



A deterministic and nature-inspired algorithm for the fuzzy multi-objective path optimization problem

Yi-Ming Ma¹ · Xiao-Bing Hu^{2,3,4} · Hang Zhou¹

Received: 24 March 2022 / Accepted: 7 July 2022
© The Author(s) 2022

Abstract

Increasing evaluation indexes have been involved in the network modeling, and some parameters cannot be described precisely. Fuzzy set theory becomes a promising mathematical method to characterize such uncertain parameters. This study investigates the fuzzy multi-objective path optimization problem (FMOPOP), in which each arc has multiple crisp and fuzzy weights simultaneously. Fuzzy weights are characterized by triangular fuzzy numbers or trapezoidal fuzzy numbers. We adopt two fuzzy number ranking methods based on their fuzzy graded mean values and distances from the fuzzy minimum number. Motivated by the ripple spreading patterns on the natural water surface, we propose a novel ripple-spreading algorithm (RSA) to solve the FMOPOP. Theoretical analyses prove that the RSA can find all Pareto optimal paths from the source node to all other nodes within a single run. Numerical examples and comparative experiments demonstrate the efficiency and robustness of the newly proposed RSA. Moreover, in the first numerical example, the processes of the RSA are illustrated using metaphor-based language and ripple spreading phenomena to be more comprehensible. To the best of our knowledge, the RSA is the first algorithm for the FMOPOP that can adopt various fuzzy numbers and ranking methods while maintaining optimality.

Keywords Fuzzy graph · Multi-objective optimization · Path optimization · Pareto optimal path · Ripple-spreading algorithm

Introduction

The path optimization problem (POP) is fundamental to various practical applications, such as transportation [1], supply chain management [2], telecommunication [3], and robot control [4]. The shortest path problem (SPP) is the most well-studied in the field of POP, and many classical algo-

rithms have been proposed [5,6]. The SPP aims to minimize the summation of deterministic weights associated with each arc. With the fast development of the network modeling, each arc may have multiple criteria [7], which forms the multi-objective path optimization problem (MOPOP).

Generally, the weights of each arc represent the cost, time, and reliability that are assumed to be deterministic. However, in many real-world situations, some kinds of weights are uncertain and difficult to be represented by crisp numbers. For example, the traversing time of a road greatly depends on traffic conditions [8]. Zadeh et al. [9] proposed the fuzzy set theory, which is a typical way to handle uncertainties. An element of a universe discourse is characterized by a membership degree between 0 and 1 [10]. The fuzzy set theory has emerged as a powerful platform for various real-world applications with uncertainties, such as decision making [11], network modeling [12], and pattern recognition [13]. The POP on fuzzy networks has been widely studied in recent years on account of its application values. Moreover, a network may have both crisp and fuzzy weights simultaneously. For example, path length and population exposure are two evaluation indexes for hazardous material transportation [14]. The path length can be measured accurately, while

✉ Xiao-Bing Hu
huxbtg@163.com

Yi-Ming Ma
xavier_mayiming@163.com

Hang Zhou
enzo-zhouhang@hotmail.com

¹ Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Tianjin 300300, China

² Laboratory of Complex System Safety and Intelligent Decisions, College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China

³ Collaborative Innovation Center of eTourism, Beijing Union University, Beijing 100101, China

⁴ School of Engineering, University of Warwick, Coventry CV4 7AL, UK

the population exposure depends on the population density, which is always time-varying. Therefore, providing complete solutions efficiently to the MOPOP with crisp and fuzzy weights may make a difference in numerous real-world problems.

The MOPOP belongs to the scope of the multi-objective optimization problem (MOOP), whose primary goal is to find Pareto optimal solutions [15]. There exist mainly four categories of MOOP algorithms: (1) using an aggregate function to combine objectives into a single one [15,16]; (2) optimizing one objective while treating the others as constraints [17]; (3) evolutionary algorithms based on Pareto ranking [18]; (4) evolutionary algorithms based on decomposition [19]. The first two categories decompose the MOOP into single-objective optimization problems that deem partial objectives more important than the others. Therefore, they may miss some Pareto optimal solutions [20]. The last two categories are based on evolutionary algorithms, which are stochastic and cannot guarantee optimality [21]. In addition, some specific algorithms for MOPOPs have been proposed, such as the two-phase method [22], labeling method [23], and ranking method [24]. These algorithms are logic-based and extended from SPP algorithms. Moreover, they mainly focus on bi-objective problems. For this reason, these MOPOP algorithms have relatively low flexibility to be adapted to fuzzy graphs with more than two objectives.

Recently, researchers have shown a growing interest in POPs with fuzzy weights. Dubois et al. [25] first proposed an algorithm based on the Floyd algorithm to solve the SPP in a fuzzy network (FSPP), in which each arc is characterized by a fuzzy number. Various algorithms for the FSPP have been proposed [26], such as the dynamic programming [27], fuzzy Dijkstra algorithm [28], fuzzy Physarum algorithm [29], and fuzzy genetic algorithm [30]. Furthermore, some researchers also investigate MOPOPs with fuzzy weights. Kim et al. [31] proposed a quantum-inspired evolutionary algorithm to plan robotic paths considering three objectives. In [32], fuzzy objectives are aggregated using sum-weighted functions. Adhikari et al. [33] planned 3D UAV paths with a multi-objective fuzzy logic controller using fuzzy adaptive differential evolution. Zero et al. [14] modified Martin's Algorithm and A* algorithm to solve a bi-objective POP with a crisp objective and a min-max fuzzy objective. Abbaszadeh et al. [34] implemented the fuzzy inference system to find a path considering three objectives while respecting resource constraints. Bagheri et al. [35] devised fuzzy efficiency scores to convert fuzzy objectives to a single objective and implemented the data envelopment analysis.

The ripple-spreading algorithm (RSA) is an agent-based and deterministic algorithm that consists in solving various POPs [36–39]. The RSA is proposed by simulating the following natural phenomena: a ripple spreads out on a disturbed water surface and triggers new ripples when encountering

obstacles. Different RSAs share the same optimization principle: the traveling path of the first ripple reaching a node is the shortest path, which is consistent with natural phenomena. Although this principle makes the RSA a deterministic algorithm, the RSA can also be implemented as an agent-based model whose primary merit is flexibility [40]. All ripples can have their spreading motions and triggering conditions, and their combined behaviors help solve a specific problem. We can modify the triggering condition and termination judgment of the RSA to make it solve the SPP [36], the POP on dynamic networks [37], the k shortest paths problem [38], and the MOPOP [39].

The primary goal of this paper is to provide complete solutions efficiently to MOPOPs with multiple crisp and fuzzy weights. We define the fuzzy MOPOP (FMOPOP), in which each arc has at least one crisp weight and multiple fuzzy weights. Each fuzzy weight can be represented by a triangular fuzzy number (TFN) [41] or a trapezoidal fuzzy number (TrFN) [42]. Moreover, we adopt two methods to rank fuzzy numbers. The first method is based on fuzzy graded mean values [28,30], and the second method is based on the distances between fuzzy numbers [27,29,43]. Motivated by the RSA, we adapt the triggering condition to make it solve the FMOPOP. A ripple triggers another ripple on a water surface when reaching a stone. In the RSA, we define the Pareto optimal ripple (POR), and only PORs can trigger new ripples at nodes. The newly proposed RSA is a deterministic algorithm that ensures optimality. Furthermore, the RSA can determine all Pareto optimal paths from the source node to all other nodes within a single run. Different kinds of fuzzy numbers and ranking methods can be adopted using the RSA while maintaining optimality.

The remainder of this paper is organized as follows. The second section provides preliminaries about the fuzzy set theory, Pareto concepts, and the RSA. The mathematical formulation of the FMOPOP is provided in the third section. In the fourth section, the RSA is adapted to solve the FMOPOP. Several numerical examples and comparative experiments are presented in the fifth section. Finally, the paper ends up with some conclusions in the last sections.

Preliminaries

Basic definitions

Definition 1 (Fuzzy set) Let X denote a non-empty set, and \tilde{A} denote a subset of X . If \tilde{A} is a fuzzy set, it can be described as:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}, \quad (1)$$

where $\mu_{\tilde{A}} : X \rightarrow [0, 1]$ is the membership function of \tilde{A} . For $\forall x \in X$, $\mu_{\tilde{A}}(x)$ denotes the membership degree of x in \tilde{A} .

Definition 2 (Fuzzy number) If and only if a fuzzy set \tilde{A} is convex and normal, it is a fuzzy number. Where "convex" denotes that:

$$\mu_{\tilde{A}}((1 - \lambda)x_1 + \lambda x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)), \quad \forall x_1, x_2 \in X, \forall \lambda \in [0, 1], \tag{2}$$

and "normal" denotes that:

$$\sup_{x \in X} \mu_{\tilde{A}}(x) = 1. \tag{3}$$

Definition 3 (Triangular fuzzy number) Let $\tilde{A} = (a_1, a_2, a_3)$ be a triangular fuzzy number (TFN), whose membership function is defined as follows:

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2, \\ \frac{a_3 - x}{a_3 - a_2}, & a_2 \leq x \leq a_3, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Definition 4 (Trapezoidal fuzzy number) Let $\tilde{A} = (a_1, a_2, a_3, a_4)$ be a trapezoidal fuzzy number (TrFN), whose membership function is defined as follows:

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2, \\ 1, & a_2 \leq x \leq a_3, \\ \frac{a_4 - x}{a_4 - a_3}, & a_3 \leq x \leq a_4, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Definition 5 (Fuzzy operations) For two TFNs $\tilde{A} = (a_1, a_2, a_3)$ and $\tilde{B} = (b_1, b_2, b_3)$, their fuzzy addition (\oplus) and multiplication (\otimes) operations are defined as:

$$\begin{aligned} \tilde{A} \oplus \tilde{B} &= (a_1 + b_1, a_2 + b_2, a_3 + b_3), \\ \tilde{A} \otimes \tilde{B} &= (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3). \end{aligned} \tag{6}$$

And for two TrFNs $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$, their fuzzy addition (\oplus) and multiplication (\otimes) operations are defined as:

$$\begin{aligned} \tilde{A} \oplus \tilde{B} &= (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4), \\ \tilde{A} \otimes \tilde{B} &= (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3, a_4 \times b_4). \end{aligned} \tag{7}$$

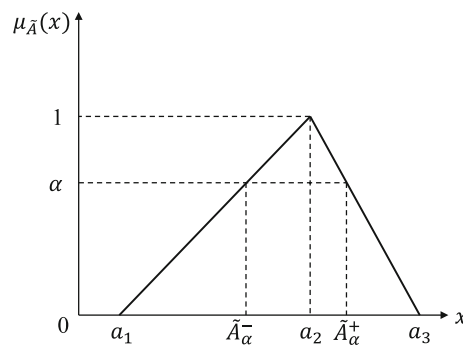


Fig. 1 A TFN $\tilde{A} = (a_1, a_2, a_3)$ and its α -cut

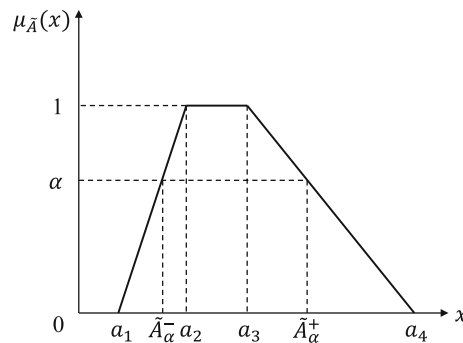


Fig. 2 A TrFN $\tilde{A} = (a_1, a_2, a_3, a_4)$ and its α -cut

Definition 6 (α -cut) \tilde{A} is a fuzzy subset of X . The α -cut (\tilde{A}_α) and strong α -cut ($\tilde{A}_{\alpha+}$) of \tilde{A} are defined as:

$$\begin{aligned} \tilde{A}_\alpha &= \{x \mid \mu_{\tilde{A}}(x) \geq \alpha, x \in X\}, \\ \tilde{A}_{\alpha+} &= \{x \mid \mu_{\tilde{A}}(x) > \alpha, x \in X\}, \end{aligned} \tag{8}$$

where $\alpha \in [0, 1]$. Moreover, we use \tilde{A}_α^+ to denote $\sup(\tilde{A}_\alpha)$ and \tilde{A}_α^- to denote $\inf(\tilde{A}_\alpha)$.

We provide Figs. 1 and 2 to illustrate an example of TFN and TrFN and their α -cut. From the two figures, we can quickly determine \tilde{A}_α^+ and \tilde{A}_α^- .

Definition 7 (Fuzzy graded mean value) For a TFN $\tilde{A} = (a_1, a_2, a_3)$, its fuzzy graded mean value is defined as:

$$P(\tilde{A}) = \frac{1}{6}(a_1 + 4 \times a_2 + a_3). \tag{9}$$

And for a TrFN $\tilde{A} = (a_1, a_2, a_3, a_4)$, its fuzzy graded mean value is defined as:

$$P(\tilde{A}) = \frac{1}{6}(a_1 + 2 \times a_2 + 2 \times a_3 + a_4). \tag{10}$$

Definition 8 (Fuzzy distance) We adopt a non-negative function to calculate the distance $D_{p,q}(\tilde{A}, \tilde{B})$ between two fuzzy numbers \tilde{A} and \tilde{B} , where $p > 1$ and $0 < q < 1$ [44]:

$$D_{p,q}(\tilde{A}, \tilde{B}) = \begin{cases} \left[(1-q) \int_0^1 |\tilde{A}_\alpha^- - \tilde{B}_\alpha^-|^p d\alpha + q \int_0^1 |\tilde{A}_\alpha^+ - \tilde{B}_\alpha^+|^p d\alpha \right]^{\frac{1}{p}}, & p < \infty, \\ (1-q) \sup_{0 < \alpha \leq 1} (|\tilde{A}_\alpha^- - \tilde{B}_\alpha^-|) + q \inf_{0 < \alpha \leq 1} (|\tilde{A}_\alpha^+ - \tilde{B}_\alpha^+|), & p = \infty. \end{cases} \tag{11}$$

The first parameter p determines the analytical features of $D_{p,q}$, while the second parameter q describes the subjective weight attributed to the end points of support, i.e., $(\tilde{A}_\alpha^-, \tilde{A}_\alpha^+)$ for the fuzzy number \tilde{A} [27]. When q is close to 0, the left side of the support is given more consideration. The value $q = 1/2$ is adopted in this study as the significance of the end points of the support of fuzzy numbers is assumed to be the same. Moreover, $p = 2$ is the most frequently used for fuzzy distance [27,29,43]. For two TFNs $\tilde{A} = (a_1, a_2, a_3)$ and $\tilde{B} = (b_1, b_2, b_3)$, when $p = 2$ and $q = 1/2$, their distance is calculated as:

$$D_{2,1/2}(\tilde{A}, \tilde{B}) = \sqrt{\frac{1}{6} \left[\sum_{i=1}^3 (a_i - b_i)^2 + (a_2 - b_2)^2 + \sum_{i \in \{1,2\}} (a_i - b_i)(a_{i+1} - b_{i+1}) \right]}. \tag{12}$$

And for two TrFNs $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$, their distance is calculated as:

$$D_{2,1/2}(\tilde{A}, \tilde{B}) = \sqrt{\frac{1}{6} \left[\sum_{i=1}^4 (a_i - b_i)^2 + \sum_{i \in \{1,3\}} (a_i - b_i)(a_{i+1} - b_{i+1}) \right]}. \tag{13}$$

Ranking methods of fuzzy numbers

How to rank the fuzzy path cost is of great importance in solving the FSPP. A number of studies have investigated how to rank fuzzy numbers [45–47]. In this paper, we adopt two methods:

Ranking method 1 The fuzzy numbers are ranked according to their fuzzy graded mean values [28,30].

Ranking method 2 The fuzzy numbers are ranked according to their distances between the fuzzy minimum number ($M\tilde{N}$) [27,29,43]. For two TFNs $\tilde{A} = (a_1, a_2, a_3)$ and $\tilde{B} = (b_1, b_2, b_3)$, their $M\tilde{N}$ is defined as:

$$M\tilde{N}(\tilde{A}, \tilde{B}) = (\min(a_1, b_1), \min(a_2, b_2), \min(a_3, b_3)). \tag{14}$$

In the same way, the $M\tilde{N}$ of two TrFNs $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$ is:

$$M\tilde{N}(\tilde{A}, \tilde{B}) = (\min(a_1, b_1), \min(a_2, b_2), \min(a_3, b_3), \min(a_4, b_4)). \tag{15}$$

We take two TrFNs $\tilde{A} = (2, 5, 9, 14)$ and $\tilde{B} = (4, 6, 11, 12)$ as an example, as shown in Fig. 3. Using ranking method 1, $P(\tilde{A}) = 7.33$ and $P(\tilde{B}) = 8.33$, so \tilde{A} is less than \tilde{B} . Using ranking method 2, $M\tilde{N} = (2, 5, 9, 12)$. The distances are $D_{2,1/2}(\tilde{A}, M\tilde{N}) = 0.82$ and $D_{2,1/2}(\tilde{B}, M\tilde{N}) = 1.35$, so \tilde{A} is less than \tilde{B} .

Pareto concepts

Pareto concepts are firstly applied in economic research [48] and become the core concept of the MOOP [15]. For a MOPOP with q objectives, it aims to minimize all of them. Let P_1 and P_2 be two feasible paths, and $f(P)(i)$ denote the i^{th} objective value of path $P, i = 1, \dots, q$. If P_1 is no worth than P_2 on all objectives and better than P_2 on at least one objective, P_1 Pareto dominates P_2 . We use $P_1 < P_2$ to

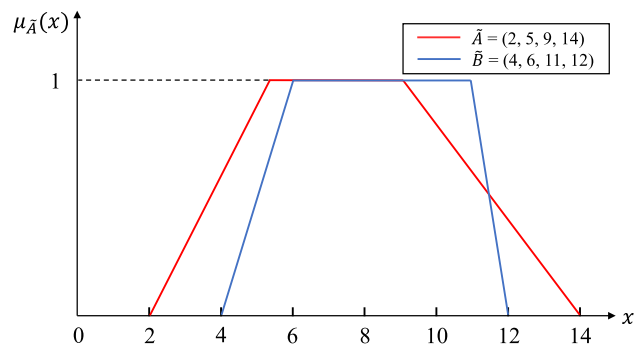


Fig. 3 An illustration of two TrFNs

denote that P_1 Pareto dominates P_2 , and its mathematical formulation is:

$$\begin{cases} f(P_1)(i) \leq f(P_2)(i), & \forall i \in [1, \dots, q], \\ f(P_1)(j) < f(P_2)(j), & \exists j \in [1, \dots, q]. \end{cases} \quad (16)$$

“Pareto optimal” means that the progression on one objective must lead to regression on the others. Let a set Ω_P contain all possible paths. If a path P^* is Pareto optimal, there is no path P that Pareto dominates P^* . The mathematical relation is:

$$\exists P \in \Omega_P, P < P^*. \quad (17)$$

All Pareto optimal paths construct the Pareto set. The projection of a Pareto optimal path in the objective space is a Pareto point, and all Pareto points form the Pareto front of the MOPOP [7].

Basic ideas behind the RSA

The RSA is proposed by simulating natural ripple patterns on a water surface: all ripples spread out in all directions at the same speed. A ripple triggers another ripple when encountering an obstacle. The basic principle behind this phenomenon is that the first ripple that reaches a node travels along the shortest path [36]. From the perspective of algorithm design, each ripple has basic features: the epicenter, state, and radius. The epicenter is the node where the ripple is generated. Only ripples in the active state spread out. In the RSA, ripples spread out on a network where the distance between two nodes is represented by a crisp number. We present the basic steps of the RSA for the SPP as follows:

1. An active ripple is initialized at the source node.
2. At each time, all active ripples spread out at the same speed.
3. If a ripple reaches an unvisited node first, it triggers another ripple.
4. If an active ripple has visited all nodes connected to its epicenter, it becomes inactive.
5. If a ripple has visited the destination node, the SPP is solved. The traveling path of this ripple is the shortest.

Figure 4 provides a toy example of the RSA for the SPP, and only active ripples are plotted. Following the steps mentioned above, ripple R_3 reaches the destination node d first and becomes inactive. Its traveling path $s-2-d$ is the shortest path. We can observe that the whole processes resemble a ripple relay race.

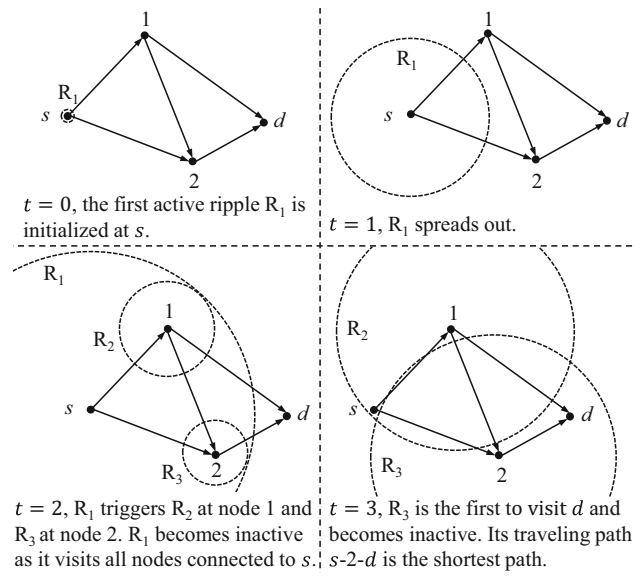


Fig. 4 The illustration of how the RSA solves the SPP

Mathematical formulation

In this section, we provide the mathematical formulation of the FMOPOP. We consider a directed network $G = \{V, E\}$, where V is the node set, and E is the arc set. $V = \{1, 2, \dots, n_n\}$ contains n_n different nodes, and $E = \{(i, j) \mid i \in V, j \in V\}$ contains n_a different arcs. The arc from node i to node j is denoted as (i, j) . Moreover, there are n_c crisp weights and n_f fuzzy weights associated with each arc, and n_w denotes the total number of weights. For arc (i, j) , its m^{th} crisp weight is denoted as c_{ij}^m , and its n^{th} fuzzy weight is \tilde{c}_{ij}^n . A path P_{sd} from the source node s to the destination node d is a sequence of nodes $P_{sd} = \{s = n_1, n_2, \dots, n_{m-1}, n_m = d\}$, where $(n_h, n_{h+1}) \in E$ holds for $\forall h = 1, \dots, m - 1$. The mathematical formulation of the FMOPOP is:

$$\begin{aligned} \min & \sum_{i=1}^{n_n} \sum_{j=1}^{n_n} x_{ij} [c_{ij}^1, \dots, c_{ij}^{n_c}, \tilde{c}_{ij}^1, \dots, \tilde{c}_{ij}^{n_f}] \\ \text{s.t.} & \sum_{i=1}^{n_n} x_{ij} - \sum_{j=1}^{n_n} x_{ji} = \begin{cases} 1, & \text{for } i = s, \\ 0, & \text{otherwise,} \\ -1, & \text{for } i = d, \end{cases} \\ & x_{ij} \in \{0, 1\}, \quad \forall i, j \in [1, \dots, n_n]. \end{aligned} \quad (18)$$

In model (18), $x_{ij} = 1$ denotes that arc (i, j) is a part of the path, otherwise $x_{ij} = 0$.

Moreover, we define the objective function of a path $P = \{n_1, n_2, \dots, n_{m-1}, n_m\}$ as:

$$f(P, i)(j) = \begin{cases} 0, & i = 1, 1 \leq j \leq n_c, \\ (0, 0, 0) \text{ or } (0, 0, 0, 0), & i = 1, n_c < j \leq n_w, \\ f(P, i - 1)(j) + c_{n_{i-1}n_i}^j, & i > 1, 1 \leq j \leq n_c, \\ f(P, i - 1)(j) \oplus \tilde{c}_{n_{i-1}n_i}^j, & i > 1, n_c < j \leq n_w. \end{cases} \quad (19)$$

In Eq. (19), $f(P, i)(j)$ denotes the j^{th} objective value of the sub-path of P from node n_1 to node n_i . The objective value is stored in a list structure, in which the first n_c components denote crisp values, and the last n_f components denote fuzzy values. For simplicity, we use $f(P)$ to denote $f(P, m)$ hereafter.

The adapted RSA

This section proposes an adapted RSA for the FMOPOP and conducts theoretical analyses to demonstrate its optimality and efficiency.

Algorithm description

Different from the RSA for the SPP illustrated in Subsection “Basic ideas behind the RSA”, we need to adapt the triggering condition and termination judgment for the FMOPOP. We first introduce a new definition of Pareto optimal ripple (POR):

Definition 9 (POR) If a ripple R reaches node n along path P , and the traveling paths of other ripples that reach n before R do not Pareto dominate P , then R is a POR at node n .

The new triggering condition is that if an incoming ripple is a POR at this node, it triggers a new ripple. Moreover, the termination judgment is modified as all ripples are inactive. Other notations used in the RSA are listed in Table 1.

Algorithm 1 is the RSA for the FMOPOP. The flowchart in Fig. 5 uses metaphor-based language to illustrate the processes of the RSA step by step. After executing the RSA, the traveling paths of all PORs at the destination d are exactly all Pareto optimal paths of the FMOPOP. According to Line 3 of Algorithm 1, ripples spread out on the network described by the k^{th} crisp weight. The ripple spreading speed v is chosen as the minimum value of the k^{th} crisp weight, which can increase efficiency while maintaining optimality [39]. The new ripple triggering condition is Line 12, where $n \notin P(i)$ is to avoid loops in paths, $c_{E(i)n}^k \leq R(i) < c_{E(i)n}^k + v$ denotes that ripple

Table 1 The notations in the RSA and their meanings

Notations	Meanings
R	Radius set, the radius of ripple i is $R(i)$
S	State set, $S(i) = 0/1$ denotes that ripple i is inactive/active
E	Epicenter set, ripple i is generated at node $E(i)$
P	Path set, ripple i travels along $P(i)$ from s to $E(i)$
O	Objective value set, $O(i) = f(P(i))$
F	Forward node set, $F(i) = \{j \mid (i, j) \in E\}, i = 1 \dots, n_n$
n_r	The current ripple number
v	The ripple spreading speed
t	The time index
Ω_{POR}	POR set, $\Omega_{POR}(n)$ contains all PORs at node n
n_c	The number of crisp weights
n_f	The number of fuzzy weights
n_n	The number of nodes
s	The source node
d	The destination node
P_R	The output Pareto optimal path set
O_R	The output Pareto front

Algorithm 1 The RSA for the FMOPOP

Input: $G = \{V, E\}, c_{ij}^m, \tilde{c}_{ij}^n, (i, j) \in E, m = 1, \dots, n_c, n = 1, \dots, n_f$
Output: P_R, O_R
1: $R = S = E = P = O = P_R = O_R = \{\}, n_r = 0, \Omega_{POR}(n) = \{\}, n = 1, \dots, n_n$
2: $t = 0$
3: $v = \min(c_{ij}^k)$ for k with minimum value of $\max(c_{ij}^k) / \min(c_{ij}^k)$
4: $n_r = 1, R(n_r) = 0, S(n_r) = 1, E(n_r) = s, P(n_r) = \{s\}, O(n_r) = f(P(n_r))$
5: $n_t = n_r$
6: **while** $\exists i \in [1, \dots, n_t], S(i) = 1$ **do**
7: $t = t + 1$
8: **for** $i = 1, \dots, n_r$ **do**
9: **if** $S(i) = 1$ **then**
10: $R(i) = R(i) + v$
11: **for** $n \in F(E(i))$ **do**
12: **if** $(n \notin P(i)) \wedge (c_{E(i)n}^k \leq R(i) < c_{E(i)n}^k + v) \wedge \text{IsPOR}(i)$ **then**
13: $n_r = n_r + 1, R(n_r) = R(i) - c_{E(i)n}^k, S(n_r) = 1, E(n_r) = n, P(n_r) = P(i) \cup \{n\}, O(n_r) = f(P(n_r)), \Omega_{POR}(n) = \Omega_{POR}(n) \cup \{n_r\}$
14: **if** $n = d$ **then**
15: $P_R = P_R \cup P(n_r), O_R = O_R \cup O(n_r)$
16: **end if**
17: **end if**
18: **end for**
19: **if** $\forall n \in F(E(i)), R(i) \geq c_{E(i)n}^k$ **then**
20: $S(i) = 0$
21: **end if**
22: **end if**
23: **end for**
24: $n_t = n_r$
25: **end while**

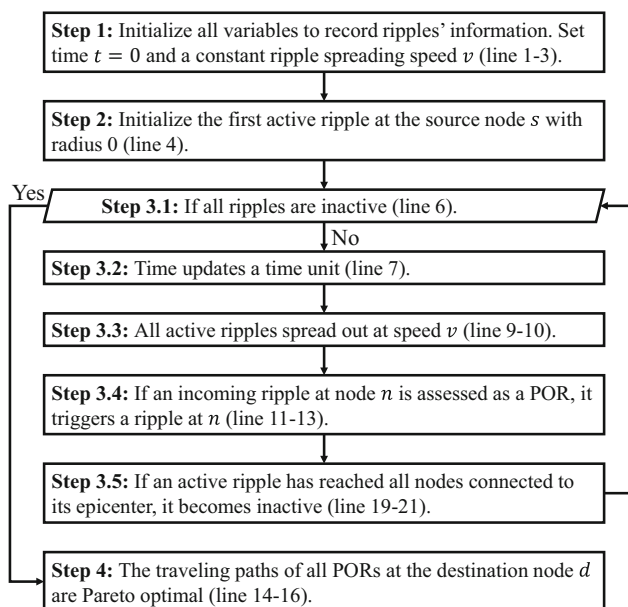


Fig. 5 The flowchart of the RSA using the metaphor-based language

i is an incoming ripple at node n in this period, and $IsPOR(i)$ is to judge whether ripple i is a POR or not. Algorithm 2 provides the pseudo code of determining whether path P_1 Pareto dominates P_2 using ranking method 2, in which $f(P)$ denotes the objective value of P according to (19). Actually, we can adopt any fuzzy number ranking method. As long as the Pareto domination relation between two paths can be determined, the RSA can solve the FMOPOP.

Algorithm 2 Determine whether $P_1 < P_2$

```

Input:  $f(P_1), f(P_2), n_c, n_w$ 
Output: True or False
1:  $num \leftarrow 0$ 
2: for  $i = 1, \dots, n_w$  do
3:   if  $i \leq n_c$  then
4:     if  $f(P_1)(i) > f(P_2)(i)$  then
5:       return False
6:     else if  $f(P_1)(i) < f(P_2)(i)$ 
7:        $num \leftarrow num + 1$ 
8:     end if
9:   else
10:     $dis1 \leftarrow D_{2,1/2}(M\tilde{N}(f(P_1)(i), f(P_2)(i)), f(P_1)(i))$ 
11:     $dis2 \leftarrow D_{2,1/2}(M\tilde{N}(f(P_1)(i), f(P_2)(i)), f(P_2)(i))$ 
12:    if  $dis1 > dis2$  then
13:      return False
14:    else if  $dis1 < dis2$ 
15:       $num \leftarrow num + 1$ 
16:    end if
17:  end if
18: end for
19: if  $num > 0$  then
20:   return True
21: else
22:   return False
23: end if
    
```

Theoretical analysis

In this subsection, we prove that the traveling paths of PORs at each node are exactly all Pareto optimal paths to this node. Moreover, the time complexity of the RSA is calculated.

Lemma 1 *In the RSA, the i^{th} ripple that reaches a node travels along the i^{th} shortest path.*

Lemma 2 *A Pareto optimal path $P = \{s = n_1, \dots, n_m = d\}$ is formed by Pareto optimal sub-paths, i.e., $\{n_1, \dots, n_i\}$ is Pareto optimal from s to node n_i .*

Lemma 1 states the basic optimization principle of RSAs, which directly comes from the theorems in [38]. And Lemma 2 comes from the theorems in [49]. Readers can get the detailed proof in these two papers.

Lemma 3 *The traveling path P of a POR at node n is a Pareto optimal path from the source node s to n .*

Proof The proof is divided into two scenarios: (1) According to Definition 9, the traveling paths of all ripples that reach n before the POR do not Pareto dominate P . (2) If a ripple reaches n after the POR along path P_1 . We assume that ripples spread out on the network described by the k^{th} crisp weight. According to Lemma 1, its k^{th} objective value is greater than the k^{th} objective value of P , i.e., $f(P_1)(k) > f(P)(k)$. According to relation (16), it is impossible that $P_1 < P$. Therefore, Lemma 3 holds. □

Theorem 1 *After executing the RSA, the traveling paths of all PORs at a node are all Pareto optimal paths to this node. In other words, we can determine all Pareto optimal paths from the source node s to all other nodes within a single run.*

Proof According to Lemma 2, all the sub-paths of a Pareto optimal path are also Pareto optimal. According to Definition 9, the ripple traveling along a Pareto optimal path is assessed as a POR. Step 3.4 in Fig. 5 denotes that if an incoming ripple is a POR, it triggers a new ripple. All Pareto optimal paths are traveled by PORs at this node, i.e., the RSA does not miss any Pareto optimal path. Moreover, Lemma 3 guarantees that the traveling paths of PORs are Pareto optimal. Therefore, Theorem 1 is true. □

Theorem 2 *The time complexity of the RSA for the FMOPOP is $O(n_w \times n_a \times n_{POR}^2)$, where n_{POR} is the average number of PORs at each node.*

Proof In the RSA, only active ripples need computational steps. There are n_n nodes and n_a arcs on the network, so each node has n_a/n_n arcs on average. Each active ripple takes average n_{at} time units to travel through the longest arc connected to its epicenter, and then it becomes inactive. During its active state, there are three parts of computational

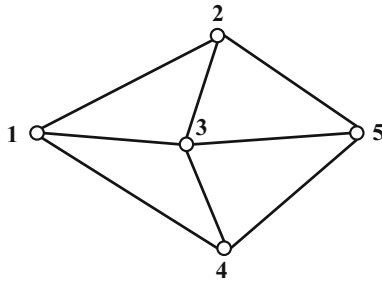


Fig. 6 The undirected network for Example 1

steps: (1) ripple spreading and comparing its radius with the arc’s length; (2) determining the objective value of its traveling path; (3) determining whether it is a POR at its visiting node. The first part executes every time unit, which takes $(2 \times n_{at} \times n_a/n_n)$ steps. The second and third parts only execute when reaching a new node, and there are average n_{POR} ripples to be compared with on n_w objectives. These two parts take $((1 + n_{POR}) \times n_w \times n_a/n_n)$ steps. There are $(n_n \times n_{POR})$ active ripples ever generated, so the whole process of the RSA takes n_{cs} computational steps:

$$n_{cs} = n_n \times n_{POR} \times (2 \times n_{at} + (1 + n_{POR}) \times n_w) \times n_a/n_n = (2 \times n_{at} + (1 + n_{POR}) \times n_w) \times n_{POR} \times n_a. \tag{20}$$

In large-scale problems, the number of PORs is much greater than n_{at} and 1. So the time complexity is calculated as $O(n_w \times n_a \times n_{POR}^2)$. □

Examples and comparative experiments

In this section, we present two numerical examples and some comparative experiments to demonstrate the capability and robustness of the newly proposed RSA.

Numerical examples

Example 1 We first consider an undirected network shown in Fig. 6, consisting of 5 nodes and 8 arcs. A crisp weight and a fuzzy weight described by a TFN are associated with each arc, as shown in Table 2. The crisp weight of an arc represents its length. The source node is node 1, and all other nodes are destination nodes. The aim is to find all Pareto optimal paths from node 1 to destination nodes.

At $t = 0$, the first active ripple R_1 is initialized at the source node s . The ripple spreading speed v is chosen as $\min(c_{ij}^1) = 32$.

At $t = 1$, R_1 spreads out, and its radius becomes 32, as shown in Fig. 7.

Table 2 The crisp and fuzzy weights of the network in Fig. 6

Arc	Crisp weight	Fuzzy weight
1-2	62	(3, 10, 20)
1-3	44	(4, 5, 7)
1-4	67	(3, 10, 19)
2-3	33	(5, 9, 20)
2-5	52	(3, 9, 12)
3-4	32	(2, 5, 6)
3-5	52	(14, 17, 20)
4-5	54	(2, 3, 16)

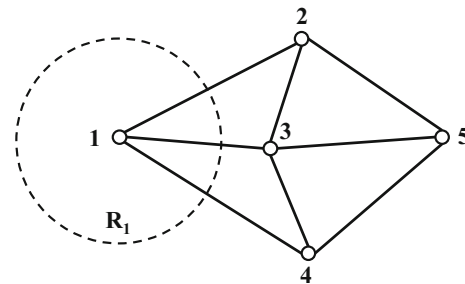


Fig. 7 The ripple spreading motion at $t = 1$

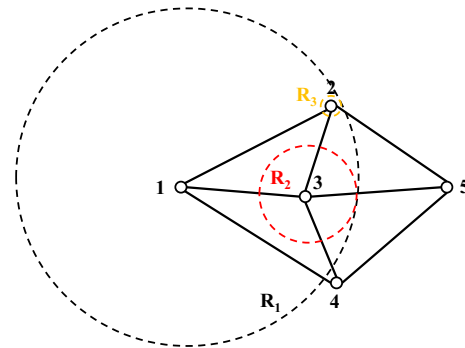


Fig. 8 The ripple spreading motion at $t = 2$

At $t = 2$, R_1 reaches node 3 and triggers a new ripple R_2 (red, radius = 20). R_1 reaches node 2 and triggers a new ripple R_3 (yellow, radius = 2), as shown in Fig. 8.

At $t = 3$, R_1 reaches node 4 and becomes inactive as it has visited all nodes connected to node 1, and we eliminate it from the network. R_2 reaches node 2, node 4, and node 5 and also becomes inactive. R_1 triggers R_4 (blue, radius = 29) at node 4. R_2 triggers R_5 (green, radius = 20) at node 4 and triggers R_6 at node 5 (black, radius = 0), as shown in Fig. 9. It is worth noticing that R_2 reaches node 2, and R_3 reaches node 3, but they do not trigger any ripple. Take R_2 as an example: when it reaches node 2, its objective value is $\{77, (9, 14, 27)\}$. It is Pareto dominated by the existing POR R_2 at node 3, whose objective value is $\{62, (3, 10, 20)\}$. So

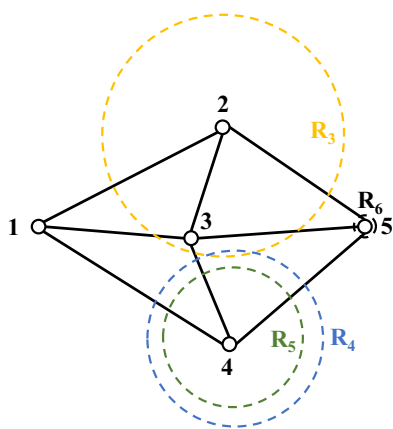


Fig. 9 The ripple spreading motion at $t = 3$

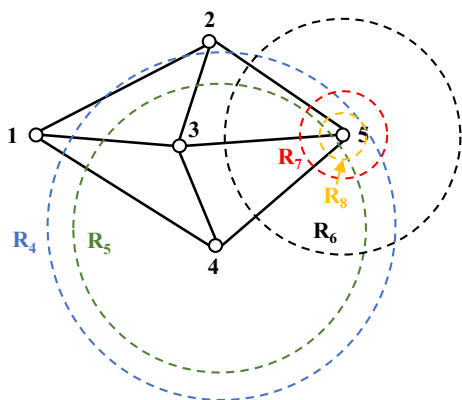


Fig. 10 The ripple spreading motion at $t = 4$

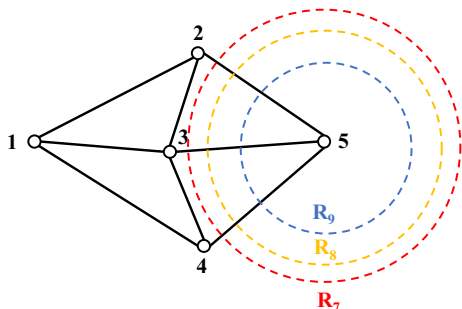


Fig. 11 The ripple spreading motion at $t = 5$

R_2 is not a POR at node 3. According to **Step 3.4** in Fig. 5, it does not trigger any ripple.

At $t = 4$, R_3 reaches node 5 and becomes inactive. At node 5, R_3 triggers R_7 (red, radius = 14), and R_4 triggers R_8 (yellow, radius = 7), as shown in Fig. 10.

At $t = 5$, R_5 triggers R_9 (blue, radius = 30) at node 5. R_4 , R_5 , and R_6 become inactive. R_7 and R_8 also spread out at $v = 32$, as shown in Fig. 11.

At $t = 6$, R_7 , R_8 , and R_9 reach all nodes connected to node 5 and become inactive. Moreover, they are not PORs at

Table 3 All PORs at each node after solving Example 1

Node	Ripple	Traveling Path	Objective value
2	R_3	1-2	{62, (3, 10, 20)}
3	R_2	1-3	{44, (4, 5, 7)}
4	R_4	1-4	{67, (3, 10, 19)}
	R_5	1-3-4	{76, (6, 10, 13)}
5	R_6	1-3-5	{96, (18, 22, 27)}
	R_7	1-2-5	{114, (6, 19, 32)}
	R_8	1-4-5	{121, (5, 13, 35)}
	R_9	1-3-4-5	{130, (8, 13, 29)}

Table 4 All possible paths to each node and the corresponding cost

Destination	Possible path	Crisp cost	Fuzzy cost	Optimal 1?	Optimal 2?
node 2	1-2	62	(3, 10, 20)	Yes	Yes
	1-3-2	77	(9, 14, 27)	No	No
	1-3-5-2	148	(21, 31, 39)	No	No
	1-4-3-2	132	(10, 24, 45)	No	No
	1-4-5-2	173	(8, 22, 47)	No	No
	1-4-3-5-2	203	(22, 41, 57)	No	No
node 3	1-3	44	(4, 5, 7)	Yes	Yes
	1-2-3	95	(8, 19, 40)	No	No
	1-4-3	99	(5, 15, 25)	No	No
	1-2-5-3	166	(20, 36, 52)	No	No
	1-4-5-3	173	(19, 30, 55)	No	No
	1-2-5-4-3	200	(10, 27, 54)	No	No
node 4	1-4-5-2-3	206	(13, 31, 67)	No	No
	1-4	67	(3, 10, 19)	Yes	Yes
	1-3-4	76	(6, 10, 13)	Yes	Yes
	1-2-3-4	127	(10, 24, 46)	No	No
	1-3-5-4	150	(20, 25, 43)	No	No
	1-2-5-4	168	(8, 22, 48)	No	No
node 5	1-2-5-3-4	198	(22, 41, 58)	No	No
	1-2-5	114	(6, 19, 32)	Yes	Yes
	1-3-5	96	(18, 22, 27)	Yes	Yes
	1-4-5	121	(5, 13, 35)	Yes	Yes
	1-2-3-5	147	(22, 36, 60)	No	No
	1-3-2-5	129	(12, 23, 39)	No	No
	1-3-4-5	130	(8, 13, 29)	Yes	Yes
	1-4-3-5	151	(19, 32, 45)	No	No
	1-2-3-4-5	181	(12, 27, 62)	No	No
	1-4-3-2-5	184	(13, 33, 57)	No	No

their forward nodes. Up to now, there is no active ripple, and Example 1 is solved.

Table 3 lists all PORs at each node after solving Example 1. Table 4 lists all possible paths to each node and their corresponding values. Optimal 1 and Optimal 2 in Table 4 denote whether the path is Pareto optimal using the ranking

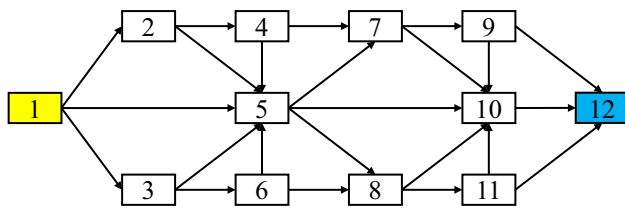


Fig. 12 The directed network for Example 2

Table 5 The crisp and fuzzy weights of the network in Fig. 12

Arc	Crisp 1	Crisp 2	Fuzzy 1	Fuzzy 2
1-2	8	1	(4, 7, 15)	(12, 19, 20)
1-3	4	5	(8, 12, 17)	(6, 14, 15)
1-5	9	7	(14, 15, 19)	(2, 19, 20)
2-4	2	2	(8, 14, 16)	(2, 19, 20)
2-5	6	8	(2, 5, 13)	(2, 10, 12)
3-5	5	8	(13, 18, 19)	(5, 9, 13)
3-6	8	1	(7, 8, 13)	(2, 10, 11)
4-5	6	2	(14, 17, 20)	(7, 11, 20)
4-7	4	5	(4, 6, 17)	(2, 12, 16)
5-7	6	9	(4, 7, 11)	(7, 10, 20)
5-8	4	3	(2, 11, 12)	(17, 19, 20)
5-10	9	7	(2, 5, 16)	(6, 10, 20)
6-5	9	4	(2, 3, 17)	(2, 11, 20)
6-8	4	3	(8, 15, 20)	(5, 11, 19)
7-9	6	8	(8, 9, 17)	(5, 6, 11)
7-10	4	7	(3, 12, 15)	(6, 12, 17)
8-10	9	2	(5, 15, 19)	(11, 14, 19)
8-11	1	6	(9, 10, 14)	(7, 9, 12)
9-10	3	8	(2, 11, 18)	(5, 12, 19)
9-12	7	1	(6, 11, 20)	(4, 9, 19)
10-12	9	6	(10, 18, 19)	(2, 3, 7)
11-10	5	3	(4, 8, 16)	(9, 13, 19)
11-12	9	2	(4, 7, 19)	(5, 13, 16)

methods 1 and 2 in the second section. After comparison, we observe that all the traveling paths of PORs at each node are exactly all Pareto optimal paths to this node. Moreover, the RSA determines Pareto optimal paths to all other nodes within a single run.

Example 2 The directed network of this example is shown in Fig. 12, consisting of 12 nodes and 23 arcs. Each arc has two crisp weights and two fuzzy weights, as shown in Table 5. Crisp 1 and Crisp 2 denote two crisp weights. Likewise, Fuzzy 1 and Fuzzy 2 denote two fuzzy weights, as shown in Table 5. The source node is node 1, and the destination node is node 12.

Table 6 lists all Pareto optimal paths determined by the RSA. Example 2 demonstrates that the RSA also can provide

Table 6 The result of Example 2 solved by the RSA

Pareto optimal path	Objective value
1-5-8-11-12	{23, 18, (29, 43, 64), (31, 60, 68)}
1-3-6-8-11-12	{26, 17, (36, 52, 83), (25, 57, 73)}
1-5-10-12	{27, 20, (26, 38, 54), (10, 32, 47)}
1-2-4-7-9-12	{27, 17, (30, 47, 85), (25, 65, 86)}
1-2-4-5-8-11-12	{30, 16, (41, 66, 96), (50, 90, 108)}
1-5-8-10-12	{31, 18, (31, 59, 69), (32, 55, 66)}
1-3-6-8-10-12	{34, 17, (38, 68, 88), (26, 52, 71)}
1-2-4-5-8-10-12	{38, 16, (43, 82, 101), (51, 85, 106)}

complete solutions for the FMOPOP where the number of crisp and fuzzy weights is more than two.

Comparative experiments

This subsection conducts comparative experiments to evaluate the computational efficiency of the RSA on large-scale networks. To the best of our knowledge, there is no exact algorithm to solve the FMOPOP using various fuzzy number ranking methods. Given the complexity of calculating the distance between two fuzzy numbers, as shown in relation (11), it is difficult to adapt existing MOPOP algorithms to solve the FMOPOP. We choose one of the most classic multi-objective evolutionary algorithms, the NSGA-II, as the benchmark for its outstanding flexibility [18]. Each path is encoded in the same method as [50]. To adapt the NSGA-II, we combine the fast-non-dominated-sort with Algorithm 2, which can determine the Pareto domination relation of two paths. Moreover, the crowding-distance-assignment in the NSGA-II is only conducted on crisp objective values. The parameters of the NSGA-II are set up as follows: the population size is 100, the number of generations is 100, the crossover probability is 0.9, and the mutation probability is 0.15.

The test networks are randomly generated and contain at least one Hamiltonian cycle. The source and destination are randomly chosen. Each arc has n_c crisp weights uniformly distributed between 1 and 20 and n_f fuzzy weights described by TFNs. For each TFN (a_1, a_1, a_3) , a_1, a_2 , and a_3 are distributed between 1 and 20. To evaluate the efficiency of the RSA under different numbers of nodes and objectives, we set n_n in the range of 50 and 500 with the increment of 50 and $n_c = n_f \in [1, 3, 5, 10]$. For each instance, we set $n_a = 4 \times n_n$. There are 40 instances in total, and we conducted ten experiments for each instance. The results of the two algorithms are gathered after solving the same problem. Using the fast-non-dominated-sort [18], all non-dominated solutions are true Pareto optimal paths, and the others are

Table 7 The results of the comparative experiments of the RSA and NSGA-II

n_n		$n_c=1, n_f=1$			$n_c=3, n_f=3$			$n_c=5, n_f=5$			$n_c=10, n_f=10$		
		RT	APN	TPN	RT	APN	TPN	RT	APN	TPN	RT	APN	TPN
50	RSA	0.01	1.9	1.9	0.05	5.2	5.2	0.11	10.4	10.4	0.33	13.1	13.1
	NSGA-II	10.14	1.9	1.9	16.59	4.7	4.7	26.09	9.3	9.3	52.70	11.9	11.9
100	RSA	0.03	2.1	2.1	0.12	8.5	8.5	0.30	8.4	8.4	0.95	19.6	19.6
	NSGA-II	10.00	2.1	2.1	19.75	5.8	5.8	30.29	6.1	5.9	39.02	11.5	11.5
150	RSA	0.04	2.4	2.4	0.21	5.0	5.0	0.58	15.0	15.0	2.01	31.9	31.9
	NSGA-II	10.36	2.4	2.1	19.46	4.2	4.1	24.38	7.8	7.7	39.54	14.4	14.1
200	RSA	0.06	2.3	2.3	0.33	9.3	9.3	0.74	16.9	16.9	2.89	31.0	31.0
	NSGA-II	10.80	1.7	1.7	16.84	6.9	6.1	25.32	8.6	8.5	45.17	11.5	11.2
250	RSA	0.07	1.7	1.7	0.38	5.5	5.5	1.14	19.3	19.3	3.81	33.4	33.4
	NSGA-II	11.62	1.6	1.6	21.68	3.4	3.0	27.00	8.5	8.0	49.98	10.8	9.9
300	RSA	0.09	2.1	2.1	0.57	12.7	12.7	1.26	13.8	13.8	4.96	25.3	25.3
	NSGA-II	11.20	1.8	1.7	18.62	5.7	5.2	31.96	7.0	5.9	65.84	7.8	6.6
350	RSA	0.11	2.9	2.9	0.65	11.4	11.4	1.65	11.8	11.8	6.52	37.3	37.3
	NSGA-II	12.19	2.5	1.9	23.97	4.2	3.6	35.76	4.2	3.8	51.99	10.6	9.4
400	RSA	0.12	2.4	2.4	0.75	9.2	9.2	1.93	16.4	16.4	7.59	31.9	31.9
	NSGA-II	13.38	1.9	1.6	24.83	4.2	3.5	32.79	5.8	5.4	60.71	8.1	7.5
450	RSA	0.13	2.8	2.8	0.91	10.0	10.0	2.33	12.2	12.2	8.84	35.3	35.3
	NSGA-II	12.62	1.9	1.9	24.52	4.9	3.7	37.41	3.9	3.7	54.54	8.4	7.4
500	RSA	0.17	2.3	2.3	0.98	9.6	9.6	2.58	18.2	18.2	10.58	28.2	28.2
	NSGA-II	13.97	2.6	1.6	22.81	5.2	3.9	34.23	5.8	4.4	57.66	6.4	6.2

false Pareto optimal paths. We record the running time (RT) in seconds, the number of Pareto optimal paths (APN), and the number of true Pareto optimal paths (TPN).

Table 7 presents the experimental data, from which one may conclude that:

- The RSA has outstanding computational efficiency. The average RTs of the RSA and NSGA-II are 1.67 s and 28.69 s, which implies that the RSA solves the FMOPOP 17.18 times faster than the NSGA-II on average.
- Due to the random features of the NSGA-II, its APN is no greater than that of the RSA. Moreover, the TPN of the NSGA-II is lower than the APN. It implies that the NSGA-II may miss some Pareto optimal paths or output some false Pareto optimal paths. On the contrary, the RSA finds all Pareto optimal paths.
- With the increment of n_n and n_c , the percentage of the Pareto optimal paths found by the NSGA-II decreases. The parameters of the NSGA-II should be readjusted according to the problem scale. Otherwise, it may lose computational efficiency or miss Pareto optimal solutions. Under different numbers of weights and nodes, the RSA finds all Pareto optimal paths within a reasonable computation time. What is striking about the RSA is its robustness, as it maintains optimality without readjusting any parameters.

Up to now, the existing studies have not dealt with the one-to-all FMOPOP, which aims to determine all Pareto optimal paths from the source node to all other nodes. We provide a comparative experiment of the one-to-all FMOPOP. We select one node as the source node from a network with 50 nodes and 200 arcs, and the others are destination nodes. Each arc has 10 crisp weights and 10 fuzzy weights. We can solve the one-to-all FMOPOP using the RSA within a single run. However, the NSGA-II needs to run 49 times to find the Pareto optimal paths from the source node to all other nodes. We ran the two algorithms on a one-to-all FMOPOP and recorded the TPN to each destination node. Fig. 13 presents the comparative results, in which the TPN of the RSA is shown in black, and the NSGA-II is in red. From the figure, the RSA finds all Pareto optimal paths to other nodes, while the NSGA-II misses some of them. Moreover, the RT of the RSA is 0.36 s, while the NSGA-II is 1752.11 s. For the FMOPOP with multiple destination nodes, the advantage of the computational efficiency of the RSA is further increased compared to other algorithms.

From two numerical examples and comparative experiments, we can obtain the following observations:

- The proposed RSA can find all Pareto optimal paths to all other nodes within a single run. Moreover, the RSA only finds true Pareto optimal paths.

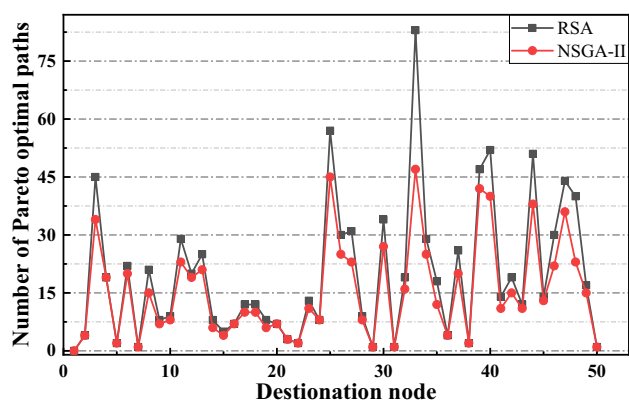


Fig. 13 The TPN of the RSA and NSGA-II in the one-to-all FMOPOP experiment

- The RSA maintains its optimality when the number of objectives is more than two.
- Various fuzzy numbers and ranking methods can be adopted in the RSA on account of its flexibility.
- We can use metaphor-based descriptions and figures to explain the detailed processes of the RSA, which is more comprehensible.
- The RSA is an efficient and robust algorithm for the FMOPOP.
- The main limitation of the RSA is that there must be at least one crisp weight. We need a network described by crisp weights to conduct the ripple relay race.

Conclusion

Networks with crisp and fuzzy weights are of interest within a wide range of scientific and industrial processes. In this paper, we define this optimization problem as the fuzzy multi-objective path optimization problem (FMOPOP). Each arc has multiple crisp weights denoted by positive real numbers and multiple fuzzy weights denoted by triangular fuzzy numbers (TFNs) or trapezoidal fuzzy numbers (TrFNs). We adopt two different methods to rank fuzzy numbers based on their fuzzy graded mean values and distances from the fuzzy minimum number. The main contribution of this paper is to propose an algorithm that can determine all Pareto optimal paths from the source node to all other nodes within a single run. To this end, a nature-inspired algorithm, the ripple-spreading algorithm (RSA), is proposed. The RSA simulates ripple spreading and triggering motions on a water surface. We adapt the triggering condition of new ripples to make the RSA solve the FMOPOP. We prove the optimality of the RSA theoretically and calculate its time complexity. Two numerical examples of the FMOPOP are provided. In the first example, we illustrate how the RSA solves the FMOPOP step by step using metaphor-based language and figures. The

second example demonstrates that the RSA can solve the FMOPOP where the number of weights is more than two. Comparative experiments imply the high computational efficiency and robustness of the RSA for the FMOPOP.

In the future study, we want to extend the RSA to path optimization in vague, intuitionistic, and neutrosophic environments [26]. More ranking methods and interval-valued fuzzy sets could be adopted. Finally, the RSA can combine with some problem-specific operations to solve real-world applications.

Funding Fundamental Research Funds for Central Universities - Civil Aviation University of China special funds (2000530441)

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Fu L, Sun D, Rilett LR (2006) Heuristic shortest path algorithms for transportation applications: state of the art. *Comput Oper Res* 33(11):3324–3343
2. Nunes L, Causer T, Ciolkosz D (2020) Biomass for energy: a review on supply chain management models. *Renew Sustain Energy Rev* 120:109658
3. Sasabe M, Hara T (2020) Capacitated shortest path tour problem-based integer linear programming for service chaining and function placement in NFV networks. *IEEE Trans Netw Service Manag* 18(1):104–117
4. Patle B, Pandey A, Parhi D, Jagadeesh A (2019) A review: On path planning strategies for navigation of mobile robot. *Defence Technol* 15(4):582–606
5. Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271
6. Floyd RW (1962) Algorithm 97: shortest path. *Commun ACM* 5(6):345
7. Gunantara N (2018) A review of multi-objective optimization: methods and its applications. *Cogent Eng* 5(1):1502242
8. Zhang T, Sun L, Yao L, Rong J (2017) Impact analysis of land use on traffic congestion using real-time traffic and POI. *Journal of Advanced Transportation* 2017
9. Zadeh LA (1996) Fuzzy sets. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh*. World Scientific, USA, pp 394–432

10. Zimmermann H-J (2011) *Fuzzy Set Theory-and Its Applications*. Springer, New York
11. Deng Y, Chan FT (2011) A new fuzzy dempster MCDM method and its application in supplier selection. *Expert Syst Appl* 38(8):9854–9861
12. Deng Y, Jiang W, Sadiq R (2011) Modeling contaminant intrusion in water distribution networks: a new similarity-based DST method. *Expert Syst Appl* 38(1):571–578
13. Yong D, Wenkang S, Feng D, Qi L (2004) A new similarity measure of generalized fuzzy numbers and its application to pattern recognition. *Pattern Recognit Lett* 25(8):875–883
14. Zero L, Bersani C, Paolucci M, Sacile R (2019) Two new approaches for the bi-objective shortest path with a fuzzy objective applied to HAZMAT transportation. *J Hazardous Materials* 375:96–106
15. Sawaragi Y, NAKAYAMA H, TANINO T (1985) *Theory of Multiobjective Optimization*. Elsevier, USA
16. Messac A, Sundararaj GJ, Tappeta RV, Renaud JE (2000) Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA J* 38(6):1084–1091
17. Messac A, Ismail-Yahaya A, Mattson CA (2003) The normalized normal constraint method for generating the Pareto frontier. *Struct Multidisciplinary Opt* 25(2):86–98
18. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
19. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
20. Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct Opt* 14(1):63–69
21. Slowik A, Kwasnicka H (2020) Evolutionary algorithms and their applications to engineering problems. *Neural Comput Appl* 32(16):12363–12379
22. Ulungu EL, Teghem J (1995) The two phases method: an efficient procedure to solve bi-objective combinatorial optimization problems. *Found Comput Decision Sci* 20(2):149–165
23. Guerriero F, Musmanno R (2001) Label correcting methods to solve multicriteria shortest path problems. *J Opt Theory Appl* 111(3):589–613
24. Hu X-B, Wang M, Di Paolo E (2013) Calculating complete and exact Pareto front for multiobjective optimization: a new deterministic approach for discrete problems. *IEEE Trans Cybernet* 43(3):1088–1101
25. Dubois DJ (1980) *Fuzzy sets and systems: theory and applications*, vol 144. Academic press, USA
26. Broumi S, Talea M, Bakali A, Smarandache F, Nagarajan D, Lathamaheswari M, Parimala M (2019) Shortest path problem in fuzzy, intuitionistic fuzzy and neutrosophic environment: an overview. *Complex Intell Syst* 5(4):371–378
27. Mahdavi I, Nourifar R, Heidarzade A, Amiri NM (2009) A dynamic programming approach for finding shortest chains in a fuzzy network. *Applied Soft Computing* 9(2):503–511
28. Deng Y, Chen Y, Zhang Y, Mahadevan S (2012) Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl Soft Comput* 12(3):1231–1237
29. Zhang Y, Zhang Z, Deng Y, Mahadevan S (2013) A biologically inspired solution for fuzzy shortest path problems. *Appl Soft Comput* 13(5):2356–2363
30. Lin L, Wu C, Ma L (2021) A genetic algorithm for the fuzzy shortest path problem in a fuzzy network. *Complex Intell Syst* 7(1):225–234
31. Kim Y-H, Kim J-H (2009) Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot. In: 2009 IEEE Congress on Evolutionary Computation, pp. 1185–1192 . IEEE
32. Yanyang W, Tietao W, Xiangju Q (2012) Study of multi-objective fuzzy optimization for path planning. *Chin J Aeronaut* 25(1):51–56
33. Adhikari D, Kim E, Reza H (2017) A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2258–2265 . IEEE
34. Abbaszadeh Sori A, Ebrahimnejad A, Motameni H (2020) The fuzzy inference approach to solve multi-objective constrained shortest path problem. *J Intell Fuzzy Syst* 38(4):4711–4720
35. Bagheri M, Ebrahimnejad A, Razavyan S, Lotfi FH, Malekmohammadi N (2021) Solving fuzzy multi-objective shortest path problem based on data envelopment analysis approach. *Complex Intell Syst* 7(2):725–740
36. Hu X-B, Wang M, Leeson MS, Di Paolo EA, Liu H (2016) Deterministic agent-based path optimization by mimicking the spreading of ripples. *Evol Comput* 24(2):319–346
37. Hu X-B, Zhang M-K, Zhang Q, Liao J-Q (2017) Co-evolutionary path optimization by ripple-spreading algorithm. *Trans Res Part B* 106:411–432
38. Hu X-B, Zhang C, Zhang G-P, Zhang M-K, Li H, Leeson MS, Liao J-Q (2020) Finding the k shortest paths by ripple-spreading algorithms. *Eng Appl Artificial Intell* 87:103229
39. Hu X-B, Gu S-H, Zhang C, Zhang G-P, Zhang M-K, Leeson MS (2021) Finding all Pareto optimal paths by simulating ripple relay race in multi-objective networks. *Swarm and Evolutionary Computation*, 100908
40. Barbati M, Bruno G, Genovese A (2012) Applications of agent-based models for optimization problems: a literature review. *Expert Syst Appl* 39(5):6020–6028
41. Chou C-C (2003) The canonical representation of multiplication operation on triangular fuzzy numbers. *Comput Math Appl* 45(10–11):1601–1610
42. Ganesan K, Veeramani P (2006) Fuzzy linear programs with trapezoidal fuzzy numbers. *Ann Oper Res* 143(1):305–315
43. Hassanzadeh R, Mahdavi I, Mahdavi-Amiri N, Tajdin A (2013) A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths. *Math Comput Modell* 57(1–2):84–99
44. Sadeghpour Gildeh B, Gien D (2001) La distance-dp, q et le coefficient de corrélation entre deux variables aléatoires floues. *Actes de LFA* 1:97–102
45. Chuang T-N, Kung J-Y (2005) The fuzzy shortest path length and the corresponding shortest path in a network. *Comput Oper Res* 32(6):1409–1428
46. Asady B, Zendehnam A (2007) Ranking fuzzy numbers by distance minimization. *Appl Math Modell* 31(11):2589–2598
47. Ezzati R, Allahviranloo T, Khezerloo S, Khezerloo M (2012) An approach for ranking of fuzzy numbers. *Expert Syst Appl* 39(1):690–695
48. Barr N (2020) *Economics of the welfare state*. Oxford University Press, USA
49. Paixão JM, Santos JL (2013) Labeling methods for the general case of the multi-objective shortest path problem—a computational study. *Comput Intell Decision Making* 61:489–502
50. Ahn CW, Ramakrishna RS (2002) A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans Evol Comput* 6(6):566–579

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.