate a cooperative game using the α -level functions of the multiple fuzzy objective functions. Under this setting, the suitable coefficients can be determined mechanically by involving the core values of the cooperative game, which is formulated using the multiple fuzzy objective functions. We shall prove that the optimal solutions of the transformed scalar optimization problem are indeed the nondominated solutions of fuzzy multiobjective optimization problem. Since the core-nondominated solutions will depend on the coefficients that are determined by the core values of cooperative game, there will be a lot of core-nondominated solutions that will also depend on the corresponding coefficients. In order to obtain the best core-nondominated solution, we shall invoke the genetic algorithms by evolving the coefficients.

Keywords: fuzzy sets; cooperative games; genetic algorithms; fuzzy multiobjective optimization; scalar optimization

MSC: 90C29; 03E71



Citation: Wu, H.-C. Using Genetic Algorithms and Core Values of Cooperative Games to Solve Fuzzy Multiobjective Optimization Problems. *Axioms* **2024**, *13*, 298. <https://doi.org/10.3390/axioms13050298>

Academic Editors: Manuel Arana-Jimenez, Amit K. Shukla, Darjan Karabašević and Juan Vicente Riera

Received: 15 March 2024
Revised: 15 April 2024
Accepted: 27 April 2024
Published: 29 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bellman and Zadeh [1] initiate the research topic of fuzzy optimization. The main idea of their approach is to combine the fuzzy goals and fuzzy decision space using the aggregation operators. Tanaka et al. [2] and Zimmermann [3,4] proposed the concept of aspiration level to study the linear programming problems and linear multiobjective programming problems. Herrera et al. [5] also used the concept of aspiration level and triangular norm (t-norm) to aggregate the fuzzy goals and fuzzy constraints. On the other hand, by mimicking the probability distribution in stochastic optimization, Buckley [6], Julien [7] and Luhandjula et al. [8] considered the concept of possibility distribution to study the fuzzy optimization problems. Inuiguchi [9] also used the so-called possibility and necessity measures to study the modality constrained optimization problems. Those approaches were mainly to fuzzify the crisp constraints and crisp objective functions. More precisely, given the following crisp constraints

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \text{ for } i = 1, \dots, m,$$

where a_{ij} and b_i are real numbers, we can fuzzify the crisp constraints as follows

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \lesssim b_i \text{ for } i = 1, \dots, m,$$

where the membership functions are assigned using the aspiration level to describe the degree of violation for the original crisp constraints. Another method is to fuzzify the real numbers a_{ij} and b_i using the possibility distributions.

There is another interesting approach without considering the fuzzification. This kind of approach mainly takes care of the coefficients of optimization problems. Usually, those coefficients are assumed to be fuzzy intervals (fuzzy numbers). For instance, the fuzzy linear programming problem (FLP) is formulated as follows

$$\begin{aligned}
 \text{(FLP)} \quad & \text{maximize} \quad \left(\tilde{A}^{(1)} \otimes \tilde{I}_{\{x_1\}} \right) \oplus \left(\tilde{A}^{(2)} \otimes \tilde{I}_{\{x_2\}} \right) \oplus \cdots \oplus \left(\tilde{A}^{(n)} \otimes \tilde{I}_{\{x_n\}} \right) \\
 & \text{subject to} \quad b_{j1}x_1 + b_{j2}x_2 + \cdots + b_{jn}x_n \leq \gamma_j \text{ for } j = 1, \dots, m; \\
 & \quad \quad \quad x_i \geq 0 \text{ for } i = 1, \dots, n,
 \end{aligned}$$

where the addition \oplus and multiplication \otimes of fuzzy intervals are involved and appeared as the coefficients. Owing to the unexpected fluctuation and turbulence in a uncertain environment, we sometimes cannot precisely measure the desired data. In this case, the corresponding optimization problems cannot be precisely formulated, since the data appear to be uncertain. Therefore, the reasonable way is to consider the fuzzy intervals or fuzzy numbers to be the coefficients of these optimization problems. In other words, the fuzzy optimization problems can be formulated such that the coefficients are assumed to be fuzzy intervals or fuzzy numbers. This kind of approach seems to become a mainstream of the topic of fuzzy optimization.

Regarding the fuzzy coefficients and using the Hukuhara derivative, Wu [10–13] studied the duality theorems and optimality conditions for fuzzy optimization problems. The so-called generalized Hukuhara derivative was adopted by Chalco-Cano et al. [14] to extend the Karush–Kuhn–Tucker optimality conditions for fuzzy optimization problems with fuzzy coefficients. The concept of generalized convexity was also considered by Li et al. [15] to study the optimality conditions of fuzzy optimization problems. On the other hand, regarding the issue of numerical methods, Chalco-Cano et al. [16] and Pirezada and Pathak [17] proposed the so-called Newton method to solve the fuzzy optimization problems.

The multiobjective programming problems with fuzzy objective functions was studied by Luhandjula [18] in which the approach of defuzzification was adopted. An interactive method was proposed by Yano [19] to solve the multiobjective linear programming problems in which fuzzy coefficients were considered in the objective functions. Regarding the applications of fuzzy multiobjective optimization problem, Ebeunuwa et al. [20] proposed a multi-objective design optimization approach for the optimal analysis of buried pipe. Charles et al. [21] studied a probabilistic fuzzy goal multi-objective supply chain network problem. Roy et al. [22] studied a multiobjective multi-product solid transportation problem in which the system parameters are assumed to be rough fuzzy variables.

In this paper, we study the fuzzy multiobjective linear programming problem (FMLP) as follows

$$\begin{aligned}
 \text{(FMLP)} \quad & \text{maximize} \quad \left(\tilde{H}^{(1)}(\tilde{\mathbf{A}}, \mathbf{x}), \dots, \tilde{H}^{(u)}(\tilde{\mathbf{A}}, \mathbf{x}) \right) \\
 & \text{subject to} \quad b_{j1}x_1 + b_{j2}x_2 + \cdots + b_{jn}x_n \leq \gamma_j \text{ for } j = 1, \dots, m; \\
 & \quad \quad \quad x_i \geq 0 \text{ for } i = 1, \dots, n,
 \end{aligned}$$

where the objective functions are fuzzy linear functions given by

$$\begin{aligned}
 \tilde{H}^{(1)}(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}) &= \left(\tilde{A}^{(11)} \otimes \tilde{I}_{\{x_1\}} \right) \oplus \left(\tilde{A}^{(12)} \otimes \tilde{I}_{\{x_2\}} \right) \oplus \cdots \oplus \left(\tilde{A}^{(1n)} \otimes \tilde{I}_{\{x_n\}} \right) \\
 \tilde{H}^{(2)}(\tilde{\mathbf{A}}^{(2)}, \mathbf{x}) &= \left(\tilde{A}^{(21)} \otimes \tilde{I}_{\{x_1\}} \right) \oplus \left(\tilde{A}^{(22)} \otimes \tilde{I}_{\{x_2\}} \right) \oplus \cdots \oplus \left(\tilde{A}^{(2n)} \otimes \tilde{I}_{\{x_n\}} \right) \\
 &\quad \vdots \\
 \tilde{H}^{(u)}(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}) &= \left(\tilde{A}^{(u1)} \otimes \tilde{I}_{\{x_1\}} \right) \oplus \left(\tilde{A}^{(u2)} \otimes \tilde{I}_{\{x_2\}} \right) \oplus \cdots \oplus \left(\tilde{A}^{(un)} \otimes \tilde{I}_{\{x_n\}} \right).
 \end{aligned}$$

In order to introduce the concepts of nondominated solution, we need to propose an ordering relation among the set of all fuzzy intervals or fuzzy numbers. Using this ordering relation, the concept of nondominated solution of fuzzy multiobjective optimization problems can be defined. One of the main approaches is to transform the original fuzzy multiobjective optimization problem into a scalar optimization problem by considering the suitable weights, which is a conventional optimization problem such that the coefficients are real numbers. Under these settings, the important issue is to show that the optimal solution of the transformed scalar optimization problem is also a nondominated solution of the original fuzzy multiobjective optimization problem. This also says that it is sufficient to just solve the transformed scalar optimization problem. As we can see later, the set of all nondominated solutions can be a large set, which depends on the weights that are determined in the step of transformation. In order to find the best nondominated solution, we are going to design a genetic algorithm by providing a suitable fitness function.

There are many ways to formulate the scalar optimization problem. The issue is to assign the suitable weights to the fuzzy objective functions. Usually, the weights are determined by the decision maker using their subjectivity. In order to avoid the possible bias caused by their subjectivity, this paper considers a cooperative game, which is formulated by using the fuzzy objective functions. The important assignment of weights are mechanically determined according to the core values of this formulated cooperative game. This kind of mechanical assignment can rule out the bias caused by the intuition that are believed by the decision-makers for determining the weights.

Game theory mainly concerns the behavior of players like cooperation or non-cooperation such that the decisions determined by the players may affect each other. The pioneering work was initiated by von Neumann and Morgenstern [23]. Nash [24] proposed the concept of a two-person cooperative game in which the concept of Nash equilibrium was proposed. Another solution concept called monotonic solutions was proposed by Young [25]. The monographs by Barron [26], Branzei et al. [27], Curiel [28], González-Díaz et al. [29] and Owen [30] also provide detailed concepts of game theory. On the other hand, Yu and Zhang [31] used the generalized triangular fuzzy number to study a cooperative game with fuzzy payoffs. Three solution concepts called fuzzy cores were defined using the fuzzy max order.

Jing et al. [32] studied a bi-objective optimization problem, where the multi-benefit allocation constraints are modeled. The approach by Lokeshgupta and Sivasubramani [33] treated the two objective functions as a cooperative game with two players. Alternatively, the approach by Lee [34] treated the two objective functions as a non-cooperative game with two players and tried to obtain the Nash equilibrium. Meng and Xie [35] formulated a competitive-cooperative game method to obtain the optimal preference solutions. A three-objective optimization problem was studied by Li et al. [36] in which a three-players game was formulated. The approach by Yu et al. [37] and Zhang et al. [38] also formulated a three-players game. Zhang et al. [38] considered the sub-game perfect Nash equilibrium, and Yu et al. [37] incorporated the genetic algorithm to obtain the solutions. A four-objective optimization problem by Chai et al. [39] and a bi-objective optimization problem by Cao et al. [40] were solved by using the genetic algorithm in which the non-cooperative game theory was adopted.

Solving the multiobjective optimization problems via genetic algorithms has attracted attention for a long time. We may refer to the monographs by Deb [41], Osyczka [42], and Tan et al. [43] for more details on this topic. Also, the topic of fuzzy the multiobjective optimization problem can refer to the monograph by Sakawa [44]. On the other hand, Tiwari et al. [45] studied a nonlinear optimization problem in which a genetic algorithm was proposed to solve the problem.

In Section 2, a multiobjective optimization problem with fuzzy coefficients is introduced, and its nondominated solutions are also defined. In Section 3, the concept of core values in cooperative games is introduced. In Section 4, the multiple objective functions are formulated as a cooperative game. In Section 5, the suitable coefficients of the

scalar optimization problem are determined by using the core values of the formulated cooperative games. The different settings of coefficients can generate the different core-nondominated solutions. Therefore, in Section 6, an genetic algorithm is designed to find the best core-nondominated solution by providing a suitable fitness function. Finally, a practical numerical example is provided in Section 7 to illustrate the possible usage of the methodology proposed in this paper.

2. Formulation

The fuzzy set \tilde{A} in \mathbb{R} is defined by a membership function $\zeta_{\tilde{A}} : \mathbb{R} \rightarrow [0, 1]$. The α -level set of \tilde{A} is denoted and defined by

$$\tilde{A}_\alpha = \{x \in \mathbb{R} : \zeta_{\tilde{A}}(x) \geq \alpha\}$$

for all $\alpha \in (0, 1]$. According to the usual topology of \mathbb{R} , the 0-level set \tilde{A}_0 is defined to be the closure of the support

$$\{x \in \mathbb{R} : \zeta_{\tilde{A}}(x) > 0\}.$$

In other words, the 0-level set \tilde{A}_0 is defined by

$$\tilde{A}_0 = \text{cl}(\{x \in \mathbb{R} : \zeta_{\tilde{A}}(x) > 0\}).$$

Then, we have $\tilde{A}_\alpha \subseteq \tilde{A}_\beta$ for $\alpha, \beta \in [0, 1]$ with $\alpha > \beta$.

Given a subset A of \mathbb{R} , we can treat it as a fuzzy set $\tilde{1}_A$ in \mathbb{R} with membership function defined by

$$\zeta_{\tilde{1}_A}(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A. \end{cases}$$

In particular, if A is a singleton $\{a\}$, then we write $\tilde{1}_{\{a\}}$. In other words, each real number $a \in \mathbb{R}$ can be identified with the membership function $\tilde{1}_{\{a\}}$.

We say that \tilde{A} is a fuzzy interval when it is a fuzzy set in \mathbb{R} satisfying the following conditions.

- \tilde{A} is normal; that is, $\zeta_{\tilde{A}}(x) = 1$ for some $x \in \mathbb{R}$.
- The membership function $\zeta_{\tilde{A}}$ is quasi-concave and upper semicontinuous.
- The 0-level set \tilde{A}_0 is a closed and bounded subset of \mathbb{R} .

Since \tilde{A} is normal, it says that the α -level sets \tilde{A}_α are nonempty for all $\alpha \in [0, 1]$. We also see that the α -level sets \tilde{A}_α of fuzzy interval \tilde{A} are bounded closed intervals given by

$$\tilde{A}_\alpha = [\tilde{A}_\alpha^L, \tilde{A}_\alpha^U] \text{ for all } \alpha \in [0, 1].$$

Let $\mathcal{F}_{\mathbb{R}}$ denote the family of all fuzzy intervals in \mathbb{R} . The fuzzy optimization problem considers the fuzzy-valued functions $\tilde{H} : X \rightarrow \mathcal{F}_{\mathbb{R}}$, which is defined on a nonempty subset X of \mathbb{R}^n . It means that, for each $x \in X$, the function value $\tilde{H}(x)$ is a fuzzy interval. Now, given any fixed $\alpha \in [0, 1]$, we can generate two real-valued functions

$$\tilde{H}_\alpha^L : X \rightarrow \mathbb{R} \text{ and } \tilde{H}_\alpha^U : X \rightarrow \mathbb{R},$$

which are defined by

$$\tilde{H}_\alpha^L(x) = (\tilde{H}_\alpha(x))^L \text{ and } \tilde{H}_\alpha^U(x) = (\tilde{H}_\alpha(x))^U.$$

Given any two fuzzy \tilde{A} and \tilde{B} in \mathbb{R} , according to the extension principle, the addition and multiplication between \tilde{A} and \tilde{B} are defined by

$$\zeta_{\tilde{A} \oplus \tilde{B}}(z) = \sup_{\{(x,y):z=x+y\}} \min\{\zeta_{\tilde{A}}(x), \zeta_{\tilde{B}}(y)\}$$

and

$$\zeta_{\tilde{A} \otimes \tilde{B}}(z) = \sup_{\{(x,y):z=x*y\}} \min\{\zeta_{\tilde{A}}(x), \zeta_{\tilde{B}}(y)\}.$$

In this paper, the following fuzzy multiobjective optimization problem (FMOP)

$$\begin{aligned} \text{(FMOP)} \quad & \text{maximize} \quad \left(\tilde{H}^{(1)}(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}), \dots, \tilde{H}^{(u)}(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}) \right) \\ & \text{subject to} \quad \mathbf{x} \in X \end{aligned}$$

is considered, where X is a feasible region and is a subset of \mathbb{R}^n , $\tilde{\mathbf{A}}^{(j)}$ are vectors of fuzzy intervals and $\tilde{H}^{(j)}(\tilde{\mathbf{A}}, \mathbf{x})$ are the fuzzy objective functions of (FMOP) for $j = 1, \dots, u$. For example, we can take

$$\begin{aligned} \tilde{H}^{(1)}(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}) &= (\tilde{A}^{(11)} \otimes \tilde{I}_{\{x_1\}}) \oplus (\tilde{A}^{(12)} \otimes \tilde{I}_{\{x_2\}}) \oplus \dots \oplus (\tilde{A}^{(1n)} \otimes \tilde{I}_{\{x_n\}}) \\ \tilde{H}^{(2)}(\tilde{\mathbf{A}}^{(2)}, \mathbf{x}) &= (\tilde{A}^{(21)} \otimes \tilde{I}_{\{x_1\}}) \oplus (\tilde{A}^{(22)} \otimes \tilde{I}_{\{x_2\}}) \oplus \dots \oplus (\tilde{A}^{(2n)} \otimes \tilde{I}_{\{x_n\}}) \\ &\vdots \\ \tilde{H}^{(u)}(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}) &= (\tilde{A}^{(u1)} \otimes \tilde{I}_{\{x_1\}}) \oplus (\tilde{A}^{(u2)} \otimes \tilde{I}_{\{x_2\}}) \oplus \dots \oplus (\tilde{A}^{(un)} \otimes \tilde{I}_{\{x_n\}}), \end{aligned}$$

where the coefficients \tilde{A}_{rs} for $r = 1, \dots, u$ and $s = 1, \dots, n$ are taken to be the fuzzy intervals. In particular, the fuzzy multiobjective linear programming problem (FMLP) is formulated below

$$\begin{aligned} \text{(FMLP)} \quad & \text{maximize} \quad \left(\tilde{H}^{(1)}(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}), \dots, \tilde{H}^{(u)}(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}) \right) \\ & \text{subject to} \quad b_{j1}x_1 + b_{j2}x_2 + \dots + b_{jn}x_n \leq \gamma_j \text{ for } j = 1, \dots, m; \\ & \quad \quad \quad x_i \geq 0 \text{ for } i = 1, \dots, n, \end{aligned}$$

where the objective functions are fuzzy-valued functions and the constraint functions are real-valued functions. The meaning of nondominated solution of problem (FMOP) should be defined based on an ordering relation among the set of all fuzzy intervals, which is shown below.

Definition 1. Let \tilde{A} and \tilde{B} be two fuzzy intervals in \mathbb{R} .

- We define $\tilde{A} \preceq \tilde{B}$ when $\tilde{A}_\alpha^L \leq \tilde{B}_\alpha^L$ and $\tilde{A}_\alpha^U \leq \tilde{B}_\alpha^U$ for all $\alpha \in [0, 1]$.
- We define $\tilde{A} \prec \tilde{B}$ when $\tilde{A} \preceq \tilde{B}$ and there exists $\alpha^* \in [0, 1]$ satisfying

$$\tilde{A}_{\alpha^*}^L < \tilde{B}_{\alpha^*}^L \text{ for all } 0 \leq \alpha \leq \alpha^* \text{ or } \tilde{A}_{\alpha^*}^U < \tilde{B}_{\alpha^*}^U \text{ for all } \alpha^* \leq \alpha \leq 1.$$

We see that $\tilde{A} \prec \tilde{B}$ implies $\tilde{A} \preceq \tilde{B}$. The ordering relation “ \prec ” is transitive on $\mathcal{F}_{\mathbb{R}}$ in the sense of $\tilde{A} \prec \tilde{B}$ and $\tilde{B} \prec \tilde{C}$ implying $\tilde{A} \prec \tilde{C}$.

Definition 2. Given a feasible solution $\mathbf{x}^* \in X$, when there does not exist another feasible solution $\mathbf{x} \in X$ satisfying

$$\tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*) \preceq \tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) \text{ for all } j = 1, \dots, u$$

and

$$\tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \prec \tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}) \text{ for some } j^* \in \{1, \dots, u\},$$

this feasible solution \mathbf{x}^* is said to be a nondominated solution of fuzzy multiobjective optimization problem (FMOP).

For convenience, we write

$$\tilde{H}_\alpha^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) \equiv \left(\tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x})\right)_\alpha^L \text{ and } \tilde{H}_\alpha^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) \equiv \left(\tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x})\right)_\alpha^U$$

for $j = 1, \dots, u$ and $\alpha \in [0, 1]$. Let $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ be a partition of $[0, 1]$. For $j = 1, \dots, u$, we define the following functions

$$H_j(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) = w_{j1} \cdot \tilde{H}_{\alpha_1}^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) + \dots + w_{jm} \cdot \tilde{H}_{\alpha_m}^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) + w_{j,m+1} \cdot \tilde{H}_{\alpha_1}^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) + \dots + w_{j,2m} \cdot \tilde{H}_{\alpha_m}^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}) \tag{1}$$

where $w_{jk} \geq 0$ for all $j = 1, \dots, u$ and $k = 1, \dots, 2m$ satisfying

$$\sum_{j=1}^u \sum_{k=1}^{2m} w_{jk} = 1.$$

The following scalar optimization problem

$$\begin{aligned} \text{(SOP)} \quad & \text{maximize} \quad H_1(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}) + H_2(\tilde{\mathbf{A}}^{(2)}, \mathbf{x}) + \dots + H_u(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}) \\ & \text{subject to} \quad \mathbf{x} \in X \end{aligned}$$

is considered. The nondominated solutions of problem (FMOP) can be obtained by following the theorem presented below.

Theorem 1. For $w_{jk} > 0$ for all $j = 1, \dots, u$ and $k = 1, \dots, 2m$, If \mathbf{x}^* is an optimal solution of problem (SOP), then it is also a nondominated solution of problem (FMOP).

Proof. Assume that \mathbf{x}^* is not a nondominated solution of problem (FMOP). Then, we want to lead to a contradiction. In this case, there exists \mathbf{x}^\dagger satisfying

$$\tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*) \preceq \tilde{H}^{(j)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^\dagger) \text{ for all } j = 1, \dots, u$$

and

$$\tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \prec \tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ for some } j^* \in \{1, \dots, u\}.$$

For $j = 1, \dots, u$ and $\alpha \in [0, 1]$, by referring to (1), we have

$$\tilde{H}_\alpha^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*) \leq \tilde{H}_\alpha^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^\dagger) \text{ and } \tilde{H}_\alpha^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*) \leq \tilde{H}_\alpha^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^\dagger),$$

which imply

$$H_j(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*) \leq H_j(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^\dagger) \text{ for all } j = 1, \dots, u. \tag{2}$$

Since $\tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \prec \tilde{H}^{(j^*)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger)$, the following conditions are satisfied.

- For $\alpha \in [0, 1]$, we have

$$\tilde{H}_\alpha^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_\alpha^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ and } \tilde{H}_\alpha^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_\alpha^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger). \tag{3}$$

- There exists $\alpha^* \in [0, 1]$ satisfying

$$\tilde{H}_{\alpha^*}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha^*}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ for all } 0 \leq \alpha \leq \alpha^* \tag{4}$$

or

$$\tilde{H}_{\alpha^*}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha^*}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ for all } \alpha^* \leq \alpha \leq 1. \tag{5}$$

We want to show that the following conditions are satisfied.

- For $i = 1, \dots, m$, we have

$$\tilde{H}_{\alpha_i}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha_i}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ and } \tilde{H}_{\alpha_i}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha_i}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger).$$

- There exists $\alpha_r \in \Gamma$ satisfying

$$\tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ or } \tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger).$$

Using (3), it follows

$$\tilde{H}_{\alpha_i}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha_i}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ and } \tilde{H}_{\alpha_i}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha_i}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger)$$

for all $i = 1, \dots, m$. Since Γ is a partition of $[0, 1]$, we consider the following cases.

- Using (4), there exists $\alpha_r \in \Gamma$ satisfying $\alpha_r \leq \alpha^*$ and

$$\tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha^*}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger)$$

- Using (5), there exists $\alpha_r \in \Gamma$ satisfying $\alpha_r \geq \alpha^*$ and

$$\tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \leq \tilde{H}_{\alpha^*}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger).$$

Therefore, we conclude that there exists $\alpha_r \in \Gamma$ satisfying

$$\tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ or } \tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < \tilde{H}_{\alpha_r}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger).$$

Since each $w_{jk} > 0$, we have

$$\begin{aligned} & w_{j^*1} \cdot \tilde{H}_{\alpha_1}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) + \dots + w_{j^*m} \cdot \tilde{H}_{\alpha_m}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \\ & + w_{j^*,m+1} \cdot \tilde{H}_{\alpha_1}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) + \dots + w_{j^*,2m} \cdot \tilde{H}_{\alpha_m}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) \\ & < w_{j^*1} \cdot \tilde{H}_{\alpha_1}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) + \dots + w_{j^*m} \cdot \tilde{H}_{\alpha_m}^{(j^*L)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \\ & + w_{j^*,m+1} \cdot \tilde{H}_{\alpha_1}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) + \dots + w_{j^*,2m} \cdot \tilde{H}_{\alpha_m}^{(j^*U)}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger), \end{aligned}$$

which also says

$$H_{j^*}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^*) < H_{j^*}(\tilde{\mathbf{A}}^{(j^*)}, \mathbf{x}^\dagger) \text{ for some } j^* \in \{1, \dots, u\}.$$

Using (2), we obtain

$$\begin{aligned} & H_1(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}^*) + H_2(\tilde{\mathbf{A}}^{(2)}, \mathbf{x}^*) + \dots + H_u(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}^*) \\ & < H_1(\tilde{\mathbf{A}}^{(1)}, \mathbf{x}^\dagger) + H_2(\tilde{\mathbf{A}}^{(2)}, \mathbf{x}^\dagger) + \dots + H_u(\tilde{\mathbf{A}}^{(u)}, \mathbf{x}^\dagger). \end{aligned}$$

Since \mathbf{x}^* is an optimal solution of problem (SOP), we lead to a contradiction. This completes the proof. \square

The assignment of coefficients w_{jk} for $j = 1, \dots, u$ and $k = 1, \dots, 2m$ are frequently determined by the decision makers via intuition. We may argue that this kind of determination can subject to bias caused by the decision makers. Therefore, a mechanical way is suggested in this paper by following the solution concept of game theory to determine the coefficients w_{jk} . The main idea is that the objective functions $\tilde{H}_{\alpha_i}^{(jL)}$ and $\tilde{H}_{\alpha_i}^{(jU)}$ are treated as the payoffs of players.

3. Cores of Cooperative Games

Given a set $N = \{1, \dots, n\}$ of players, any nonempty subset S of N is called a coalition. We consider the function $v : 2^N \rightarrow \mathbb{R}$ satisfying $v(\emptyset) = 0$. A cooperative game is defined to be an ordered pair (N, v) . Given any coalition S , the function value $v(S)$ is interpreted as the worth of coalition S in the game (N, v) .

Given a payoff vector (or an allocation) $\mathbf{x} \in \mathbb{R}^n$, each x_i represents the value received by player i for $i = 1, \dots, n$. Many concepts are defined below.

- We say that the vector $\mathbf{x} \in \mathbb{R}^n$ is a pre-imputation when the following group rationality is satisfied

$$v(N) = \sum_{i \in N} x_i.$$

- We say that the vector $\mathbf{x} \in \mathbb{R}^n$ is an imputation when it is a pre-imputation and satisfies the following individual rationality

$$x_i \geq v(i) \text{ for } i \in N.$$

The set of all imputations is denoted by $\mathcal{I}(v)$, and the set of all pre-imputations is denoted by $\mathcal{I}^*(v)$.

Given a coalition S and a payoff vector \mathbf{x} , the excess of S with respect to \mathbf{x} is defined by

$$e(S, \mathbf{x}) = v(S) - \sum_{i \in S} x_i.$$

Now, the core of a game (N, v) is defined by

$$\begin{aligned} \mathcal{C}(v) &= \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i \in N} x_i = v(N) \text{ and } \sum_{i \in S} x_i \geq v(S) \text{ for all } S \subseteq N \right\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i \in N} x_i = v(N) \text{ and } \sum_{i \in S} x_i \geq v(S) \text{ for all } S \subset N \text{ and } S \neq N \right\}. \end{aligned}$$

It is also clear to see

$$\begin{aligned} \mathcal{C}(v) &= \{ \mathbf{x} \in \mathcal{I}(v) : e(S, \mathbf{x}) \leq 0 \text{ for all } S \subseteq N \} \\ &= \{ \mathbf{x} \in \mathcal{I}^*(v) : e(S, \mathbf{x}) \leq 0 \text{ for all } S \subseteq N \}. \end{aligned}$$

We can see that the core of a game (N, v) is a set of all imputations such that only nonpositive excesses are taken into account.

4. Formulation of Cooperative Game

Given any fixed j , the function $\tilde{H}_{\alpha_i}^{(jL)}$ is treated as the payoff of player (i, j) for $i = 1, \dots, m$, and the function $\tilde{H}_{\alpha_i}^{(jU)}$ is treated as the payoff of player $(m + i, j)$ for $i = 1, \dots, m$. In this case, the player is taken to be an ordered pair (i, j) . Therefore, the set of all players is given by

$$N = \{(i, j) : i = 1, \dots, 2m \text{ and } j = 1, \dots, u\}.$$

We have total $2mu$ players corresponding to $2mu$ functions. Let

$$d_{ij}^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_i}^{(jL)}(\mathbf{x}) \text{ and } d_{m+i,j}^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_i}^{(jU)}(\mathbf{x}) \text{ for } i = 1, \dots, m,$$

where d_{ij}^* are regarded as the ideal payoffs for $i = 1, \dots, 2m$ and $j = 1, \dots, u$. Therefore, we must assume

$$v(\{(i, j)\}) \leq d_{ij}^* \text{ for all } i = 1, \dots, 2m \text{ and } j = 1, \dots, u,$$

which means that the true payoff $v(\{(i, j)\})$ of player (i, j) may not reach the ideal payoff d_{ij}^* .

Given $S \subseteq N$ with $s = |S|$, this coalition is written as $S = \{i_1, i_2, \dots, i_s\}$. By intuition, the payoff of coalition S must be larger than the total payoffs of each player in S such that the cooperation is meaningful. In other words, the following inequality

$$v(S) \geq v(\{i_1\}) + v(\{i_2\}) + \dots + v(\{i_s\})$$

must be satisfied. Also, the payoff of coalition S cannot be larger than the total ideal payoffs on S , which means that the payoff of coalition may not reach the total ideal payoffs. That is to say, the following inequalities must be satisfied:

$$v(\{i_1\}) + v(\{i_2\}) + \dots + v(\{i_s\}) \leq v(S) \leq d_{i_1}^* + d_{i_2}^* \dots + d_{i_s}^* \tag{6}$$

Now, the payoff of coalition S with $|S| \geq 2$ is defined by

$$v(S) = \sum_{k=1}^s v(\{i_k\}) + \frac{\gamma_s}{s} \sum_{k=1}^s v(\{i_k\}), \tag{7}$$

where γ_s is a non-negative constant. The second term

$$\frac{\gamma_s}{s} \sum_{k=1}^s v(\{i_k\})$$

can be treated as the extra payoff subject to cooperation by forming a coalition S with $|S| = s$. We assume that this extra payoff is obtained by taking a non-negative constant γ_s that multiplies the average of individual payoffs. In this situation, the non-negative constant γ_s should be independent of any coalition S with $|S| = s$.

According to the upper bound of $v(S)$ given in (6), the constant γ_s must satisfy

$$\sum_{k=1}^s v(\{i_k\}) + \frac{\gamma_s}{s} \sum_{k=1}^s v(\{i_k\}) \leq \sum_{k=1}^s d_{i_k}^*.$$

Equivalently, we obtain

$$0 \leq \gamma_s \leq \frac{s \cdot \sum_{k=1}^s (d_{i_k}^* - v(\{i_k\}))}{\sum_{k=1}^s v(\{i_k\})} = \frac{s \cdot \sum_{k=1}^s d_{i_k}^*}{\sum_{k=1}^s v(\{i_k\})} - s \equiv \mathcal{V}_s(S) \tag{8}$$

for $s = 2, \dots, 2mu$. Now, we define

$$\mathcal{V}_s = \min\{\mathcal{V}_s(S) : S \subseteq N \text{ with } |S| = s\}. \tag{9}$$

Then, we have

$$0 \leq \gamma_s \leq \mathcal{V}_s \tag{10}$$

for $s = 2, \dots, 2mu$. We also define $\gamma_1 = 0$ for convenience.

5. Solving the Problem Using Core Values

Recall that the core of cooperative game (N, v) is given by

$$\mathcal{C}(v) = \left\{ \mathbf{w} \in \mathbb{R}^{2mu} : \sum_{i \in N} w_i = v(N) \text{ and } \sum_{i \in S} w_i \geq v(S) \text{ for all } S \subset N \text{ with } S \neq N \right\}.$$

Using (7), it follows that $\mathbf{w} \in \mathcal{C}(v)$ if and only if \mathbf{w} satisfies

$$\sum_{i \in N} w_i = \sum_{k=1}^{2mu} v(\{i_k\}) + \frac{\gamma_{2mu}}{2mu} \sum_{k=1}^{2mu} v(\{i_k\})$$

and

$$\sum_{i \in S} w_i \geq \sum_{k=1}^s v(\{i_k\}) + \frac{\gamma_s}{s} \sum_{k=1}^s v(\{i_k\}) \text{ for all } S \subseteq N \text{ with } S \neq N.$$

Since the payoffs $v(S)$ are non-negative for $S \subseteq N$, the positive core $\mathcal{C}^+(v)$ is defined by

$$\begin{aligned} \mathcal{C}^+(v) &= \mathcal{C}(v) \cap \mathbb{R}_+^{2mu} \\ &= \left\{ \mathbf{w} \in \mathbb{R}_+^{2mu} : \sum_{i \in N} w_i = v(N) \text{ and } \sum_{i \in S} w_i \geq v(S) \text{ for all } S \subseteq N \text{ with } S \neq N \right\}. \end{aligned}$$

We normalize the values of w_i to be \bar{w}_i given by

$$\bar{w}_i = \frac{w_i}{\sum_{j=1}^{2mu} w_j} \text{ for } i = 1, \dots, 2mu.$$

In this case, we also write $\bar{\mathcal{C}}^+(v)$ to denote the set of all normalized values $\bar{\mathbf{w}}$ obtained from $\mathbf{w} \in \mathcal{C}^+(v)$.

Now, the following linear programming problem

$$\begin{aligned} \text{(LP)} \quad & \text{minimize} \quad \sum_{i \in N} w_i \\ & \text{subject to} \quad \sum_{i \in S} w_i \geq v(S) \text{ for all } S \subseteq N \\ & \quad \quad \quad w_i \geq 0 \text{ for } i = 1, \dots, 2mu \end{aligned} \tag{11}$$

is considered. The following property is useful for further study.

Proposition 1. *Let \mathbf{w}^* be an optimal solution of problem (LP). If $v(N) \geq v(S)$ for all $S \subseteq N$ then $\mathbf{w}^* \in \mathcal{C}^+(v)$.*

Proof. Since \mathbf{w}^* is a feasible solution of problem (LP), we have

$$\sum_{i \in S} w_i^* \geq v(S) \text{ for all } S \subseteq N. \tag{12}$$

We take $\bar{w}_1 = v(N)$ and $\bar{w}_i = 0$ for $s = 2, 3, \dots, 2mu$. Since $v(N) \geq v(S)$ for all $S \subseteq N$, it follows that

$$\bar{\mathbf{w}} = (\bar{w}_1, \dots, \bar{w}_{2mu}) = (v(N), 0, \dots, 0)$$

is a feasible solution of problem (LP) with objective value $v(N)$, which implies

$$\sum_{i \in N} w_i^* \leq v(N).$$

By taking $S = N$ in (12), we also have

$$\sum_{i \in N} w_i^* = v(N),$$

Therefore, we obtain $\mathbf{w}^* \in \mathcal{C}^+(v)$. This completes the proof. \square

From the payoff defined in (7), the formulation of cooperative game (N, v) is determined by the non-negative constants γ_s for $s = 1, \dots, 2mu$, which also says that the payoff function v must be determined by the vector $\gamma = (\gamma_1, \dots, \gamma_{2mu})$. We also write

$$\mathcal{C}^+(v) = \mathcal{C}^+(\gamma) \text{ and } \bar{\mathcal{C}}^+(v) = \bar{\mathcal{C}}^+(\gamma).$$

Proposition 1 shows that the optimal solution \mathbf{w}^* is determined by the payoff functions v . In this case, we can write $\mathbf{w}^*(\gamma)$ in Proposition 1. Then, the normalized values $\bar{\mathbf{w}}^*(\gamma) \in \bar{\mathcal{C}}^+(\gamma)$ are given by

$$\bar{w}_i^*(\gamma) = \frac{w_i^*(\gamma)}{\sum_{j=1}^{2mu} w_j^*(\gamma)} \text{ for } i = 1, \dots, 2mu, \tag{13}$$

Now, we assign the normalized core values given in (13) to the coefficients of scalar optimization problem (SOP). In this case, according to Theorem 1, the optimal solution of problem (SOP) is called the core-nondominated solution of problem (FMOP), which means that the solution concept of core is involved. Therefore, we can solve the following scalar optimization problem

$$\begin{aligned} \text{(SOP) maximize} \quad & \sum_{j=1}^u \sum_{i=1}^m \bar{w}_i^*(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jL)}(\tilde{\mathbf{A}}^{(i)}, \mathbf{x}) + \sum_{j=1}^u \sum_{i=1}^m \bar{w}_{m+i}^*(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jU)}(\tilde{\mathbf{A}}^{(m+i)}, \mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in X \subseteq \mathbb{R}^{2mu} \end{aligned}$$

to obtain the core-nondominated solution. Since the core-nondominated solution depends on the vector γ of non-negative constants, we also write $\mathbf{x}^*(\gamma)$ for convenience.

Let \mathfrak{P} be the set of all core-nondominated solutions of problem (FMOP). From (10), we have

$$\mathfrak{P} = \{\mathbf{x}^*(\gamma) : 0 \leq \gamma_s \leq \mathcal{V}_s \text{ for } s = 1, \dots, 2mu\}.$$

Since \mathfrak{P} is a large set, we intend to find a best core-nondominated solution from \mathfrak{P} by using the genetic algorithm. In this case, we plan to maximize the following fitness function

$$\eta(\gamma) = \sum_{j=1}^u \sum_{i=1}^m \bar{w}_i^*(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jL)}(\tilde{\mathbf{A}}^{(i)}, \mathbf{x}^*(\gamma)) + \sum_{j=1}^u \sum_{i=1}^m \bar{w}_{m+i}^*(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jU)}(\tilde{\mathbf{A}}^{(m+i)}, \mathbf{x}^*(\gamma)),$$

where $0 \leq \gamma_s \leq \mathcal{V}_s$ for $s = 1, \dots, 2mu$.

6. Genetic Algorithms

The purpose is to design an genetic algorithm such that a best core-nondominated solution can be obtained from the set \mathfrak{P} of all core-nondominated solutions of problem (FMOP). Therefore, we are going to maximize the following fitness function

$$\eta(\gamma) = \sum_{j=1}^u \sum_{i=1}^m \bar{w}_i(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jL)}(\tilde{\mathbf{A}}^{(i)}, \mathbf{x}^*(\gamma)) + \sum_{j=1}^u \sum_{i=1}^m \bar{w}_{m+i}(\gamma) \cdot \tilde{H}_{\alpha_i}^{(jU)}(\tilde{\mathbf{A}}^{(m+i)}, \mathbf{x}^*(\gamma)). \tag{14}$$

We shall evolve the non-negative vector γ by performing crossover and mutation to find a best chromosome according to the fitness function given in (14).

From (10), the non-negative constants γ_s must satisfy

$$0 \leq \gamma_s \leq \mathcal{V}_s \text{ for } s = 2, \dots, 2mu.$$

In this algorithm, each non-negative constant γ_s will be a random number from the closed interval $[0, \mathcal{V}_s]$ for $s = 2, \dots, 2mu$. The chromosome in this algorithm is defined to be a vector $\gamma = (0, \gamma_2, \dots, \gamma_{2mu})$ in \mathbb{R}^{2mu} satisfying $\gamma_1 = 0$ and $\gamma_s \in [0, \mathcal{V}_s]$ for $s = 2, \dots, 2mu$.

Two phases will be performed in this algorithm. Since the scalar optimization problem (SOP) depends on the partition $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ of $[0, 1]$, phase I will obtain

the approximated best core-nondominated solution when the partition Γ is taken to be fixed. In phase II, we shall use the more finer partition Γ of $[0, 1]$ to perform the algorithm in phase I until the approximated best core-nondominated solution cannot be improved.

6.1. Phase I

In phase I, given a fixed partition $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ of $[0, 1]$, we are going to obtain the approximated best core-nondominated solution by solving the scalar optimization problem (SOP) through performing crossover and mutation operations.

Proposition 2. (Crossover) Let $\hat{\gamma} = (0, \hat{\gamma}_2, \dots, \hat{\gamma}_{2mu})$ and $\bar{\gamma} = (0, \bar{\gamma}_2, \dots, \bar{\gamma}_{2mu})$ be two chromosomes satisfying

$$0 \leq \hat{\gamma}_s \leq \mathcal{V}_s \text{ and } 0 \leq \bar{\gamma}_s \leq \mathcal{V}_s \text{ for } s = 2, \dots, 2mu.$$

Given any $\lambda \in (0, 1)$, we consider the following crossover operation

$$\gamma = \lambda \hat{\gamma} + (1 - \lambda) \bar{\gamma},$$

where the components are given by

$$\gamma_1 = 0 \text{ and } \gamma_s = \lambda \hat{\gamma}_s + (1 - \lambda) \bar{\gamma}_s \text{ for } s = 2, \dots, 2mu.$$

Then, the new chromosome γ also satisfies $0 \leq \gamma_s \leq \mathcal{V}_s$ for $s = 2, \dots, 2mu$.

Proof. Since $0 \leq \hat{\gamma}_s \leq \mathcal{V}_s$ and $0 \leq \bar{\gamma}_s \leq \mathcal{V}_s$, it is clear to see that the convex combination $\gamma_s = \lambda \hat{\gamma}_s + (1 - \lambda) \bar{\gamma}_s$ also satisfies $0 \leq \gamma_s \leq \mathcal{V}_s$. \square

Given a vector $\bar{\gamma} = (0, \bar{\gamma}_2, \dots, \bar{\gamma}_{2mu})$, we shall perform the mutation to obtain γ from $\bar{\gamma}$. Given a fixed $s = 2, \dots, 2mu$, we first generate a random Gaussian number with mean zero and standard deviation σ_s . Then, we assign

$$\hat{\gamma}_s = \bar{\gamma}_s + N(0, \sigma_s^2) = \bar{\gamma}_s + \sigma_s \cdot N(0, 1).$$

The new mutated chromosome γ is defined by

$$\gamma_s = \begin{cases} \hat{\gamma}_s & \text{if } \hat{\gamma}_s \in [0, \mathcal{V}_s] \\ \mathcal{V}_s & \text{if } \hat{\gamma}_s > \mathcal{V}_s \\ 0 & \text{if } \hat{\gamma}_s < 0 \end{cases} \tag{15}$$

where \mathcal{V}_s is given in (9) for $s = 2, \dots, 2mu$.

Proposition 3. (Mutation) Suppose that $\bar{\gamma} = (0, \bar{\gamma}_2, \dots, \bar{\gamma}_{2mu})$ is a chromosome. We consider the mutation in the way of (15). Then, the new mutated chromosome γ satisfies

$$0 \leq \gamma_s \leq \mathcal{V}_s \text{ for } s = 2, \dots, 2mu.$$

Proof. It is clear to see from (15). \square

6.2. Phase II

In phase II, we shall use the more finer partition Γ of $[0, 1]$ to perform the algorithm proposed in phase I. Assume that the partition $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ of $[0, 1]$ was considered in phase I. Now, given a new partition $\bar{\Gamma} = \{0 = \alpha_1, \alpha_2, \dots, \alpha_n = 1\}$ of $[0, 1]$, which is finer than Γ in the sense of $\Gamma \subset \bar{\Gamma}$. We shall perform the algorithm proposed in phase I using this new partition $\bar{\Gamma}$. In other words, we are going to continuously execute Phase II by using the more finer partitions such that the approximated best core-nondominated solution cannot be improved. Two ways are suggested in this paper to obtain the finer partition $\bar{\Gamma}$ as follows.

For the first way, we can simply take the partition $\bar{\Gamma}$ of $[0, 1]$ such that $[0, 1]$ is equally divided and satisfies $\Gamma \subset \bar{\Gamma}$. The second way is to design another genetic algorithm to generate a new finer partition $\bar{\Gamma}$ by evolving the old partition Γ . More precisely, a population $\mathcal{P} = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ is taken from the old partition Γ . After performing the crossover and mutation operations in the old population \mathcal{P} , we can generate new points β_1, \dots, β_r such that a new finer partition $\bar{\Gamma}$ is obtained as follows

$$\bar{\Gamma} = \Gamma \cup \{\beta_1, \dots, \beta_r\}.$$

For example, the crossover and mutation operations can be designed as follows.

- (Crossover operation). Take two α_s and α_t from the old partition \mathcal{P} . We perform the convex combination $\lambda\alpha_s + (1 - \lambda)\alpha_t$ for different $\lambda \in (0, 1)$, which can generate the different new points.
- (Mutation operation). Take α_s from \mathcal{P} . We perform the mutation $\alpha_s + \delta$, which δ is a random number in $[0, 1]$. When $\alpha_s + \delta$ is in $[0, 1]$, the new generated point is taken to be $\alpha_s + \delta$. When $\alpha_s + \delta > 1$, the new generated point is taken to be $\alpha_s + \delta - 1$.

When a new finer partition $\bar{\Gamma}$ is generated, the algorithm in phase I is again performed using this new partition $\bar{\Gamma}$. In this case, we can obtain a new approximated best core-nondominated solution. Also, this partition $\bar{\Gamma}$ is now regarded as the old partition. Afterward, a new finer partition $\hat{\Gamma}$ of $[0, 1]$ is also generated to satisfy $\bar{\Gamma} \subset \hat{\Gamma}$. Now, we again perform the algorithms in phase I using this new finer partition $\hat{\Gamma}$.

6.3. Computational Procedure

Given a fixed partition $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ of $[0, 1]$, We present the detailed computational procedure of genetic algorithm for phase I as follows.

- **Step 1 (Initialization).** The population size of this algorithm is assumed to be p . The chromosomes in the initial population is determined by setting $\gamma^{(k)} = (\gamma_1^{(k)}, \dots, \gamma_{2mu}^{(k)})$, where $\gamma_1^{(k)} = 0$ and $\gamma_s^{(k)}$ are taken to be the random numbers in $[0, \mathcal{V}_{2mu}]$ for all $k = 1, \dots, p$ and $s = 2, \dots, 2mu$, where \mathcal{V}_s is given in (9). For each $\gamma^{(k)}$, we solve the linear programming problem in (11) and use Proposition 1 to obtain the positive core $\mathbf{w}^*(\gamma^{(k)}) \in \mathcal{C}^+(v)$ for $j = 1, \dots, p$. Using (13), we also calculate the normalized positive core $\bar{w}_i^*(\gamma^{(k)})$ for $i = 1, \dots, 2mu$ and $k = 1, \dots, p$. For each chromosome $\gamma^{(k)}$, we assign its corresponding fitness value by calculating the following expression

$$\begin{aligned} \eta(\gamma^{(k)}) = & \sum_{j=1}^u \sum_{i=1}^m \bar{w}_i^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_i}^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*(\gamma^{(k)})) \\ & + \sum_{j=1}^u \sum_{i=1}^m \bar{w}_{m+i}^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_i}^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*(\gamma^{(k)})) \end{aligned}$$

for $k = 1, \dots, p$. Then, the p chromosomes $\gamma^{(k)}$ for $k = 1, \dots, p$ are ranked in descending order according to their fitness values $\eta(\gamma^{(k)})$ for $k = 1, \dots, p$. In this case, the top one is saved to be the (initial) best chromosome and is named as $\bar{\eta}_0$. We also save $\gamma^{(k)}$ to be the old elites $\gamma^{(*k)}$ given by $\gamma_s^{(*k)} \leftarrow \gamma_s^{(j)}$ for $s = 1, \dots, 2mu$ and $k = 1, \dots, p$.

- **Step 2 (Tolerance).** We set the tolerance ϵ . We also set the number m^* of maximum times of iterations such that the tolerance ϵ is satisfied. We set $t = 0$ to mean the initial generation. We set $l^* = 1$ to mean the first time such that the tolerance ϵ is satisfied. This step is related with the stopping criterion to avoid to be trapped in the local optimum. Therefore, it can be more clear by referring to step 7.
- **Step 3 (Mutation).** We set $t \leftarrow t + 1$ to mean the t -th generation. By referring to Proposition 3, each chromosome $\gamma^{(k)}$ is mutated, and is assigned to $\gamma^{(k+p)}$ using (15)

for $k = 1, \dots, p$. For each $s = 2, \dots, 2mu$, the random Gaussian numbers with mean zero and standard deviation σ_s are generated in which σ_s is taken by

$$\sigma_s = \beta_s \cdot \eta(\gamma^{(k)}) + b_s.$$

The constant β_s is the proportionality to scale $\eta(\gamma^{(k)})$ and the constant b_s represents the offset. Then, we assign

$$\widehat{\gamma}_s^{(k)} = \gamma_s^{(k)} + \sigma_s \cdot N(0, 1) = \gamma_s^{(k)} + (\beta_s \cdot \eta(\gamma^{(k)}) + b_s) \cdot N(0, 1).$$

In this case, the mutated chromosome $\gamma^{(k+p)}$ is obtained in which the components are given by $\gamma_1^{(k+p)} = 0$ and

$$\gamma_s^{(k+p)} = \begin{cases} \widehat{\gamma}_s^{(k)} & \text{if } \widehat{\gamma}_s^{(k)} \in [0, \mathcal{V}_s] \\ \mathcal{V}_s & \text{if } \widehat{\gamma}_s^{(k)} > \mathcal{V}_s \\ 0 & \text{if } \widehat{\gamma}_s^{(k)} < 0 \end{cases}$$

for $s = 2, \dots, 2mu$ and $k = 1, \dots, p$. After this mutation step, we shall have $2p$ chromosomes $\gamma^{(k)}$ for $k = 1, \dots, 2p$.

- **Step 4 (Crossover).** By referring to Proposition 2, randomly select $\gamma^{(k_1)}$ and $\gamma^{(k_2)}$ for $k_1, k_2 \in \{1, \dots, 2p\}$ with $k_1 \neq k_2$. A random number $\lambda \in (0, 1)$ is generated. In this case, the new chromosome is taken by

$$\gamma^{(2p+1)} = \lambda \gamma^{(k_1)} + (1 - \lambda) \gamma^{(k_2)},$$

where the components are given by

$$\gamma_s^{(2p+1)} = \lambda \gamma_s^{(k_1)} + (1 - \lambda) \gamma_s^{(k_2)} \in [0, \mathcal{V}_s] \text{ for } s = 1, \dots, 2mu.$$

After this step, we shall have $2p + 1$ chromosomes $\gamma^{(k)}$ for $j = 1, \dots, 2p + 1$.

- **Step 5 (Calculate New Fitness).** Using Proposition 1 and (13), for each new chromosome $\gamma^{(k+p)}$, we calculate the normalized positive core $\bar{w}_i^*(\gamma^{(k+p)})$ by solving the linear programming problem in (11) for $i = 1, \dots, 2mu$ and $k = 1, \dots, p + 1$. For each $\gamma^{(k+p)}$, we assign its corresponding fitness value by calculating the following expression

$$\eta(\gamma^{(k+p)}) = \sum_{j=1}^u \sum_{i=1}^m \bar{w}_i^*(\gamma^{(k+p)}) \cdot \tilde{H}_{\alpha_i}^{(jL)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*(\gamma^{(k+p)})) + \sum_{j=1}^u \sum_{i=1}^m \bar{w}_{m+i}^*(\gamma^{(k+p)}) \cdot \tilde{H}_{\alpha_i}^{(jU)}(\tilde{\mathbf{A}}^{(j)}, \mathbf{x}^*(\gamma^{(k+p)}))$$

for $k = 1, \dots, p + 1$.

- **Step 6 (Selection).** The p old elites $\gamma^{(*k)}$ for $j = 1, \dots, p$ and the $p + 1$ new chromosomes $\gamma^{(k)}$ for $k = 1, \dots, p + 1$ obtained from Steps 3 and 4 are ranked in descending order according to their fitness values $\eta(\gamma^{(*k)})$ and $\eta(\gamma^{(k)})$, respectively. In this case, the top p chromosomes are saved as the new elites $\gamma^{(*k)}$ for $k = 1, \dots, p$. Also, the top one is saved as the best chromosome that is named as $\bar{\eta}_t$ for the t -th generation.
- **Step 7 (Stopping Criterion).** After step 6, we may obtain $\bar{\eta}_{t-1} = \bar{\eta}_t$, which seems to be trapped in the local optimum. In order to escape this trap, we are going to proceed more iterations for m^* times by referring to step 2 even though the criterion $\|\bar{\eta}_t - \bar{\eta}_{t-1}\| < \epsilon$ is satisfied. When the criterion $\|\bar{\eta}_t - \bar{\eta}_{t-1}\| < \epsilon$ is satisfied and the iterations reach m^* times, we stop the algorithm and return the solution for phase I. Otherwise, the new elites $\gamma^{(*k)}$ for $k = 1, \dots, p$ are copied as the next generation $\gamma^{(k)}$

for $k = 1, \dots, p$. Then, we set $l^* \leftarrow l^* + 1$, and proceed to step 3. We also remark that the number l^* counts the times such that the tolerance $\|\tilde{\eta}_t - \tilde{\eta}_{t-1}\| < \epsilon$ is satisfied.

After step 7, given a fixed partition $\Gamma = \{0 = \alpha_1, \alpha_2, \dots, \alpha_m = 1\}$ of $[0, 1]$, an approximated best core-nondominated solution can be obtained, which is denoted by $\mathbf{x}^*(\Gamma)$. It also means that this solution is determined the partition Γ . Then, the algorithm proceeds to phase II by considering more finer partitions of $[0, 1]$ as follows.

- **Step 1.** A new finer partition $\bar{\Gamma}$ is generated to satisfy $\Gamma \subset \bar{\Gamma}$.
- **Step 2.** By using this new finer partition $\bar{\Gamma}$ to perform the genetic algorithm in phase I, we can obtain a new approximated best core-nondominated solution $\mathbf{x}^*(\bar{\Gamma})$.
- **Step 3.** Given a pre-determined tolerance ϵ , once the criterion $\|\mathbf{x}^*(\Gamma) - \mathbf{x}^*(\bar{\Gamma})\| < \epsilon$ is reached, the algorithm halts and returns the final solution $\mathbf{x}^*(\bar{\Gamma})$. Otherwise, $\bar{\Gamma}$ is set as the old partition Γ , and proceeds to step 1 to generate a new finer partition.

Finally, after step 3, we obtain the approximated best core-nondominated solution. In other words, by referring to Theorem 1, an approximated nondominated solution of problem (FMOP) is obtained.

7. Numerical Example

The membership function of triangular fuzzy interval $\tilde{A} = (a^L, a, a^U)$ is defined by

$$\zeta_{\tilde{A}}(r) = \begin{cases} \frac{r - a^L}{a - a^L} & \text{if } a^L \leq r \leq a \\ \frac{a^U - r}{a^U - a} & \text{if } a < r \leq a^U \\ 0 & \text{otherwise.} \end{cases}$$

Then, its α -level set $\tilde{A}_\alpha = [\tilde{A}_\alpha^L, \tilde{A}_\alpha^U]$ is given by

$$\tilde{A}_\alpha^L = (1 - \alpha)a^L + \alpha a \text{ and } \tilde{A}_\alpha^U = (1 - \alpha)a^U + \alpha a.$$

In particular, we consider the triangular fuzzy intervals as follows

$$\tilde{4} = (3.5, 4, 4.5), \quad \tilde{5} = (4, 5, 5.5) \text{ and } \tilde{6} = (5, 6, 7).$$

Then, their α -level sets are given by

$$\begin{aligned} \tilde{4}_\alpha &= [3.5 + 0.5 \cdot \alpha, 4.5 - 0.5 \cdot \alpha] \\ \tilde{5}_\alpha &= [4 + \alpha, 5.5 - 0.5 \cdot \alpha] \\ \tilde{6}_\alpha &= [5 + \alpha, 7 - \alpha]. \end{aligned}$$

In this case, the following fuzzy linear programming problem

$$\begin{aligned} \text{maximize} \quad & (\tilde{5} \otimes \tilde{1}_{x_1}) \oplus (\tilde{4} \otimes \tilde{1}_{x_2}) \oplus (\tilde{6} \otimes \tilde{1}_{x_3}) \\ \text{subject to} \quad & x_1 - x_2 + x_3 \leq 20 \\ & 3x_1 + 2x_2 + 4x_3 \leq 42 \\ & 3x_1 + 2x_2 \leq 30 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

will be solved. According to the above formulation, we equally divide the unit closed interval $[0, 1]$ by taking $\alpha_1 = 0, \alpha_2 = 0.5$ and $\alpha_3 = 1$. Let $\tilde{\mathbf{A}} = (\tilde{5}, \tilde{4}, \tilde{6})$. Using the above notations, we obtain

$$\begin{aligned} \tilde{H}_{\alpha_1}^L(\tilde{\mathbf{A}}, \mathbf{x}) &= \tilde{H}_0^L(\tilde{\mathbf{A}}, \mathbf{x}) = 3.5x_1 + 4x_2 + 5x_3 \\ \tilde{H}_{\alpha_2}^L(\tilde{\mathbf{A}}, \mathbf{x}) &= \tilde{H}_{0.5}^L(\tilde{\mathbf{A}}, \mathbf{x}) = 3.75x_1 + 4.5x_2 + 5.5x_3 \end{aligned}$$

$$\tilde{H}_{\alpha_3}^L(\tilde{\mathbf{A}}, \mathbf{x}) = \tilde{H}_1^L(\tilde{\mathbf{A}}, \mathbf{x}) = 4x_1 + 5x_2 + 6x_3$$

and

$$\tilde{H}_{\alpha_1}^U(\tilde{\mathbf{A}}, \mathbf{x}) = \tilde{H}_0^U(\tilde{\mathbf{A}}, \mathbf{x}) = 4.5x_1 + 5.5x_2 + 7x_3$$

$$\tilde{H}_{\alpha_2}^U(\tilde{\mathbf{A}}, \mathbf{x}) = \tilde{H}_{0.5}^U(\tilde{\mathbf{A}}, \mathbf{x}) = 4.25x_1 + 5.25x_2 + 6.5x_3$$

$$\tilde{H}_{\alpha_3}^U(\tilde{\mathbf{A}}, \mathbf{x}) = \tilde{H}_1^U(\tilde{\mathbf{A}}, \mathbf{x}) = 4x_1 + 5x_2 + 6x_3$$

We see that $\tilde{H}_{\alpha_3}^L(\tilde{\mathbf{A}}, \mathbf{x}) = \tilde{H}_{\alpha_3}^U(\tilde{\mathbf{A}}, \mathbf{x})$. Next, we are going to solve the following scalar optimization problem

$$\begin{aligned} &\text{maximize} && w_1\tilde{H}_{\alpha_1}^L(\tilde{\mathbf{A}}, \mathbf{x}) + w_2\tilde{H}_{\alpha_2}^L(\tilde{\mathbf{A}}, \mathbf{x}) + w_3\tilde{H}_{\alpha_3}^L(\tilde{\mathbf{A}}, \mathbf{x}) + w_4\tilde{H}_{\alpha_1}^U(\tilde{\mathbf{A}}, \mathbf{x}) + w_5\tilde{H}_{\alpha_2}^U(\tilde{\mathbf{A}}, \mathbf{x}) \\ &\text{subject to} && (x_1, x_2, x_3) \in X, \end{aligned}$$

where the feasible set X is given by

$$X = \left\{ (x_1, x_2, x_3) \in \mathbb{R}_+^3 : x_1 - x_2 + x_3 \leq 20, 3x_1 + 2x_2 + 4x_3 \leq 42 \text{ and } 3x_1 + 2x_2 \leq 30 \right\}.$$

In order to formulate the corresponding cooperative game, we must obtain the ideal objective values $\mathbf{d}^* = (d_1^*, \dots, d_5^*)$ given by

$$d_1^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_1}^L(\tilde{\mathbf{A}}, \mathbf{x}) = 75$$

$$d_2^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_2}^L(\tilde{\mathbf{A}}, \mathbf{x}) = 84$$

$$d_3^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_3}^L(\tilde{\mathbf{A}}, \mathbf{x}) = 93$$

$$d_4^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_1}^U(\tilde{\mathbf{A}}, \mathbf{x}) = 103.5$$

$$d_5^* = \sup_{\mathbf{x} \in X} \tilde{H}_{\alpha_2}^U(\tilde{\mathbf{A}}, \mathbf{x}) = 98.25$$

Therefore, we consider five players $N = \{1, 2, 3, 4, 5\}$ such the cooperative game (N, v) is defined by

$$v(\{1\}) = 0.5 \cdot d_1^*, \quad v(\{2\}) = 0.6 \cdot d_2^* \text{ and } v(\{3\}) = 0.7 \cdot d_3^*$$

and

$$v(\{4\}) = 0.5 \cdot d_4^* \text{ and } v(\{5\}) = 0.7 \cdot d_5^*.$$

By referring to (8), for $|S| = 2$, we have

$$\mathcal{V}_2(\{1, 2\}) = \frac{2 \cdot (d_1^* + d_2^*)}{v(\{1\}) + v(\{2\})} - 2 = \frac{2 \cdot (d_1^* + d_2^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^*} - 2 = 1.61774$$

$$\mathcal{V}_2(\{1, 3\}) = \frac{2 \cdot (d_1^* + d_3^*)}{v(\{1\}) + v(\{3\})} - 2 = \frac{2 \cdot (d_1^* + d_3^*)}{0.5 \cdot d_1^* + 0.7 \cdot d_3^*} - 2 = 1.27485$$

$$\mathcal{V}_2(\{1, 4\}) = \frac{2 \cdot (d_1^* + d_4^*)}{v(\{1\}) + v(\{4\})} - 2 = \frac{2 \cdot (d_1^* + d_4^*)}{0.5 \cdot d_1^* + 0.5 \cdot d_4^*} - 2 = 2$$

$$\mathcal{V}_2(\{1, 5\}) = \frac{2 \cdot (d_1^* + d_5^*)}{v(\{1\}) + v(\{5\})} - 2 = \frac{2 \cdot (d_1^* + d_5^*)}{0.5 \cdot d_1^* + 0.7 \cdot d_5^*} - 2 = 1.26041$$

$$\mathcal{V}_2(\{2, 3\}) = \frac{2 \cdot (d_2^* + d_3^*)}{v(\{2\}) + v(\{3\})} - 2 = \frac{2 \cdot (d_2^* + d_3^*)}{0.6 \cdot d_2^* + 0.7 \cdot d_3^*} - 2 = 1.06493$$

$$\mathcal{V}_2(\{2, 4\}) = \frac{2 \cdot (d_2^* + d_4^*)}{v(\{2\}) + v(\{4\})} - 2 = \frac{2 \cdot (d_2^* + d_4^*)}{0.6 \cdot d_2^* + 0.5 \cdot d_4^*} - 2 = 1.67107$$

$$\mathcal{V}_2(\{2,5\}) = \frac{2 \cdot (d_2^* + d_5^*)}{v(\{2\}) + v(\{5\})} - 2 = \frac{2 \cdot (d_2^* + d_5^*)}{0.6 \cdot d_2^* + 0.7 \cdot d_5^*} - 2 = 1.05853$$

$$\mathcal{V}_2(\{3,4\}) = \frac{2 \cdot (d_3^* + d_4^*)}{v(\{3\}) + v(\{4\})} - 2 = \frac{2 \cdot (d_3^* + d_4^*)}{0.7 \cdot d_3^* + 0.5 \cdot d_4^*} - 2 = 1.36329$$

$$\mathcal{V}_2(\{3,5\}) = \frac{2 \cdot (d_3^* + d_5^*)}{v(\{3\}) + v(\{5\})} - 2 = \frac{2 \cdot (d_3^* + d_5^*)}{0.7 \cdot d_3^* + 0.7 \cdot d_5^*} - 2 = 0.85714$$

$$\mathcal{V}_2(\{4,5\}) = \frac{2 \cdot (d_4^* + d_5^*)}{v(\{4\}) + v(\{5\})} - 2 = \frac{2 \cdot (d_4^* + d_5^*)}{0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 2 = 1.34785.$$

Using (9), we obtain

$$\mathcal{V}_2 = \min\{\mathcal{V}_2(\{1,2\}), \mathcal{V}_2(\{1,3\}), \mathcal{V}_2(\{1,4\}), \mathcal{V}_2(\{1,5\}), \mathcal{V}_2(\{2,3\}), \mathcal{V}_2(\{2,4\}), \mathcal{V}_2(\{2,5\}), \mathcal{V}_2(\{3,4\}), \mathcal{V}_2(\{3,5\}), \mathcal{V}_2(\{4,5\})\} = 0.85714.$$

For $|S| = 3$, we have

$$\mathcal{V}_3(\{1,2,3\}) = \frac{3 \cdot (d_1^* + d_2^* + d_3^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^*} - 3 = 1.94117$$

$$\mathcal{V}_3(\{1,2,4\}) = \frac{3 \cdot (d_1^* + d_2^* + d_4^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^*} - 3 = 2.63909$$

$$\mathcal{V}_3(\{1,2,5\}) = \frac{3 \cdot (d_1^* + d_2^* + d_5^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_5^*} - 3 = 1.92580$$

$$\mathcal{V}_3(\{1,3,4\}) = \frac{3 \cdot (d_1^* + d_3^* + d_4^*)}{0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*} - 3 = 2.27697$$

$$\mathcal{V}_3(\{1,3,5\}) = \frac{3 \cdot (d_1^* + d_3^* + d_5^*)}{0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*} - 3 = 1.66083$$

$$\mathcal{V}_3(\{1,4,5\}) = \frac{3 \cdot (d_1^* + d_4^* + d_5^*)}{0.5 \cdot d_1^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 3 = 2.25391$$

$$\mathcal{V}_3(\{2,3,4\}) = \frac{3 \cdot (d_2^* + d_3^* + d_4^*)}{0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*} - 3 = 2.03139$$

$$\mathcal{V}_3(\{2,3,5\}) = \frac{3 \cdot (d_2^* + d_3^* + d_5^*)}{0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*} - 3 = 1.48107$$

$$\mathcal{V}_3(\{2,4,5\}) = \frac{3 \cdot (d_2^* + d_4^* + d_5^*)}{0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 3 = 2.01536$$

$$\mathcal{V}_3(\{3,4,5\}) = \frac{3 \cdot (d_3^* + d_4^* + d_5^*)}{0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 3 = 1.76363.$$

Therefore, we obtain

$$\mathcal{V}_3 = \min\{\mathcal{V}_3(\{1,2,3\}), \mathcal{V}_3(\{1,2,4\}), \mathcal{V}_3(\{1,2,5\}), \mathcal{V}_3(\{1,3,4\}), \mathcal{V}_3(\{1,3,5\}), \mathcal{V}_3(\{1,4,5\}), \mathcal{V}_2(\{2,3,4\}), \mathcal{V}_2(\{2,3,5\}), \mathcal{V}_2(\{2,4,5\}), \mathcal{V}_2(\{3,4,5\})\} = 1.48107.$$

For $|S| = 4$, we have

$$\mathcal{V}_4(\{1,2,3,4\}) = \frac{4 \cdot (d_1^* + d_2^* + d_3^* + d_4^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*} - 4 = 2.94505$$

$$\mathcal{V}_4(\{1,2,3,5\}) = \frac{4 \cdot (d_1^* + d_2^* + d_3^* + d_5^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*} - 4 = 2.31721$$

$$\mathcal{V}_4(\{1,2,4,5\}) = \frac{4 \cdot (d_1^* + d_2^* + d_4^* + d_5^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 4 = 2.92335$$

$$\mathcal{V}_4(\{1,3,4,5\}) = \frac{4 \cdot (d_1^* + d_3^* + d_4^* + d_5^*)}{0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 4 = 2.62857$$

$$\mathcal{V}_4(\{2,3,4,5\}) = \frac{4 \cdot (d_2^* + d_3^* + d_4^* + d_5^*)}{0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 4 = 2.41881.$$

Therefore, we obtain

$$\mathcal{V}_4 = \min\{\mathcal{V}_4(\{1,2,3,4\}), \mathcal{V}_4(\{1,2,3,5\}), \mathcal{V}_4(\{1,2,4,5\}), \mathcal{V}_4(\{1,3,4,5\}), \mathcal{V}_4(\{2,3,4,5\})\} = 2.31721.$$

Finally, we obtain

$$\mathcal{V}_5 = \mathcal{V}_5(1,2,3,4,5) = \frac{5 \cdot (d_1^* + d_2^* + d_3^* + d_4^* + d_5^*)}{0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*} - 5 = 3.29449.$$

For $s = |S| \geq 2$, according to (7), we have

$$v(S) = \sum_{k=1}^s v(\{i_k\}) + \frac{\gamma_s}{s} \sum_{k=1}^s v(\{i_k\}).$$

More precisely, for $|S| = 2$, we have

$$v(\{1,2\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{1\}) + v(\{2\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^*)$$

$$v(\{1,3\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{1\}) + v(\{3\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.5 \cdot d_1^* + 0.7 \cdot d_3^*)$$

$$v(\{1,4\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{1\}) + v(\{4\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.5 \cdot d_1^* + 0.5 \cdot d_4^*)$$

$$v(\{1,5\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{1\}) + v(\{5\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.5 \cdot d_1^* + 0.7 \cdot d_5^*)$$

$$v(\{2,3\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{2\}) + v(\{3\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.6 \cdot d_2^* + 0.7 \cdot d_3^*)$$

$$v(\{2,4\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{2\}) + v(\{4\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.6 \cdot d_2^* + 0.5 \cdot d_4^*)$$

$$v(\{2,5\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{2\}) + v(\{5\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.6 \cdot d_2^* + 0.7 \cdot d_5^*)$$

$$v(\{3,4\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{3\}) + v(\{4\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.7 \cdot d_3^* + 0.5 \cdot d_4^*)$$

$$v(\{3,5\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{3\}) + v(\{5\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.7 \cdot d_3^* + 0.7 \cdot d_5^*)$$

$$v(\{4,5\}) = \left(1 + \frac{\gamma_2}{2}\right)(v(\{4\}) + v(\{5\})) = \left(1 + \frac{\gamma_2}{2}\right)(0.5 \cdot d_4^* + 0.7 \cdot d_5^*).$$

For $|S| = 3$, we have

$$v(\{1,2,3\}) = \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{2\}) + v(\{3\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^*)$$

$$v(\{1,2,4\}) = \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{2\}) + v(\{4\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^*)$$

$$\begin{aligned}
 v(\{1, 2, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{2\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_5^*) \\
 v(\{1, 3, 4\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{3\}) + v(\{4\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*) \\
 v(\{1, 3, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{3\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 v(\{1, 4, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{1\}) + v(\{4\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.5 \cdot d_1^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 v(\{2, 3, 4\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{2\}) + v(\{3\}) + v(\{4\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*) \\
 v(\{2, 3, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{2\}) + v(\{3\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 v(\{2, 4, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{2\}) + v(\{4\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 v(\{3, 4, 5\}) &= \left(1 + \frac{\gamma_3}{3}\right)(v(\{3\}) + v(\{4\}) + v(\{5\})) = \left(1 + \frac{\gamma_3}{3}\right)(0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*).
 \end{aligned}$$

For $|S| = 4$, we have

$$\begin{aligned}
 v(\{1, 2, 3, 4\}) &= \left(1 + \frac{\gamma_4}{4}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*) \\
 v(\{1, 2, 3, 5\}) &= \left(1 + \frac{\gamma_4}{4}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 v(\{1, 2, 4, 5\}) &= \left(1 + \frac{\gamma_4}{4}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 v(\{1, 3, 4, 5\}) &= \left(1 + \frac{\gamma_4}{4}\right)(0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 v(\{2, 3, 4, 5\}) &= \left(1 + \frac{\gamma_4}{4}\right)(0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*).
 \end{aligned}$$

Finally, for $|S| = 5$, we have

$$v(\{1, 2, 3, 4, 5\}) = \left(1 + \frac{\gamma_5}{5}\right)(0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*).$$

Under the above settings, the computational procedure is presented below.

- **Step 1 (Initialization).** The population size is taken to be $p = 20$. The initial population is given by $\gamma^{(k)} = (\gamma_1^{(k)}, \gamma_2^{(k)}, \gamma_3^{(k)}, \gamma_4^{(k)}, \gamma_5^{(k)})$ such that $\gamma_1^{(k)} = 0$ and $\gamma_s^{(k)}$ are random numbers in $[0, \mathcal{V}_5] = [0, 3.29449]$ for all $s = 2, 3, 4$ and $k = 1, \dots, 20$. Given each chromosome $\gamma^{(k)}$, we solve the linear programming problem in (11) given by

$$\begin{aligned}
 \text{(LP) minimize} \quad & w_1 + w_2 + w_3 + w_4 + w_5 \\
 \text{subject to} \quad & w_1 \geq v(\{1\}), w_2 \geq v(\{2\}), w_3 \geq v(\{3\}), w_1 + w_2 \geq v(\{1, 2\}) \\
 & w_1 + w_3 \geq v(\{1, 3\}), w_1 + w_4 \geq v(\{1, 4\}), w_1 + w_5 \geq v(\{1, 5\}) \\
 & w_2 + w_3 \geq v(\{2, 3\}), w_2 + w_4 \geq v(\{2, 4\}), w_2 + w_5 \geq v(\{2, 5\}) \\
 & w_3 + w_4 \geq v(\{3, 4\}), w_3 + w_5 \geq v(\{3, 5\}), w_4 + w_5 \geq v(\{4, 5\}) \\
 & w_1 + w_2 + w_3 \geq v(\{1, 2, 3\}), w_1 + w_2 + w_4 \geq v(\{1, 2, 4\}) \\
 & w_1 + w_2 + w_5 \geq v(\{1, 2, 5\}), w_1 + w_3 + w_4 \geq v(\{1, 3, 4\}) \\
 & w_1 + w_3 + w_5 \geq v(\{1, 3, 5\}), w_1 + w_4 + w_5 \geq v(\{1, 4, 5\}) \\
 & w_2 + w_3 + w_4 \geq v(\{2, 3, 4\}), w_2 + w_3 + w_5 \geq v(\{2, 3, 5\}) \\
 & w_2 + w_4 + w_5 \geq v(\{2, 4, 5\}), w_3 + w_4 + w_5 \geq v(\{3, 4, 5\}) \\
 & w_1 + w_2 + w_3 + w_4 \geq v(\{1, 2, 3, 4\}), w_1 + w_2 + w_3 + w_5 \geq v(\{1, 2, 3, 5\}) \\
 & w_1 + w_2 + w_4 + w_5 \geq v(\{1, 2, 4, 5\}), w_1 + w_3 + w_4 + w_5 \geq v(\{1, 3, 4, 5\}) \\
 & w_2 + w_3 + w_4 + w_5 \geq v(\{2, 3, 4, 5\}) \\
 & w_1 + w_2 + w_3 + w_4 + w_5 \geq v(\{1, 2, 3, 4, 5\}) \\
 & w_1 \geq 0, w_2 \geq 0, w_3 \geq 0, w_4 \geq 0, w_5 \geq 0.
 \end{aligned}$$

More precisely, we are going to solve the following linear programming problem

$$\begin{aligned}
 \text{(LP) minimize} \quad & w_1 + w_2 + w_3 + w_4 + w_5 \\
 \text{subject to} \quad & w_1 \geq 0.5 \cdot d_1^*, w_2 \geq 0.6 \cdot d_2^*, w_3 \geq 0.7 \cdot d_3^*, w_4 \geq 0.5 \cdot d_4^*, w_5 \geq 0.7 \cdot d_5^* \\
 & C_{12}, C_{13}, C_{14}, C_{15}, C_{23}, C_{24}, C_{25}, C_{34}, C_{35}, C_{45} \\
 & C_{123}, C_{124}, C_{125}, C_{134}, C_{135}, C_{145}, C_{234}, C_{235}, C_{245}, C_{345} \\
 & C_{1234}, C_{1235}, C_{1245}, C_{1345}, C_{2345}, C_{12345} \\
 & w_1 \geq 0, w_2 \geq 0, w_3 \geq 0, w_4 \geq 0, w_5 \geq 0,
 \end{aligned}$$

where the detailed constraints are given below

$$C_{12} : w_1 + w_2 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^*)$$

$$C_{13} : w_1 + w_3 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.5 \cdot d_1^* + 0.7 \cdot d_3^*)$$

$$C_{14} : w_1 + w_4 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.5 \cdot d_1^* + 0.5 \cdot d_4^*)$$

$$C_{15} : w_1 + w_5 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.5 \cdot d_1^* + 0.7 \cdot d_5^*)$$

$$C_{23} : w_2 + w_3 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.6 \cdot d_2^* + 0.7 \cdot d_3^*)$$

$$C_{24} : w_2 + w_4 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.6 \cdot d_2^* + 0.5 \cdot d_4^*)$$

$$C_{25} : w_2 + w_5 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.6 \cdot d_2^* + 0.7 \cdot d_5^*)$$

$$C_{34} : w_3 + w_4 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.7 \cdot d_3^* + 0.5 \cdot d_4^*)$$

$$C_{35} : w_3 + w_5 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.7 \cdot d_3^* + 0.7 \cdot d_5^*)$$

$$C_{45} : w_4 + w_5 \geq \left(1 + \frac{\gamma_2^{(k)}}{2}\right) (0.5 \cdot d_4^* + 0.7 \cdot d_5^*)$$

$$C_{123} : w_1 + w_2 + w_3 \geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^*)$$

$$C_{124} : w_1 + w_2 + w_4 \geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^*)$$

$$C_{125} : w_1 + w_2 + w_5 \geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_5^*)$$

$$C_{134} : w_1 + w_3 + w_4 \geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*)$$

$$\begin{aligned}
 C_{135} : w_1 + w_3 + w_5 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 C_{145} : w_1 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.5 \cdot d_1^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 C_{234} : w_2 + w_3 + w_4 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*) \\
 C_{235} : w_2 + w_3 + w_5 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 C_{245} : w_2 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 C_{345} : w_3 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_3^{(k)}}{3}\right) (0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 \\
 C_{1234} : w_1 + w_2 + w_3 + w_4 &\geq \left(1 + \frac{\gamma_4^{(k)}}{4}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^*) \\
 C_{1235} : w_1 + w_2 + w_3 + w_5 &\geq \left(1 + \frac{\gamma_4^{(k)}}{4}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.7 \cdot d_5^*) \\
 C_{1245} : w_1 + w_2 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_4^{(k)}}{4}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 C_{1345} : w_1 + w_3 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_4^{(k)}}{4}\right) (0.5 \cdot d_1^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 C_{2345} : w_2 + w_3 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_4^{(k)}}{4}\right) (0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*) \\
 \\
 C_{12345} : w_1 + w_2 + w_3 + w_4 + w_5 &\geq \left(1 + \frac{\gamma_5^{(k)}}{5}\right) (0.5 \cdot d_1^* + 0.6 \cdot d_2^* + 0.7 \cdot d_3^* + 0.5 \cdot d_4^* + 0.7 \cdot d_5^*).
 \end{aligned}$$

Proposition 1 says that the optimal solution $\mathbf{w}^*(\gamma^{(k)})$ of problem (LP) is a positive core for $k = 1, \dots, 20$. According to (13), we also calculate the normalized positive core as follows

$$\bar{w}_i^*(\gamma^{(k)}) = \frac{w_i^*(\gamma^{(k)})}{w_1^*(\gamma^{(k)}) + w_2^*(\gamma^{(k)}) + w_3^*(\gamma^{(k)}) + w_3^*(\gamma^{(k)}) + w_4^*(\gamma^{(k)})}$$

for $i = 1, 2, 3, 4, 5$ and $k = 1, \dots, 20$. Each chromosome $\gamma^{(k)}$ is assigned a fitness value given by

$$\begin{aligned}
 \eta(\gamma^{(k)}) &= \bar{w}_1^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_1}^L(\mathbf{x}^*(\gamma^{(k)})) + \bar{w}_2^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_2}^L(\mathbf{x}^*(\gamma^{(k)})) \\
 &\quad + \bar{w}_3^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_3}^L(\mathbf{x}^*(\gamma^{(k)})) + \bar{w}_4^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_1}^U(\mathbf{x}^*(\gamma^{(k)})) \\
 &\quad + \bar{w}_5^*(\gamma^{(k)}) \cdot \tilde{H}_{\alpha_2}^U(\mathbf{x}^*(\gamma^{(k)})).
 \end{aligned}$$

for $k = 1, \dots, 20$. We rank the 20 chromosomes $\gamma^{(k)}$ for $k = 1, \dots, 20$ in descending order of their corresponding fitness values $\eta(\gamma^{(k)})$ for $k = 1, \dots, 20$. The top one is saved to be the (initial) best chromosome named as $\bar{\eta}_0$. Save $\gamma^{(k)}$ as an elite $\gamma^{(*k)}$ given by $\gamma_s^{(*k)} \leftarrow \gamma_s^{(k)}$ for $s = 1, \dots, 5$ and $j = 1, \dots, p$.

- **Step 2 (Tolerance).** Set $t = 0, m^* = 20, l^* = 1$ and the tolerance $\epsilon = 10^{-6}$.
- **Step 3 (Mutation).** We set $t \leftarrow t + 1$ to mean the t -th generation. For $k = 1, \dots, 20$, each chromosome

$$\gamma^{(k)} = (0, \gamma_2^{(k)}, \gamma_3^{(k)}, \gamma_4^{(k)}, \gamma_5^{(k)})$$

is mutated, and is assigned to

$$\gamma^{(k+20)} = (0, \gamma_2^{(k+20)}, \gamma_3^{(k+20)}, \gamma_4^{(k+20)}, \gamma_5^{(k+20)})$$

by using (15). Generate the random Gaussian numbers with mean zero and standard deviation σ_s , where σ_s is taken to be the following form

$$\sigma_s = \beta_s \cdot \eta(\gamma^{(k)}) + b_s \text{ for } s = 2, 3, 4, 5.$$

The constant β_s is the proportionality to scale $\eta(\gamma^{(k)})$ and the constant b_s represents the offset. In this example, we take

$$\beta_2 = 0.02, \quad \beta_3 = 0.01, \quad \beta_4 = 0.02, \quad \beta_5 = 0.01$$

and $b_s = 0$ for $s = 2, 3, 4, 5$. Then, we assign

$$\hat{\gamma}_s^{(k)} = \gamma_s^{(k)} + \sigma_s \cdot N(0, 1) = \gamma_s^{(k)} + (\beta_s \cdot \eta(\gamma^{(k)}) + b_s) \cdot N(0, 1).$$

Therefore, we obtain the mutated chromosome $\gamma^{(k+p)}$ with components given by $\gamma_1^{(k+p)} = 0$ and

$$\gamma_s^{(k+20)} = \begin{cases} \hat{\gamma}_s^{(k)} & \text{if } \hat{\gamma}_s^{(k)} \in [0, \mathcal{V}_s] \\ \mathcal{V}_s & \text{if } \hat{\gamma}_s^{(k)} > \mathcal{V}_s \\ 0 & \text{if } \hat{\gamma}_s^{(k)} < 0 \end{cases}$$

for $s = 2, 3, 4, 5$ and $k = 1, \dots, 20$. After this step, we shall have 40 chromosomes $\gamma^{(k)}$ for $k = 1, \dots, 40$.

- **Step 4 (Crossover).** We randomly select

$$\gamma^{(k_1)} = (0, \gamma_2^{(k_1)}, \gamma_3^{(k_1)}, \gamma_4^{(k_1)}, \gamma_5^{(k_1)}) \text{ and } \gamma^{(k_2)} = (0, \gamma_2^{(k_2)}, \gamma_3^{(k_2)}, \gamma_4^{(k_2)}, \gamma_5^{(k_2)})$$

for $k_1, k_2 \in \{1, \dots, 40\}$ with $k_1 \neq k_2$. We generate a random number $\lambda \in (0, 1)$, the new chromosome is given by

$$\gamma^{(41)} = \lambda \gamma^{(k_1)} + (1 - \lambda) \gamma^{(k_2)}$$

with components

$$\gamma_s^{(41)} = \lambda \gamma_s^{(k_1)} + (1 - \lambda) \gamma_s^{(k_2)} \text{ for } s = 2, 3, 4, 5.$$

After this step, we shall have 41 chromosomes $\gamma^{(k)}$ for $j = 1, \dots, 41$.

- **Step 5 (Calculate New Fitness).** For the new generated chromosomes

$$\gamma^{(k+20)} = (0, \gamma_2^{(k+20)}, \gamma_3^{(k+20)}, \gamma_4^{(k+20)}, \gamma_5^{(k+20)})$$

for $k = 1, \dots, 20$, using Proposition 1 and (13), we calculate the normalized positive value $\bar{w}_i^*(\gamma^{(k+20)})$ by solving the linear programming problem in (11) for $i = 1, 2, 3, 4, 5$ and $k = 1, \dots, 20 + 1$. Each chromosome $\gamma^{(k+20)}$ is assigned a fitness value given by

$$\begin{aligned} \eta(\gamma^{(k+20)}) &= \bar{w}_1^*(\gamma^{(k+20)}) \cdot \tilde{H}_{\alpha_1}^L(\mathbf{x}^*(\gamma^{(k+20)})) + \bar{w}_2^*(\gamma^{(k+20)}) \cdot \tilde{H}_{\alpha_2}^L(\mathbf{x}^*(\gamma^{(k+20)})) \\ &+ \bar{w}_3^*(\gamma^{(k+20)}) \cdot \tilde{H}_{\alpha_3}^L(\mathbf{x}^*(\gamma^{(k+20)})) + \bar{w}_4^*(\gamma^{(k+20)}) \cdot \tilde{H}_{\alpha_1}^U(\mathbf{x}^*(\gamma^{(k+20)})) \\ &+ \bar{w}_5^*(\gamma^{(k+20)}) \cdot \tilde{H}_{\alpha_2}^U(\mathbf{x}^*(\gamma^{(k+20)})). \end{aligned}$$

for $k = 1, \dots, 20$.

- **Step 6 (Selection).** We rank the 20 old elites

$$\gamma^{(*k)} = (0, \gamma_2^{(*k)}, \gamma_3^{(*k)}, \gamma_4^{(*k)}, \gamma_5^{(*k)}) \text{ for } k = 1, \dots, 20$$

and the 41 new chromosomes

$$\gamma^{(k)} = (0, \gamma_2^{(k)}, \gamma_3^{(k)}, \gamma_4^{(k)}, \gamma_5^{(k)}) \text{ for } k = 1, \dots, 41$$

obtained from Steps 3 and 4 in descending order of their corresponding fitness values $\eta(\gamma^{(*k)})$ and $\eta(\gamma^{(k)})$. The top 20 chromosomes are saved to be the new elites

$$\gamma^{(*k)} = (0, \gamma_2^{(*k)}, \gamma_3^{(*k)}, \gamma_4^{(*k)}, \gamma_5^{(*k)}) \text{ for } k = 1, \dots, 20.$$

Also, the top one is saved to be the best chromosome named as $\bar{\eta}_t$ for the t -th generation

- **Step 7 (Stopping Criterion).** After step 6, we may obtain $\bar{\eta}_t = \bar{\eta}_{t-1}$, which seems to be trapped in the local optimum. In order to escape this trap, we proceed more iterations for $m^* = 20$ times even though the criterion $\|\bar{\eta}_t - \bar{\eta}_{t-1}\| < \epsilon$ is satisfied. When the criterion $\|\bar{\eta}_t - \bar{\eta}_{t-1}\| < \epsilon$ is satisfied and the iterations reach 20 times, we stop the algorithm and return the solution for phase I. Otherwise, the new elites

$$\gamma^{(*k)} = (0, \gamma_2^{(*k)}, \gamma_3^{(*k)}, \gamma_4^{(*k)}, \gamma_5^{(*k)}) \text{ for } k = 1, \dots, 20$$

must be copied to be the next generation

$$\gamma^{(k)} = (0, \gamma_2^{(k)}, \gamma_3^{(k)}, \gamma_4^{(k)}, \gamma_5^{(k)}) \text{ for } k = 1, \dots, 20.$$

We set $l^* \leftarrow l^* + 1$ and the algorithm proceeds to step 3. Note that the number l^* counts the times for satisfying the tolerance $\|\bar{\eta}_t - \bar{\eta}_{t-1}\| < \epsilon$.

The computer code is implemented using Microsoft Excel VBA. The best fitness value is 119.6879088 and the approximated core-nondominated solution is $\mathbf{x}^*(\Lambda) = (x_1, x_2, x_3) = (0, 15, 3)$.

Now, we consider more finer partitions of $[0, 1]$ according to the suggestion of phase II.

- **Step 1.** From Section 6.2, we consider a new finer partition by equally divide the unit interval $[0, 1]$. Therefore, we take

$$\bar{\Lambda} = \{\alpha_1 = 0, \alpha_2 = 0.25, \alpha_3 = 0.5, \alpha_4 = 0.75, \alpha_5 = 1\}.$$

It is clear to see $\Lambda \subset \bar{\Lambda}$.

- **Step 2.** Using this new finer partition $\bar{\Lambda}$ from Step 1 and the genetic algorithm in phase I, a new approximated best core-nondominated solution $\mathbf{x}^*(\bar{\Lambda}) = (0, 15, 3)$ can be obtained.
- **Step 3.** Step 2 says $\mathbf{x}^*(\Lambda) = \mathbf{x}^*(\bar{\Lambda}) = (0, 15, 3)$. Therefore, the final best core-nondominated solution is given by $(x_1, x_2, x_3) = (0, 15, 3)$.

Finally, using Theorem 1, we obtain the approximated nondominated solution $(x_1, x_2, x_3) = (0, 15, 3)$ of the original fuzzy linear programming problem.

8. Conclusions

This paper proposes a new methodology by incorporating the core values of cooperative games and the genetic algorithms to solve the fuzzy multiobjective optimization problem, which is a new attempt for solving this kind of problem. Usually, the fuzzy multiobjective optimization problem can be transformed into a conventional single-objective optimization problem such that the suitable weights are determined by the decision makers. In order to avoid the possible biased assignment of weights, a mechanical procedure is proposed in this paper by assigning the core values of cooperative game as the weights of this conventional single-objective optimization problem.

The purpose is to use the popular numerical methods of optimization to solve this conventional single-objective optimization problem. For example, for solving the fuzzy multiobjective linear programming problem, this original problem can be transformed into a conventional single-objective linear programming problem. In this case, the simplex method can be used to solve the desired problem. Frequently, the core-nondominated solutions can be a large set. Therefore, the genetic algorithms is adopted to obtain the best core-nondominated solution from this large set of core-nondominated solutions. This paper does not intend to use the genetic algorithm to directly solve the fuzzy multiobjective optimization problems. The genetic algorithm used in this paper is just to obtain the best core-nondominated solution from a large set of core-nondominated solutions. The monograph by Sakawa [44] provides the method for using the genetic algorithms to directly solve the fuzzy multiobjective optimization problems. Although the genetic algorithms is adopted in this paper to obtain the best core-nondominated solution, some other heuristic algorithms like Particle Swarm Optimization, Scatter Search, Tabu Search, Ant Colony Optimization, Artificial Immune Systems, and Simulated Annealing can still be used to obtain the best core-nondominated solutions.

Although the core values of cooperative games are considered in this paper, many other solution concepts of cooperative games can also be adopted to set up conventional single-objective optimization problem, which can be the future research. The theory of non-cooperative games may be another way to set up the conventional single-objective optimization problem, which can be the future research.

Funding: This research was funded by NSTC Taiwan with grant number 110-2221-E-017-008.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflict of interest.

References

- Bellman, R.E.; Zadeh, L.A. Decision-Making in a Fuzzy Environment. *Manag. Sci.* **1970**, *17*, 141–164. [\[CrossRef\]](#)
- Tanaka, H.; Okuda, T.; Asai, K. On Fuzzy-Mathematical Programming. *J. Cybern.* **1974**, *3*, 37–46. [\[CrossRef\]](#)
- Zimmermann, H.-J. Description and Optimization of Fuzzy Systems. *Int. J. Gen. Syst.* **1976**, *2*, 209–215. [\[CrossRef\]](#)
- Zimmermann, H.-J. Fuzzy Programming and Linear Programming with Several Objective Functions. *Fuzzy Sets Syst.* **1978**, *1*, 45–55. [\[CrossRef\]](#)
- Herrera, F.; Kovács, M.; Verdegay, J.L. Optimality for fuzzified mathematical programming problems: A parametric approach. *Fuzzy Sets Syst.* **1993**, *54*, 279–285. [\[CrossRef\]](#)
- Buckley, J.J. Joint Solution to Fuzzy Programming Problems. *Fuzzy Sets Syst.* **1995**, *72*, 215–220. [\[CrossRef\]](#)
- Julien, B. An extension to possibilistic linear programming. *Fuzzy Sets Syst.* **1994**, *64*, 195–206. [\[CrossRef\]](#)
- Luhandjula, M.K.; Ichihashi, H.; Inuiguchi, M. Fuzzy and semi-infinite mathematical programming. *Inf. Sci.* **1992**, *61*, 233–250. [\[CrossRef\]](#)
- Inuiguchi, M. Necessity Measure Optimization in Linear Programming Problems with Fuzzy Polytopes. *Fuzzy Sets Syst.* **2007**, *158*, 1882–1891. [\[CrossRef\]](#)
- Wu, H.-C. Duality Theory in Fuzzy Optimization Problems. *Fuzzy Optim. Decis. Mak.* **2004**, *3*, 345–365. [\[CrossRef\]](#)
- Wu, H.-C. Duality Theorems and Saddle Point Optimality Conditions in Fuzzy Nonlinear Programming Problems Based on Different Solution Concepts. *Fuzzy Sets Syst.* **2007**, *158*, 1588–1607. [\[CrossRef\]](#)

12. Wu, H.-C. The Optimality Conditions for Optimization Problems with Fuzzy-Valued Objective Functions. *Optimization* **2008**, *57*, 473–489. [[CrossRef](#)]
13. Wu, H.-C. The Karush-Kuhn-Tucker Optimality Conditions for Multi-objective Programming Problems with Fuzzy-Valued Objective Functions. *Fuzzy Optim. Decis. Mak.* **2009**, *8*, 1–28. [[CrossRef](#)]
14. Chalco-Cano, Y.; Lodwick, W.A.; Osuna-Gómez, R.; Rufian-Lizana, A. The Karush-Kuhn-Tucker Optimality Conditions for Fuzzy Optimization Problems. *Fuzzy Optim. Decis. Mak.* **2016**, *15*, 57–73. [[CrossRef](#)]
15. Li, L.; Liu, S.; Zhang, J. On fuzzy generalized convex mappings and optimality conditions for fuzzy weakly univex mappings. *Fuzzy Sets Syst.* **2015**, *280*, 107–132. [[CrossRef](#)]
16. Chalco-Cano, Y.; Silva, G.N.; Rufian-Lizana, A. On the Newton method for solving fuzzy optimization problems. *Fuzzy Sets Syst.* **2015**, *272*, 60–69. [[CrossRef](#)]
17. Pirzada, U.M.; Pathak, V.D. Newton method for solving the multi-variable fuzzy optimization problem. *J. Optim. Theorey Appl.* **2013**, *156*, 867–881. [[CrossRef](#)]
18. Luhandjula, M.K. An Interval Approximation Approach for a Multiobjective Programming Model with Fuzzy Objective Functions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2015**, *23*, 845–860. [[CrossRef](#)]
19. Yano, H. Multiobjective Programming Problems Involving Fuzzy Coefficients, Random Variable Coefficients and Fuzzy Random Variable Coefficients. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2015**, *23*, 483–504. [[CrossRef](#)]
20. Ebenuwa, A.U.; Tee, K.F.; Zhang, Y. Fuzzy-Based Multi-Objective Design Optimization of Buried Pipelines. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2021**, *29*, 209–229. [[CrossRef](#)]
21. Charles, V.; Gupta, S.; Ali, I. A Fuzzy Goal Programming Approach for Solving Multi-Objective Supply Chain Network Problems with Pareto-Distributed Random Variables. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2019**, *27*, 559–593. [[CrossRef](#)]
22. Roy, J.; Majumder, S.; Kar, S.; Adhikary, K. A Multiobjective Multi-Product Solid Transportation Model with Rough Fuzzy Coefficients. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2019**, *27*, 719–753. [[CrossRef](#)]
23. von Neumann, J.; Morgenstern, O. *Theory of Games and Economic Behavior*; Princeton University Press: Princeton, NJ, USA, 1944.
24. Nash, J.F. Two-Person Cooperative Games. *Econometrica* **1953**, *21*, 128–140. [[CrossRef](#)]
25. Young, H.P. Monotonic Solutions of Cooperative Games. *Int. J. Game Theory* **1985**, *14*, 65–72. [[CrossRef](#)]
26. Barron, E.N. *Game Theory: An Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
27. Branzei, R.; Dimitrov, D.; Tijs, S. *Models in Cooperative Game Theory*; Springer: Berlin, Germany, 2008.
28. Curiel, I. *Cooperative Game Theory and Applications: Cooperative Games Arising from Combinatorial Optimization Problems*; Kluwer Academic Publishers: New York, NY, USA, 1997.
29. González-Díaz, J.; García-Jurado, I.; Fiestras-Janeiro, M.G. *An Introductory Course on Mathematical Game Theory*; American Mathematical Society: Providence, RI, USA, 2010.
30. Owen, G. *Game Theory*, 3rd ed.; Academic Press: New York, NY, USA, 1995.
31. Yu, X.; Zhang, Q. Core for Game with Fuzzy Generalized Triangular Payoff Value. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2019**, *27*, 789–813. [[CrossRef](#)]
32. Jing, R.; Wang, M.; Liang, H.; Wang, X.; Li, N.; Shah, N.; Zhao, Y. Multi-objective optimization of a neighborhood-level urban energy network: Considering Game-theory inspired multi-benefit allocation constraints. *Appl. Energy* **2018**, *231*, 534–548 [[CrossRef](#)]
33. Lokeshgupta, B.; Sivasubramani, S. Cooperative game theory approach for multiobjective home energy management with renewable energy integration. *IET Smart Grid* **2019**, *2*, 34–41. [[CrossRef](#)]
34. Lee, C.-S. Multi-objective game-theory models for conflict analysis in reservoir watershed management. *Chemosphere* **2012**, *87*, 608–613. [[CrossRef](#)]
35. Meng, R.; Xie, N.-G. A Competitive-Cooperative Game Method for Multi-Objective Optimization Design of a Horizontal Axis Wind Turbine Blade. *IEEE Access* **2019**, *7*, 155748–155758. [[CrossRef](#)]
36. Li, X.; Gao, L.; Li, W. Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Syst. Appl.* **2012**, *39*, 288–297. [[CrossRef](#)]
37. Yu, Y.; Zhao, R.; Zhang, J.; Yang, D.; Zhou, T. Multi-objective game theory optimization for balancing economic, social and ecological benefits in the Three Gorges Reservoir operation. *Environ. Res. Lett.* **2021**, *16*, 085007. [[CrossRef](#)]
38. Zhang, Y.; Wang, J.; Liu, Y. Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. *J. Clean. Prod.* **2017**, *167*, 665–679. [[CrossRef](#)]
39. Chai, R.; Savvarisa, A.; Tsourdosa, A.; Chai, S. Multi-objective trajectory optimization of Space Manoeuvre Vehicle using adaptive differential evolution and modified game theory. *Acta Astronaut.* **2017**, *136*, 273–280. [[CrossRef](#)]
40. Cao, R.; Coit, D.W.; Hou, W.; Yang, Y. Game theory based solution selection for multi-objective redundancy allocation in interval-valued problem parameters. *Reliab. Eng. Syst. Saf.* **2020**, *199*, 106932. [[CrossRef](#)]
41. Deb, K. *Multiobjective Optimization Using Evolutionary Algorithms*; John Wiley & Sons: New York, NY, USA, 2001.
42. Osyczka, A. *Evolutionary Algorithms for Single and Multicriteria Design Optimization*; Studies in Fuzziness and Soft Computing 100; Physica: New York, NY, USA, 2002.
43. Tan, K.C.; Khor, E.F.; Lee, T.H. *Multiobjective Evolutionary Algorithms and Applications*; Springer: New York, NY, USA, 2005.

44. Sakawa, M. *Genetic Algorithms and Fuzzy Multiobjective Optimization*; Kluwer Academic Publishers: New York, NY, USA, 2002.
45. Tiwari, V.L.; Thapar, A.; Bansal, R. A Genetic Algorithm for Solving Nonlinear Optimization Problem with Max-Archimedean Bipolar Fuzzy Relation Equations. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2023**, *31*, 303–326. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.