This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Research on the Shortest Path Solution Method of Interval Valued Neutrosophic Graphs Based on the Ant Colony Algorithm

Lehua Yang[1], Dongmei Li[2], and Ruipu Tan[1]

[1]College of Electronics and Information Science, Fujian Jiangxia University, Fuzhou 350108, China
[2]College of Foreign Languages, Fujian Jiangxia University, Fuzhou 350108, China

Corresponding author: Dongmei Li (lidm358@163.com)

This study was supported by the National Social Science Foundation (17CGL058), Natural Science Foundation of Fujian Province (2018J01649), Educational and Scientific Research Project for Middle-Aged and Young Teachers in Fujian Province (JAT170624), Educational and Scientific Research Project for Middle-Aged and Young Teachers in Fujian Province (JAT170634), Educational and Scientific Research Project for Middle-Aged and Young Teachers in Fujian Province (JBS14221), and Planning Subject for the 13th Five-Year Plan of Fujian Province Education Sciences (FJJKCG17-200).

ABSTRACT The shortest path problem (SPP) is considerably important in several fields. After typhoons, the resulting damage leads to uncertainty regarding the path weight that can be expressed accurately. A neutrosophic set is a collection of the truth membership, indeterminacy membership, and falsity membership degrees of the elements. In an uncertain environment, neutrosophic numbers can express the edge distance more effectively. Based on the theories of interval valued neutrosophy and neutrosophic graphs, this paper proposes a shortest path solution method of interval valued neutrosophic graphs using the ant colony algorithm. Further, an analysis comparing the proposed algorithm with the Dijkstra algorithm was used to probe the potential shortcomings and advantages of the proposed method. In addition, this approach confirmed the effectiveness of the proposed algorithm. Furthermore, we investigated the convergence processes of the ant colony algorithm with different parameter settings, analyzed their results, and used different score functions to solve the SPP and analyze the results.

INDEX TERMS Ant colony algorithm, interval valued neutrosophic numbers, neutrosophic graph, shortest path problem

I. INTRODUCTION

In 1998, neutrosophy was introduced by Smarandache as a branch of philosophy that studies the nature, origin, and scope of neutrality and its interaction with various conceptual spectra [1]. Mumtaz promotes the concept of fuzzy sets [2] by adding an independent uncertain membership. Neutron collections are useful tools for dealing with inconsistent, uncertain, and incomplete information in real world and have become a concern for researchers. The concept of neutron subsets involves three degrees of independence, namely, true membership (T), uncertainty membership (I), and false membership (F). The value range is $\left]0^-,1^+\right[$. When the uncertainties of a vertex set and an edge set are obtained, a fuzzy graph can be used in the shortest path problem (SPP). However, if there is uncertainty in the relationship between nodes, the neutrosophic set (NS) theory will be a suitable concept for dealing with real-life problems [3], and the edge distance is regarded as the neutrosophic numbers (NN) for dealing with SPP. The NN can be single valued, interval valued, and/or bipolar [4], [5]. NN can handle uncertainty well, and the NS model can handle uncertain and inconsistent information. This is an important mechanism for dealing with practical scientific and engineering problems.

SPP is a basic combination problem that appears in various scientific and engineering fields. The purpose of the SPP is to find the shortest path and the minimum distance between the starting and end points in a graph [6, 7]. The length of a side can represent the actual amounts of the characteristics, such as time and cost. In the classical SPP,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

the distances between the edges of different nodes in a network graph are usually considered deterministic, and they are represented by exact numbers. However, for uncertain environments, fuzzy number techniques can be used.

As one of the countries in the world that has suffered the most from typhoons, China's coastal areas are often threatened with the loss of lives and property. The strong typhoon Lekima landed along the coast of Wenling City, Zhejiang Province, at 1:45 AM on August 10, 2019. The maximum wind speed near the center was at level 16 (52 m/s) at the time of landing, and the minimum air pressure was 930.100 Pa. It was the fifth strongest typhoon that has landed in Mainland China since 1949, affecting 403 counties in the nine provinces of Zhejiang, Shandong, Jiangsu, Anhui, Liaoning, Shanghai, Fujian, Hebei, and Jilin. When Lekima landed, it not only posed a large threat to peoples' lives and property but also caused severe storms and heavy rainfall that damaged roads and difficulties for post-disaster rescue workers. Yu et al. [8] applied the fuzzy multi-objective approach to forecast short-term typhoon rainfall, as this can be implemented without much background meteorological knowledge. Chen et al. [9] proposed an evolutionary fuzzy inference model that combines a fuzzy inference model, genetic programming, and a genetic algorithm to forecast flood stages during typhoons. Tan et al. [10] proposed a multi-attribute decision-making method based on prospect theory in a heterogeneous information environment. An illustrative example of typhoon disaster assessment is presented to show the feasibility and effectiveness of the proposed method, additionally, the advantages of the proposed method are illustrated by comparing it with other methods. SPPs of network graphs whose edges are represented as inaccurate numbers have attracted increasing attention from scholars worldwide. Post-disaster roads are flooded and blocked by trees and debris, and bridges are damaged; hence, the post-disaster rescue path has many uncertain variables that cannot be expressed with definite values. Therefore, Buckley and Jowers [11] introduced the concept of fuzzy logic into SPP. Deng et al. [12] proposed a fuzzy Dijkstra algorithm for SPP in inaccurate environments. Biswas et al. [13] introduced an algorithm for finding the shortest path in intuitionistic fuzzy environments.

The NS is an extension of the intuitionistic fuzzy set [14], which considers the uncertainty and describes the actual problem in more detail. More scholars have used NN to represent the edge weights in the SPP. Broumi et al. [3] used the Dijkstra algorithm to solve SPP within a background of neutrosophy. Broumi et al. [4] introduced SPP based on triangular fuzzy neutrosophic environments. Ye [15] proposed a single valued neutrosophic hesitant fuzzy set as, a further generalization of the concepts of fuzzy, intuitionistic fuzzy, single valued neutrosophic, hesitant fuzzy, and dual hesitant fuzzy sets. Peng [16] introduced operations of multi-valued NSs and developed a comparison method based on related research on hesitant fuzzy sets and intuitionistic fuzzy sets. Ye [17] proposed a trapezoidal NS, several operational rules, and score and accuracy functions for trapezoidal NN. Nancy and Harish [18] proposed an improved score function and applied it to the decision-making process. Ridvan et al. [19] developed a method for solving multiple attribute decision-making problems with single valued neutrosophic information or interval neutrosophic information. Deli et al. [20] developed hamming and Euclidean ranking values for the comparison of SVTN numbers. Broumi et al. [21] calculated MST in an interval valued bipolar NSting. Hu and Sotirov [22] proposed the convenience of quadratic SPP semi-definite programming and used branch and bound algorithms to solve SPPs. Zhang et al. [23] proposed a stable SPP with cyclic uncertainty. Broumi et al. [24] used SVNG to solve SPP. Peng and Dai [25] proposed an interval decision-making algorithm based on a neutrosophic environment. Smarandache [26] used trapezoidal fuzzy NN to find the shortest path. Wang et al. [27] proposed SV-trapezoidal neutrosophic preference in decision-making problems. Deli and Subas [28] put forward the ranking method of SVNN and applied it to decision-making problems. Broumi et al. [29] proposed various concepts on NS and analyzed the existing concepts and the proposed NN; thereafter, they proposed the SPP in an interval valued NS. Bolturk and Kahraman [31] proposed a new IVN analytic hierarchy process with a cosine similarity measure. Wang et al. [32] proposed interval NS and logic in detail. Biswas et al. [33] proposed distance measure using interval TrFNN. Deli [34] proposed a new notation called expansion and reduction of neutrosophic classical soft sets that are based on linguistic modifiers and then developed a neutrosophic classical soft reduction method and present a real example for the method. Deli [35] proposed single valued trapezoidal neutrosophic operators and applied them to decision-making problems. Deli and Suba [36] put forward the weighted geometric operator with the single valued neutrosophic method and applied it to decision-making problems. Basset et al. [37] proposed a decision-making technology that enables decision makers to select the most appropriate project in a neutrosophic environment. Additionally, they proposed a new technique for checking consistency and calculating the degree of consensus among experts' opinions in a neutrosophic environment. Kumar et al. [39] proposed an algorithm for solving SPP in a trapezoidal fuzzy neutrosophic environment. Broumi et al. [40] proposed a neutrosophic network method for finding the shortest path length with single valued trapezoidal NN. Tan et al. [41] proposed an extended dynamic programming method for solving the SPP to obtain the shortest path and the shortest path length. Broumi et al. [42] proposed a score

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

function for interval valued NN and used it to solve SPP. Moreover, they considered SPP with Bellman's algorithm for a network-using interval valued NN. Thereafter, Chakraborty [44] applied the developed score function and accuracy function of pentagonal NN to the SPP. Smarandache [45] proposed the theory that NeutroAlgebra is a generalization of partial algebra by introducing the indeterminate opposite. Schweizer [46] proposed uncertain factors that could be considered in the process of building the model and developed a formula to transform their model to the neutrosophic representation. Edalatpanah [47] proposed a direct algorithm to solve the neutrosophic linear programming, where the variables and right-and side were represented by triangular NN.

The ant colony algorithm is a probabilistic algorithm used to find the optimal path. This algorithm has the characteristics of distributed computing, positive information feedback, and heuristic searching. It is essentially a heuristic global optimization algorithm in evolutionary algorithms. Further, the ant colony algorithm has been widely used in industry, agriculture, and other fields [48]. However, at present, there are few literature reports on the use of the ant colony algorithm to find the shortest path in a neutrosophic graph. Therefore, this paper describes the shortest path solution method based on the ant colony algorithm with an interval valued neutrosophic graph. This paper studies the neutrosophic graph with side weights represented by interval valued NN.

II. THEORETICAL BASIS

A brief description of some basic concepts of NSs and some existing ranking functions for interval valued NN are given in the following subsections.

A. NEUTROSOPHIC SET

Definition 1 [1]: Let X be the object set and x be any one of the objects, and NS A on X can be represented by the true degree function $T_A(x)$, the uncertainty degree function $I_A(x)$, and the error degree function $F_A(x)$, where $T_A(x)$, $I_A(x)$, and $F_A(x)$ are the standard or non-standard real subsets of $]0^-,1^+[$, i.e., $T_A(x): x \to]0-,1+[$, $I_A(x): x \to]0-,1+[$, $F_A(x): x \to]0-,1+[$

where the non-standard finite number $1^+ = 1+\varepsilon$. "1" is its standard part, $\varepsilon > 0$ is an infinite decimal (which is its non-standard part), and $0^- \le \sup T_A(x) + \sup I_A(x) + \sup F_A(x) \le 3^+$.

B. INTERVAL VALUED NEUTROSOPHIC NUMBER

Definition 2 [15]: X is a domain, and a single valued NN $\widetilde{N}$ can be expressed as follows:

$$\widetilde{N} = \left\{\left\langle x, T_{\widetilde{N}}(x), I_{\widetilde{N}}(x), F_{\widetilde{N}}(x)\right\rangle \mid x \in X\right\} \tag{1}$$

where $T_{\widetilde{N}}(x) \in [0,1]$, $I_{\widetilde{N}}(x) \in [0,1]$ and $F_{\widetilde{N}}(x) \in [0,1]$.

Definition 3 [32]: X is a domain and an interval valued NN $\widetilde{N}$ can be expressed as follows:

$$\widetilde{N} = \left\{\left\langle x, T_{\widetilde{N}}(x), I_{\widetilde{N}}(x), F_{\widetilde{N}}(x)\right\rangle \mid x \in X\right\} \tag{2}$$

where $T_{\widetilde{N}}(x) \subseteq [0,1]$, $I_{\widetilde{N}}(x) \subseteq [0,1]$ and $F_{\widetilde{N}}(x) \subseteq [0,1]$ are interval valued numbers,

$T_{\widetilde{N}}(x) = \left[T^L_{\widetilde{N}}(x), T^U_{\widetilde{N}}(x)\right]$,

where $0 \le T^L_{\widetilde{N}}(x) \prec T^U_{\widetilde{N}}(x) \le 1$.

$I_{\widetilde{N}}(x) = \left[I^L_{\widetilde{N}}(x), I^U_{\widetilde{N}}(x)\right]$,

where $0 \le I^L_{\widetilde{N}}(x) \prec I^U_{\widetilde{N}}(x) \le 1$.

$F_{\widetilde{N}}(x) = \left[F^L_{\widetilde{N}}(x), F^U_{\widetilde{N}}(x)\right]$,

where $0 \le F^L_{\widetilde{N}}(x) \prec F^U_{\widetilde{N}}(x) \le 1$.

Definition 4 [32]: $\widetilde{N}_1 = \langle [a_1, a_2], [b_1, b_2], [c_1, c_2] \rangle$ and $\widetilde{N}_2 = \langle [d_1, d_2], [e_1, e_2], [f_1, f_2] \rangle$ are two interval valued NN, and then

$$(1)\ \widetilde{N}_1 \oplus \widetilde{N}_2 = \left\langle \begin{matrix} [a_1 + d_1 - a_1 d_1, a_2 + d_2 - a_2 d_2], \\ [b_1 e_1, b_2 e_2], \\ [c_1 f_1, c_2 f_2] \end{matrix} \right\rangle \tag{3}$$

$$(2)\ \widetilde{N}_1 \otimes \widetilde{N}_2 = \left\langle \begin{matrix} [a_1 d_1, a_2 d_2], \\ [b_1 + e_1 - b_1 e_1, b_2 + e_2 - b_2 e_2], \\ [c_1 + f_1 - c_1 f_1, c_2 + f_2 - c_2 f_2] \end{matrix} \right\rangle \tag{4}$$

$$(3)\ \lambda \widetilde{N}_1 = \left\langle \begin{matrix} [1 - (1 - a_1)^\lambda, 1 - (1 - a_2)^\lambda], \\ [b_1^{\ \lambda}, b_2^{\ \lambda}], \\ [c_1^{\ \lambda}, c_2^{\ \lambda}] \end{matrix} \right\rangle \tag{5}$$

$$(4)\ \tilde{N}_1^{\lambda} = \left\langle \begin{matrix} \left[a_1^{\lambda}, a_2^{\lambda}\right], \\ \left[1-(1-b_1)^{\lambda}, 1-(1-b_2)^{\lambda}\right], \\ \left[1-(1-c_1)^{\lambda}, 1-(1-c_2)^{\lambda}\right] \end{matrix} \right\rangle \tag{6}$$

C. RANKING FUNCTION

Definition 5 [19]: $\tilde{N} = \langle [a_1, a_2], [b_1, b_2], [c_1, c_2] \rangle$ is an interval valued NN, and its score function can be expressed as

$$S(\tilde{N}) = \frac{1}{4} \times (2 + a_1 + a_2 - 2b_1 - 2b_2 - c_1 - c_2) \tag{7}$$

The larger the $S(\tilde{N})$ value, the larger the interval valued NN $\tilde{N}$ will be.

Definition 6 [42]: $\tilde{N}_1 = \langle [a_1, a_2], [b_1, b_2], [c_1, c_2] \rangle$ and $\tilde{N}_2 = \langle [e_1, e_2], [f_1, f_2], [g_1, g_2] \rangle$ are two interval valued NN, and $S(\tilde{N}_1)$ and $S(\tilde{N}_2)$ are the score functions of $\tilde{N}_1$ and $\tilde{N}_2$, respectively. The ordering relationship of interval valued NN are as follows:

if $S(\tilde{N}_1) \succ S(\tilde{N}_2)$, then $\tilde{N}_1 \succ \tilde{N}_2$;

if $S(\tilde{N}_1) \prec S(\tilde{N}_2)$, then $\tilde{N}_1 \prec \tilde{N}_2$;

if $S(\tilde{N}_1) = S(\tilde{N}_2)$, then $\tilde{N}_1 = \tilde{N}_2$.

III. SPP IN EMERGENCY DECISION MAKING

When a disaster occurs, it is very important to rescue the victims. The most apparent feature of emergency decision-making is the urgency of time. The decision maker should complete the rescue plan in a short time, thereby enabling the rescuers to reach the location of the trapped people immediately. The time needed to arrive at the rescue site often determines the success or failure of the rescue mission; hence, the earliest possible arrival time is considered as the main objective function. When the police and rescue team have fixed arrival times, the shortest rescue time can be simplified as the shortest transportation time and further as the shortest path desired. For other boundary conditions, such as road water accumulation, bridge damage, and possible impact of landslides on roadside mountains, the grade interval can be determined according to the degree of its impact on the material transportation, and the path weight can be described by the interval valued NN. The traditional SPP is a combinatorial optimization problem. The challenge of finding the shortest path primarily includes the single source SPP and the multiple SPP. The single source SPP involves finding the shortest path from a given point to other points in the graph. The multiple SPP refers to finding the shortest paths between all vertex pairs in the graph. The current solution to this type of problem is primarily based on the classical Dijkstra algorithm and the Floyd algorithm. In practical applications, the expression form of an urban road network is generally a digital vector map. To analyze the shortest path efficiently, it must be abstracted as the structure of the map relative to the relationship between nodes and arcs. When a disaster occurs, the rapid determination of the shortest driving route to the disaster location is a key issue in emergency decision-making. Efficient implementation can improve the rapid response ability and overall command ability of the rescue. The shortest path algorithm discussed here is primarily aimed at directed graphs, for which the weights of all arcs are interval valued NN.

IV. NEUTROSOPHIC GRAPH THEORY

In an emergency rescue, the site of each rescue team and the point of the rescue can be abstracted as the vertices of graph theory. A disaster area is composed of N rescue team locations, rescue points, and passing points. Its network topology can be abstracted as a directed graph marked as $G(V,E)$, where point set $V = \{v_1, \cdots, v_n\}$ represents the rescue team locations, rescue points, and passing points, edge set $E \subseteq V \times V$ represents the directed connection between two points, and a directed edge $(i, j) \in E$ represents the path from point i to point j [41]. To reach the optimal decision in the shortest time, it is necessary to consider the path length, the road conditions of the path, the transportation capacity, etc. Considering the geographical location, terrain, the degree of disaster damage, and other factors, the edge weight is expressed as an interval valued NN $\tilde{N}$, node v_1 is the starting point of the rescue, and node v_n is the point of the rescue site. A directed path from node v_1 to node v_n can be expressed as a set of directed edge sequences in the form of $(v_1, v_2), (v_2, v_3), \cdots, (v_k, v_n)$ in a directed graph. Depending on the connection strength of a directed graph, the number of paths connecting node v_1 to node v_n may vary.

V. METHOD FOR SOLVING SPP OF INTERVAL VALUED NEUTROSOPHIC GRAPH BASED ON ANT COLONY ALGORITHM

The ant colony algorithm is a probabilistic algorithm used to find the optimal path. It was proposed by Marco Dorigo in his doctoral dissertation in 1992 [48], and the basic idea comes from the behavior of ants as they find a path while searching for food. Ants find the shortest path using factors such as pheromones and the environment. Suppose there are two paths from an ant colony to food. Initially, the number of ants on the two paths is similar; when ants arrive at the end of the road, they will return immediately. Ants on the short path have a short round-trip time and fast repetition rate; hence, there are more ants per unit time, leaving more pheromones. This will attract even more ants that will leave more pheromones. However, the long path exhibits the opposite characteristics; hence, an increasing number of ants will gather on the shorter path. In the simulation of pheromone effects, several paths are randomly selected each time, the pheromone level on each path is adjusted according to the weight, and the shortest path is realized after iterative convergence. The specific steps are as follows:

Step 1: Initialize the pheromone matrix and uniformly set the pheromone concentration of each path from the starting point to the end point to c .

Step 2: Randomly select n paths, calculate the weight of each path according to the score function calculation method of *Definition 5* (above), and then calculate the smallest $P_{\min_{[0]}}$ of N paths according to the intelligent number-sorting algorithm in the *Definition 6* interval.

Step 3: Update pheromone levels:

1) The concentration of all pheromones was reduced by Δp , which was used to simulate the volatilization of pheromones.

2) The pheromone concentration of a path $P_{\min_{[0]}}$ increases by Δq .

Step 4: Determine the ratio $\mathrm{per}_{(p_{\min[0]})} = \dfrac{c_{(p_{\min[0]})}}{\sum_{i=0}^{n} c_i}$ of the pheromone level of path $P_{\min_{[0]}}$ to the total pheromone level, calculate the times $N_{p_{\min[0]}} = \mathrm{round}(N^* \mathrm{per}_{(p_{\min[0]})})$ of the next path $P_{\min_{[0]}}$, randomly select other $N - N_{p_{\min[0]}}$ paths, calculate the weight of each path according to the method of the *Definition 5* score function, and then calculate the smallest $P_{\min_{[1]}}$ of the N paths according to the Ranking function in *Definition 6*. Subsequently, repeat Step 3 to update the pheromone level and randomly select n paths, including $P_{\min_{[1]}}$. Eventually, $S(P_{\min_{[i]}})$ converges to a certain number and the algorithm ends.

Moreover, iteration produces a local optimal solution. By adding the pheromone level of the local optimal solution and modifying the path selected each time, the local optimal solution gradually tends to the global optimal solution. When $N_{p_{\min[i]}} = N$, the local optimal solution converges, and the algorithm flow chart is as shown in Fig. 1.

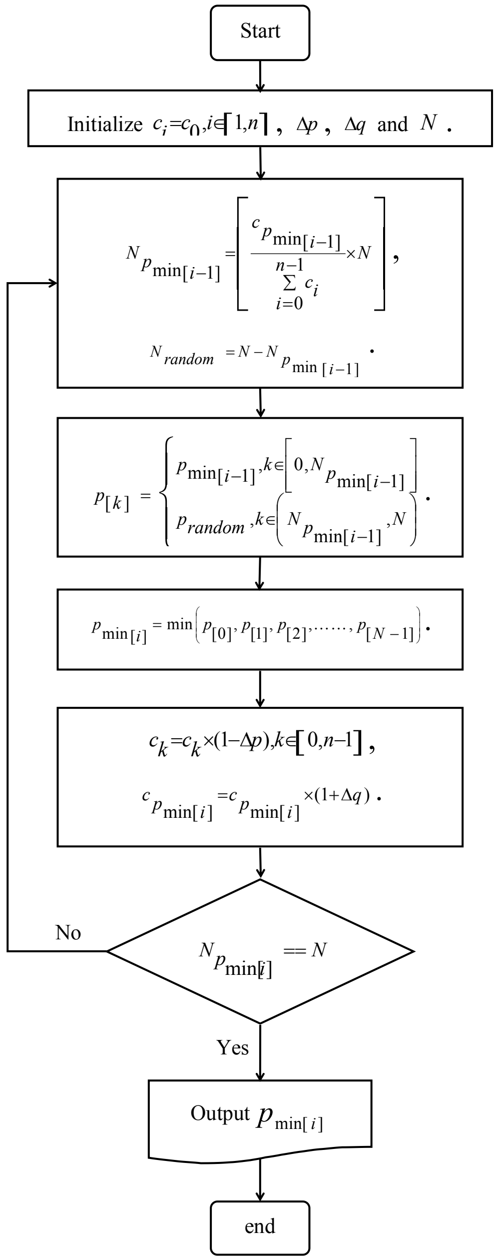

FIGURE 1. Flow chart of the ant colony algorithm.

VI. CASE STUDY AND COMPARATIVE ANALYSIS

A. CASE ANALYSIS

The powerful typhoon Lekima landed on the coast of Wenling City, Zhejiang Province, at 1:45 AM on August 10, 2019. During its landfall, the maximum wind force near the center was level 16 (52 m/s), and the minimum air pressure at the center was 930 hPa. After the disaster, roads were flooded and blocked by trees and rocks, bridges were damaged, etc., thereby making rescue work difficult. Given the road conditions, it was essential to find the best path to the rescue point and provide decision-making support to the emergency rescue teams of relevant departments. The topological structure of the road network during the period is shown in Fig. 2, and the side lengths involved are shown in Table I. A rescue team in Fuzhou must start from point ① and travel to point ⑨ to rescue trapped residents; hence, they must identify the shortest path from the starting point ① to the ending point ⑨, and the sequence is outlined below:

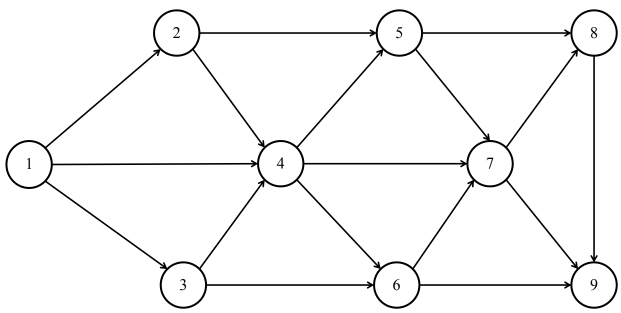

FIGURE 2. Interval valued neutrosophic graph.

TABLE I

DETAILS OF EDGES INFORMATION IN TERMS OF INTERVAL VALUED NN

Edges	Interval valued neutrosophic distance
(1,2)	<[0.1, 0.5], [0.3, 0.7], [0.1, 0.6]>
(1,3)	<[0.2, 0.7], [0.3, 0.9], [0.1, 0.4]>
(1,4)	<[0.2, 0.6], [0.2, 0.6], [0.4, 0.8]>
(2,4)	<[0.1, 0.5], [0.3, 0.7], [0.2, 0.7]>
(2,5)	<[0.4, 0.8], [0.2, 0.6], [0.1, 0.5]>
(3,4)	<[0.3, 0.8], [0.1, 0.4], [0.1, 0.6]>
(3,6)	<[0.1, 0.4], [0.2, 0.6], [0.4, 0.7]>
(4,5)	<[0.2, 0.5], [0.3, 0.6], [0.1, 0.6]>
(4,6)	<[0.4, 0.9], [0.1, 0.5], [0.1, 0.6]>
(4,7)	<[0.1, 0.6], [0.2, 0.7], [0.4, 0.8]>
(5,7)	<[0.2, 0.6], [0.2, 0.7], [0.4, 0.8]>
(5,8)	<[0.3, 0.7], [0.2, 0.6], [0.1, 0.5]>
(6,7)	<[0.3, 0.7], [0.1, 0.5], [0.3, 0.9]>
(6,9)	<[0.2, 0.6], [0.2, 0.5], [0.5, 0.9]>
(7,8)	<[0.1, 0.5], [0.2, 0.6], [0.4, 0.8]>
(7,9)	<[0.4, 0.9], [0.2, 0.6], [0.1, 0.5]>
(8,9)	<[0.2, 0.6], [0.1, 0.5], [0.5, 0.8]>

The pheromone volatility, p, indicates the degree of pheromone volatility remaining on the path as the ant walks back and forth. The value range is $(0,1)$. The size of p is directly related to the global search ability and convergence speed of the ant colony algorithm. When p is large, the pheromone volatilization increases in speed and is not very sensitive to past historical experience. Positive feedback plays a dominant role in this process, and the randomness of the search weakens, which leads to a fast convergence rate but can easily fall into the local optimal. However, when p is small, the residual information on the path dominates, the positive feedback of the information is relatively weak, the randomness of the search is enhanced, and the convergence speed of the ant colony algorithm is slow.

Moreover, the locally optimal pheromone level increases, and q represents the increment of pheromone when all the ants walk back and forth on the shortest path. However, to a certain extent, q affects the convergence speed of the algorithm. Thus, the larger q is, the more are the pheromones remaining each time an ant passes, and the faster is the accumulation of pheromones on the traversed path, which enhances the positive feedback during the ant colony search and contributes to the rapid convergence of the algorithm; however, it is easy to fall into the local optimal.

In the calculation example, the principle of giving priority to accuracy is considered, and the pheromone

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

volatility, $\Delta p = 5\%$, and locally optimal pheromone level increases, $\Delta q = 10\%$, are selected.

Step 1: Initialize the pheromone matrix $C_i = 1, i \in [1,85]$ and randomly select the path number $N = 10$ per iteration. The pheromone volatility is $\Delta p = 5\%$, and the locally optimal pheromone level increases by $\Delta q = 10\%$.

Step 2: Randomly select 10 paths from the starting point ① to the ending point ⑨ – $P_{(125789)}$, $P_{(1246789)}$, $P_{(13479)}$, $P_{(134579)}$, $P_{(145789)}$, $P_{(14579)}$, $P_{(146789)}$, $P_{(125789)}$, $P_{(134789)}$ and $P_{(134579)}$ – and then calculate the interval valued neutrosophic distance of $P_{(125789)}$ according to *Definition 4*:

$$
\begin{aligned}
&N_{(125789)} = N_{(12)} \oplus N_{(25)} \oplus N_{(57)} \oplus N_{(78)} \oplus N_{(89)} \\
&=< [0.1, 0.5], [0.3, 0.7], [0.1, 0.6] > \\
&\oplus < [0.4, 0.8], [0.2, 0.6], [0.1, 0.5] > \\
&\oplus < [0.2, 0.6], [0.2, 0.7], [0.4, 0.8] > \\
&\oplus < [0.1, 0.5], [0.2, 0.6], [0.4, 0.8] > \\
&\oplus < [0.2, 0.6], [0.1, 0.5], [0.5, 0.8] > \\
&= \left\langle \begin{matrix} [0.68900, 0.99201], \\ [0.00025, 0.08821], \\ [0.00081, 0.15362] \end{matrix} \right\rangle
\end{aligned}
$$

Similarly, the length of the other nine paths can be calculated, as shown in Table II.

TABLE II

DETAILS OF PATH INFORMATION IN TERMS OF INTERVAL VALUED NN

Paths	Interval valued neutrosophic distance
$P_{(125789)}$	<[0.68900, 0.99201], [0.00025, 0.08821], [0.00081, 0.15362]>
$P_{(1246789)}$	<[0.75508, 0.99851], [0.00002, 0.03675], [0.00013, 0.14517]>
$P_{(13479)}$	<[0.69762, 0.99761], [0.00121, 0.15121], [000041, 0.09601]>
$P_{(134579)}$	<[0.78499, 0.99881], [0.00037, 0.09073], [0.00005, 0.05761]>
$P_{(145789)}$	<[0.63139, 0.98401], [0.00025, 0.07561], [0.00321, 0.24578]>
$P_{(14579)}$	<[0.69283, 0.99201], [0.00241, 0.15121], [0.00161, 0.19201]>
$P_{(146789)}$	<[0.75810, 0.99761], [0.00005, 0.04500], [0.00241, 0.27650]>
$P_{(125789)}$	<[0.68900, 0.99201], [0.00025, 0.08821], [0.00081, 0.15362]>
$P_{(134789)}$	<[0.63715, 0.99521], [0.00013, 0.07561], [0.0008.1, 0.12290]>
$P_{(134579)}$	<[0.78498, 0.99881], [0.00037, 0.09074], [0.00005, 0.05761]>

Calculate the score function of path $P_{(125789)}$ according to *Definition 5*:

$$
\begin{aligned}
&S(\tilde{N}_{(125789)}) \\
&= \frac{1}{4} \times (2 + 0.68900 + 0.99201 - 2 \times 0.00025 \\
&- 2 \times 0.08821 - 0.00081 - 0.15362) \\
&= 0.83742
\end{aligned}
$$

The score function of the other nine paths can be calculated similarly, and results are shown in Table III.

TABLE III

SCORE FUNCTION OF 10 PATHS

Paths	Score function of paths
$P_{(125789)}$	0.83742
$P_{(1246789)}$	0.88369
$P_{(13479)}$	0.82350
$P_{(134579)}$	0.88599
$P_{(145789)}$	0.80368
$P_{(14579)}$	0.79600
$P_{(146789)}$	0.84668
$P_{(125789)}$	0.83742
$P_{(134789)}$	0.83930
$P_{(134579)}$	0.88598

According to *Definition 6*, the shortest path among the first randomly generated 10 paths can be found: $P_{\min[0]} = P_{(145789)}$.

Step 3: Update pheromone levels:

1) The concentration of all pheromones was reduced by 5%; $C_i = 1*(1-0.05) = 0.95, i \in [1,85]$.

2) The pheromone concentration of path $P_{\min_{[0]}}$ increases by 10%, $C_{(145789)} = 0.95*1.1 = 1.045$.

Step 4: Determine the ratio $\mathrm{per}_{(p_{\min[0]})} = \frac{c_{(p_{\min[0]})}}{\sum_{i=0}^{n} c_i} = \frac{1.045}{0.95*84+1.045} = 0.013$ of the pheromone level of path $P_{\min_{[0]}}$ to the total pheromone level and calculate the product $N_{p_{\min[0]}} = \mathrm{round}(N * \mathrm{per}_{(p_{\min[0]})}) = \mathrm{round}(10 * 0.013) = 0$ of the next path $P_{\min_{[0]}}$; thus, it is not necessary to specify the selection path $P_{(145789)}$ but to randomly select another 10 paths. Further, repeat Steps 2 and 3, continuously obtain the local optimal solution, update the pheromone level, continue up to the 220th cycle, and obtain the local optimal solution $P_{\min[220]} = P_{(1369)}$, the updated pheromone level $c_{(1369)} = 0.02831$, the total pheromone level, and $\sum_{i=0}^{n} c_i = 0.02972$ of all paths. Thereafter, determine the ratio of the pheromone level of path $P_{\min[220]}$ to the total pheromone level as $\mathrm{per}_{(p_{\min[220]})} = \frac{0.02831}{0.02972} = 0.95255$, and select path $P_{\min[220]}$ for the next cycle. The number of cycles $N_{p_{\min[220]}} = \mathrm{round}(N * \mathrm{per}_{(p_{\min[220]})}) = \mathrm{round}(10*0.95255) = 10$ that is, from the 221st cycle on, all the 10 paths selected are $P_{(1369)}$, after which there will be no new local optimal solution, the optimal solution converges to $P_{(1369)}$, the score function converges to 0.629 and does not change, and the algorithm ends. The information for the local optimal solution generated during iteration is shown in Table IV.

TABLE IV

LOCAL OPTIMAL SOLUTION PRODUCED BY EACH ITERATION

Cycles	Minpath	Score function	Pheromone	N(min)
1	$p_{(12469)}$	0.78120	1.04500	0
2	$p_{(12479)}$	0.74125	0.99275	0
3	$p_{(14789)}$	0.68860	0.94311	0
4	$p_{(12479)}$	0.74125	0.98555	0
5	$p_{(124789)}$	0.75800	0.85116	0
⋮	⋮	⋮	⋮	⋮
155	$p_{(14789)}$	0.68860	0.00134	0
156	$p_{(14789)}$	0.68860	0.00273	1
157	$p_{(14789)}$	0.68860	0.00285	1
158	$p_{(1369)}$	0.62900	0.00185	1
159	$p_{(1369)}$	0.62900	0.00193	1
⋮	⋮	⋮	⋮	⋮
216	$p_{(1369)}$	0.62900	0.02374	9
217	$p_{(1369)}$	0.62900	0.02481	9
218	$p_{(1369)}$	0.62900	0.02593	9
219	$p_{(1369)}$	0.62900	0.02709	9
220	$p_{(1369)}$	0.62900	0.02831	10

In Table IV, "Cycles" represents the number of iterations, "Minpath" represents the local shortest path generated during iteration, "Score function" represents the score function of the local shortest path, "Pheromone" represents the pheromone concentration after updating the local shortest path after a completed iteration, and "N (min)" represents the number of Minpath times in the next iteration. The data show that from the 220th iteration onwards, $P_{(1369)}$ is selected 10 times and no new optimal solution can appear in the subsequent iterations. The convergence process of the local optimal solution is shown in Fig. 3.

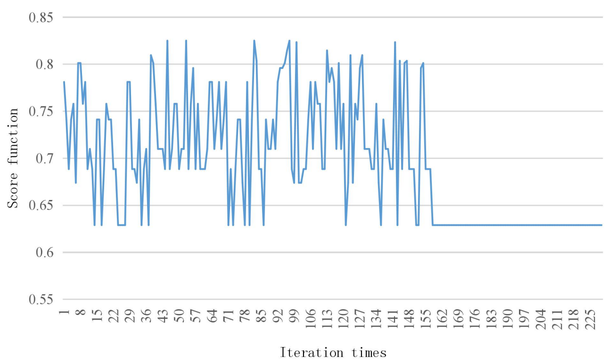

FIGURE 3. **Convergence process of local optimal solution.**

The abscissa in Fig. 3 refers to the number of iterations, and the ordinate refers to the score function of the local optimal solution obtained during iteration. Finally, the shortest path score function from the starting point ① to the ending point ⑨ is 0.629, and the shortest path is ① -> ③ -> ⑥ -> ⑨, as shown in Fig. 4.

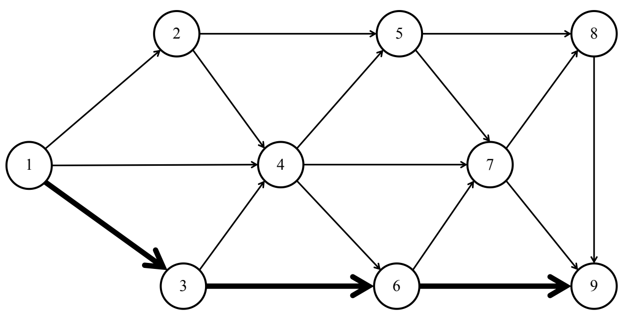

FIGURE 4. **Shortest path from starting point ① to ending point ⑨.**

B. COMPARATIVE ANALYSIS OF DIFFERENT ALGORITHMS

To illustrate the effectiveness and rationality of the algorithm, the ant colony algorithm proposed in this paper is compared with the Dijkstra algorithm proposed in reference [40] and the Bellman algorithm proposed in reference [43].

1) COMPARISON AND ANALYSIS WITH THE DIJKSTRA ALGORITHM

First, the adjacency matrix is established according to Table I:

$$A=\begin{bmatrix} 0 & \tilde{n}_{(1,2)} & \tilde{n}_{(1,3)} & \tilde{n}_{(1,4)} & M & M & M & M & M \\ M & 0 & M & \tilde{n}_{(2,4)} & \tilde{n}_{(2,5)} & M & M & M & M \\ M & M & 0 & \tilde{n}_{(3,4)} & M & \tilde{n}_{(3,6)} & M & M & M \\ M & M & M & 0 & \tilde{n}_{(4,5)} & \tilde{n}_{(4,6)} & \tilde{n}_{(4,7)} & M & M \\ M & M & M & M & 0 & M & \tilde{n}_{(5,7)} & \tilde{n}_{(5,8)} & M \\ M & M & M & M & M & 0 & \tilde{n}_{(6,7)} & M & \tilde{n}_{(6,9)} \\ M & M & M & M & M & M & 0 & \tilde{n}_{(7,8)} & \tilde{n}_{(7,9)} \\ M & M & M & M & M & M & M & 0 & \tilde{n}_{(8,9)} \\ M & M & M & M & M & M & M & M & 0 \end{bmatrix}$$

where $\tilde{n}_{(i,j)}$ represents the edge weight from node i to node j. The specific values are shown in Table I, where M represents an infinite number, indicating that there is no direct directed edge connection between node i and node j. The Dijkstra algorithm is used to calculate the shortest path from node ① to node ⑨, according to the following:

Step 1: Set $V=\{1,2,3,4,5,6,7,8,9\}$ of all vertices, set $S=\{1\}$ of initially determined vertices, set $L=\varnothing$ of initially determined edges, and set $T=V-S=\{2,3,4,5,6,7,8,9\}$ of undetermined vertices.

Step 2: Find the closest vertex to all elements in set S; that is, find the shortest edge among edges (1, 2), (1, 3) and (1, 4). According to *Definition 5*, the score functions of the three edges can be calculated as follows:

$$S(\tilde{N}_{(12)})$$
$$=\frac{1}{4}\times(2+0.1+0.5-2\times0.3-2\times0.7-0.1-0.6)$$
$$=-0.025$$

$$S(\tilde{N}_{(13)})$$
$$=\frac{1}{4}\times(2+0.2+0.7-2\times0.3-2\times0.9-0.1-0.4),$$
$$=0$$

$$S(\tilde{N}_{(14)})$$
$$=\frac{1}{4}\times(2+0.2+0.6-2\times0.2-2\times0.6-0.4-0.8)$$
$$=0$$

According to *Definition 6*, we can obtain $\min(\tilde{n}_{(1,2)},\tilde{n}_{(1,3)},\tilde{n}_{(1,4)},M,M,M,M,M)=\tilde{n}_{(1,2)}$; the nearest vertex is ②; hence, update vertices set $S=\{1,2\}$ and edge set $L=\{(1,2)\}$. Repeat Step 2 to update vertices set S and edge set L until the end vertices $9 \in S$. Finally, $S=\{1,2,4,3,7,6,5,8,9\}$, $L=\{(1,2),(1,4),(1,3),(4,7),(3,6),(4,5),(7,8),(6,9)\}$. The shortest path is ①->③->⑥->⑨, as shown in Fig. 5.

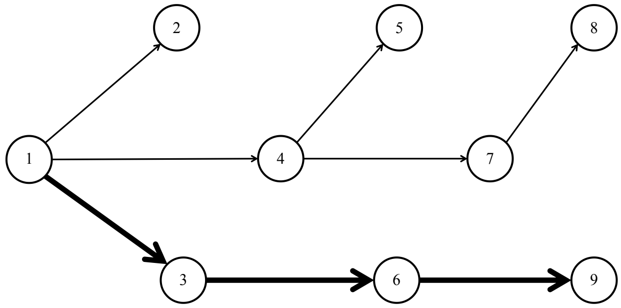

FIGURE 5. Shortest path calculated by the Dijkstra algorithm.

The result of the ant colony algorithm proposed in this paper is the same as that of the Dijkstra algorithm, and they have the same shortest path and path length.

2) COMPARISON AND ANALYSIS WITH THE BELLMAN ALGORITHM

$f(i)$ is the shortest distance between the starting point ① and point i, and $\tilde{N}_{(ij)}$ is the distance between node i and node j.

$$f(1)=0,$$

$$\begin{aligned}&f(2)=\min_{i<2}\{f(i)\oplus\tilde{N}_{(i2)}\}=\min\{f(1)\oplus\tilde{N}_{(12)}\}\\&=\min\{0\oplus\langle[0.1,0.5],[0.3,0.7],[0.1,0.6]\rangle\}\\&=\langle[0.1,0.5],[0.3,0.7],[0.1,0.6]\rangle\end{aligned},$$

$$\begin{aligned}&f(3)=\min_{i<3}\{f(i)\oplus\tilde{N}_{(i3)}\}=\min\{f(1)\oplus\tilde{N}_{(13)}\}\\&=\min\{0\oplus\langle[0.2,0.7],[0.3,0.9],[0.1,0.4]\rangle\}\\&=\langle[0.2,0.7],[0.3,0.9],[0.1,0.4]\rangle\end{aligned},$$

$$\begin{aligned}&f(4)=\min_{i<4}\{f(i)\oplus\tilde{N}_{(i4)}\}\\&=\min\{f(1)\oplus\tilde{N}_{(14)},f(2)\oplus\tilde{N}_{(24)},f(3)\oplus\tilde{N}_{(34)}\}\\&=\min\left\{\begin{array}{l}0\oplus\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle,\\\langle[0.1,0.5],[0.3,0.7],[0.1,0.6]\rangle\\\oplus\langle[0.1,0.5],[0.3,0.7],[0.2,0.7]\rangle,\\\langle[0.2,0.7],[0.3,0.9],[0.1,0.4]\rangle\\\oplus\langle[0.3,0.8],[0.1,0.4],[0.1,0.6]\rangle\end{array}\right\}\\&=\min\left\{\begin{array}{l}\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle,\\\langle[0.19,0.75],[0.09,0.49],[0.02,0.42]\rangle,\\\langle[0.44,0.94],[0.03,0.36],[0.01,0.24]\rangle\end{array}\right\}\\&=\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle\end{aligned},$$

$$\begin{aligned}&f(5)=\min_{i<5}\{f(i)\oplus\tilde{N}_{(i5)}\}\\&=\min\{f(2)\oplus\tilde{N}_{(25)},f(4)\oplus\tilde{N}_{(45)}\}\\&=\min\left\{\begin{array}{l}\langle[0.1,0.5],[0.3,0.7],[0.1,0.6]\rangle\\\oplus\langle[0.4,0.8],[0.2,0.6],[0.1,0.5]\rangle,\\\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle\\\oplus\langle[0.2,0.5],[0.3,0.6],[0.1,0.6]\rangle\end{array}\right\}\\&=\min\left\{\begin{array}{l}\langle[0.46,0.9],[0.06,0.42],[0.01,0.3]\rangle,\\\langle[0.36,0.8],[0.06,0.36],[0.04,0.48]\rangle\end{array}\right\}\\&=\langle[0.46,0.9],[0.06,0.42],[0.01,0.3]\rangle\end{aligned},$$

$$\begin{aligned}&f(6)=\min_{i<6}\{f(i)\oplus\tilde{N}_{(i6)}\}\\&=\min\{f(3)\oplus\tilde{N}_{(36)},f(4)\oplus\tilde{N}_{(46)}\}\\&=\min\left\{\begin{array}{l}\langle[0.2,0.7],[0.3,0.9],[0.1,0.4]\rangle\\\oplus\langle[0.1,0.4],[0.2,0.6],[0.4,0.7]\rangle,\\\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle\\\oplus\langle[0.4,0.9],[0.1,0.5],[0.1,0.6]\rangle\end{array}\right\}\\&=\min\left\{\begin{array}{l}\langle[0.28,0.82],[0.06,0.54],[0.04,0.28]\rangle,\\\langle[0.52,0.96],[0.02,0.3],[0.04,0.48]\rangle\end{array}\right\}\\&=\langle[0.28,0.82],[0.06,0.54],[0.04,0.28]\rangle\end{aligned},$$

$$f(7)=\min_{i<7}\left\{f(i)\oplus\widetilde{N}_{(i7)}\right\}$$
$$=\min\left\{f(4)\oplus\widetilde{N}_{(47)},f(5)\oplus\widetilde{N}_{(57)},f(6)\oplus\widetilde{N}_{(67)}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.2,0.6],[0.2,0.6],[0.4,0.8]\rangle\\ \oplus\langle[0.1,0.6],[0.2,0.7],[0.4,0.8]\rangle,\\ \langle[0.46,0.9],[0.06,0.42],[0.01,0.3]\rangle\\ \oplus\langle[0.2,0.6],[0.2,0.7],[0.4,0.8]\rangle,\\ \langle[0.28,0.82],[0.06,0.54],[0.04,0.28]\rangle\\ \oplus\langle[0.3,0.7],[0.1,0.5],[0.3,0.9]\rangle\end{array}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.28,0.84],[0.04,0.42],[0.16,0.64]\rangle,\\ \langle[0.568,0.96],[0.012,0.294],[0.004,0.24]\rangle,\\ \langle[0.496,0.946],[0.006,0.27],[0.012,0.252]\rangle\end{array}\right\}$$
$$=\langle[0.28,0.84],[0.04,0.42],[0.16,0.64]\rangle$$
,

$$f(8)=\min_{i<8}\left\{f(i)\oplus\widetilde{N}_{(i8)}\right\}$$
$$=\min\left\{f(5)\oplus\widetilde{N}_{(58)},f(7)\oplus\widetilde{N}_{(78)}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.46,0.9],[0.06,0.42],[0.01,0.3]\rangle\\ \oplus\langle[0.3,0.7],[0.2,0.6],[0.1,0.5]\rangle,\\ \langle[0.28,0.84],[0.04,0.42],[0.16,0.64]\rangle\\ \oplus\langle[0.1,0.5],[0.2,0.6],[0.4,0.8]\rangle\end{array}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.622,0.97],[0.012,0.252],[0.001,0.15]\rangle,\\ \langle[0.352,0.92],[0.008,0.252],[0.064,0.512]\rangle\end{array}\right\}$$
$$=\langle[0.352,0.92],[0.008,0.252],[0.064,0.512]\rangle$$
,

$$f(9)=\min_{i<9}\left\{f(i)\oplus\widetilde{N}_{(i9)}\right\}$$
$$=\min\left\{f(6)\oplus\widetilde{N}_{(69)},f(7)\oplus\widetilde{N}_{(79)},f(8)\oplus\widetilde{N}_{(89)}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.28,0.82],[0.06,0.54],[0.04,0.28]\rangle\\ \oplus\langle[0.2,0.6],[0.2,0.5],[0.5,0.9]\rangle,\\ \langle[0.28,0.84],[0.04,0.42],[0.16,0.64002]\rangle\\ \oplus\langle[0.4,0.9],[0.2,0.6],[0.1,0.5]\rangle,\\ \langle[0.352,0.92],[0.008,0.252],[0.064,0.512]\rangle\\ \oplus\langle[0.2,0.6],[0.1,0.5],[0.5,0.8]\rangle\end{array}\right\}$$
$$=\min\left\{\begin{array}{l}\langle[0.424,0.928],[0.012,0.27],[0.02,0.252]\rangle,\\ \langle[0.568,0.984],[0.008,0.252],[0.016,0.32]\rangle,\\ \langle[0.4816,0.968],[0.0008,0.126],[0.032,0.4096]\rangle\end{array}\right\}$$
$$=\langle[0.424,0.928],[0.012,0.27],[0.02,0.252]\rangle$$
Thus

$$f(9)=f(6)\oplus\widetilde{N}_{(69)}$$
$$=f(3)\oplus\widetilde{N}_{(36)}\oplus\widetilde{N}_{(69)}$$
$$=f(1)\oplus\widetilde{N}_{(13)}\oplus\widetilde{N}_{(36)}\oplus\widetilde{N}_{(69)}$$
$$=\widetilde{N}_{(13)}\oplus\widetilde{N}_{(36)}\oplus\widetilde{N}_{(69)}$$

Therefore, the path $p_{(1369)}$ is recognized as the neutrosophic shortest path, its neutrosophic distance is $\langle[0.424,0.928],[0.012,0.27],[0.02,0.252]\rangle$, and its score function $S(p_{(1369)})=\frac{1}{4}\times(2+0.424+0.928-2\times0.012-2\times0.27-0.02-0.252)=0.629$.

From the time complexity of the algorithm, after the debugging of multiple parameters, the number of paths randomly selected by the ant colony algorithm is preferably 1 to 1.5 times the number of vertices, and the number of iterations is approximately 150. Therefore, the time complexity of the ant colony algorithm is $O(180n)$, that of the known Dijkstra algorithm is $O(n^2)$, and that of the Bellman algorithm is $O(n^3)$. Assuming that each operation takes t = 0.0001 s, the relationships between the running time of the three algorithms and the number of vertices n of the neutrosophic graph are shown in Figs. 6 and 7.

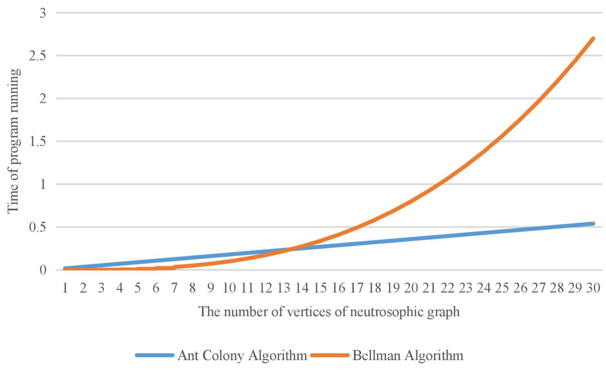

FIGURE 6. Comparison of the running time of the ant colony algorithm and Bellman algorithm.

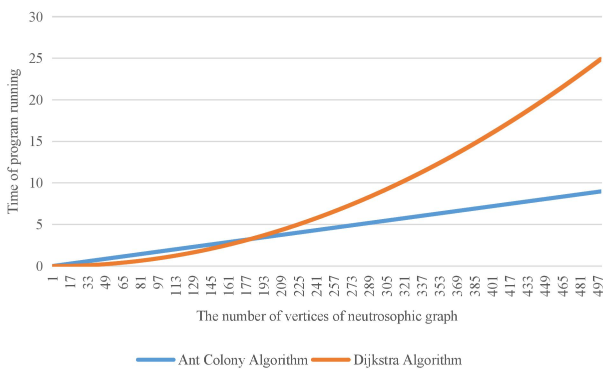

FIGURE 7. Comparison of the running time of the ant colony algorithm and Dijkstra algorithm.

In these figures, the abscissa represents the number of vertices of the neutrosophic graph, and the ordinate represents the running time of the algorithm. As shown in Figs. 6 and 7, when the number of vertices $n > 14$, the ant colony algorithm consumes less time compared to the Bellman algorithm, and when the number of vertices $n > 180$, the ant colony algorithm consumes less time than the Dijkstra algorithm. Hence, the ant colony algorithm is feasible and reasonable; compared with the Dijkstra and Bellman algorithms, it has the following advantages:

1) The ant colony algorithm is a self-organizing algorithm. In system theory, self-organization and other organizations are two basic categories of organization. The difference lies in whether organizational power or instructions come from inside or outside of the system; when they arise from the inside, the system is self-organizing, but when they arise from the outside, the system is other organizing. If there is no specific external intervention in the process of obtaining spatial, temporal, or functional structure, we may say that the system is self-organized. In the abstract sense, self-organization is the process of increasing the system fluidity without external effects. This is particularly suitable for the case of uncertain path damage arising when a disaster occurs.

2) The ant colony algorithm is essentially a parallel algorithm. Each ant search process is independent of others and only communicates via pheromones. Therefore, the ant colony algorithm can be regarded as a distributed multi-system. It begins by searching independent solutions at multiple points in the problem space at the same time; this not only increases the reliability of the algorithm but also provides the algorithm strong global search ability.

3) The ant colony algorithm is very robust. Unlike the Dijkstra and Bellman algorithms, the ant colony algorithm does not have demanding requirements for defining the initial route, the results do not depend on the choice of initial route, and there is no need for manual adjustment in the search process.

C. COMPARATIVE ANALYSES OF DIFFERENT PARAMETER SETTINGS

Five paths are randomly selected each time, and the case in this paper is recalculated with the ant colony algorithm. The local optimal solution information obtained during iteration is shown in Table V.

TABLE V

LOCAL OPTIMAL SOLUTION PRODUCED BY EACH ITERATION

Times	Minpath	Score function	Pheromone	N(min)
1	$p_{(1479)}$	0.67400	1.04500	0
2	$p_{(14789)}$	0.68860	0.99275	0
3	$p_{(14789)}$	0.68860	1.03742	0
4	$p_{(14789)}$	0.68860	1.08411	0
5	$p_{(14579)}$	0.79600	0.85116	0
⋮	⋮	⋮	⋮	⋮
440	$p_{(1369)}$	0.62900	3.03E-09	0
441	$p_{(13479)}$	0.82349	3.89E-10	0
442	$p_{(124789)}$	0.75800	9.59E-10	0

443	$p_{(1369)}$	0.62900	2.86E-09	1
444	$p_{(1369)}$	0.62900	2.99E-09	1
⋮	⋮	⋮	⋮	⋮
485	$p_{(1369)}$	0.62900	1.82E-08	4
486	$p_{(1369)}$	0.62900	1.90E-08	4
487	$p_{(1369)}$	0.62900	1.98E-08	4
488	$p_{(1369)}$	0.62900	2.07E-08	4
489	$p_{(1369)}$	0.62900	2.17E-08	5

When five paths are randomly selected each time, the shortest path P can also be found eventually. Unlike the data in Table IV, Table V shows that the number of iterations increases significantly, and many iterations lead to significant pheromone volatilization. The comparison in the convergence processes for the two is shown in Fig. 8.

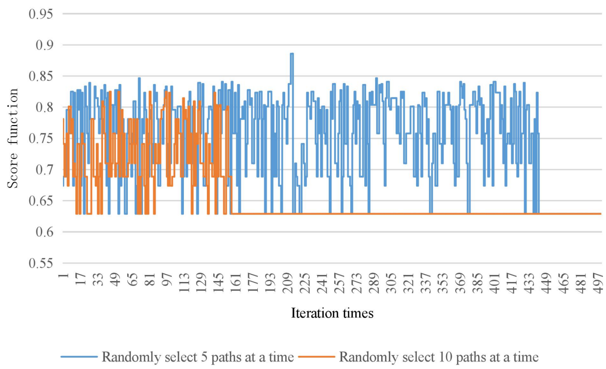

FIGURE 8. Comparison of convergence processes with different parameters.

The abscissa in Fig. 8 represents the number of iterations, the ordinate represents the score function of the local optimal solution obtained during iteration, and the blue line represents the convergence process when five paths are randomly selected. The orange line represents the convergence process when 10 paths are randomly selected. By comparing the two curves, the figure shows that reducing the number of randomly selected paths during iteration reduces the extent to which the calculation reaches the local optimal solution during iteration, but more iteration will be required for convergence. When 10 paths are randomly selected, the local optimal solution converges in the 159th iteration; when five paths are randomly selected, the local optimal solution converges on the 433rd iteration.

D. COMPARATIVE ANALYSIS OF DIFFERENT SCORE FUNCTIONS

The score function for interval valued NN, proposed in reference [31] is as follows:

$$S_{Peng}(\widetilde{N}) = \frac{2}{3} + \left(\frac{a_1 + a_2}{6} - \frac{b_1 + b_2}{6} - \frac{c_1 + c_2}{6}\right) \quad (8)$$

We use this score function to solve the SPP of the ant colony algorithm with the following approach:

Step 1: Initialize the pheromone matrix, and uniformly set the pheromone concentration of each path from starting point ① to ending point ⑨ to $C_i = 1, i \in [1,85]$.

Step 2: Randomly select 10 paths from the starting point ① to the ending point ⑨: $P_{(125789)}$, $P_{(1246789)}$, $P_{(13479)}$, $P_{(134579)}$, $P_{(145789)}$, $P_{(14579)}$, $P_{(146789)}$, $P_{(125789)}$, $P_{(134789)}$ and $P_{(134579)}$. Calculate the distances of all paths according to *Definition 4* and then calculate the score function of each path according to *Definition 5*, as shown in Table VI.

TABLE VI

SCORE FUNCTIONS OF PATHS

Paths	Score function of paths (S_{Peng})
$P_{(125789)}$	0.95783
$P_{(1246789)}$	0.97702
$P_{(13479)}$	0.93988
$P_{(134579)}$	0.95839
$P_{(145789)}$	0.96476
$P_{(14579)}$	0.95414
$P_{(146789)}$	0.99827
$P_{(125789)}$	0.95783
$P_{(134789)}$	0.94673
$P_{(134579)}$	0.95840

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

According to *Definition 6*, the shortest path of the first randomly generated 10 paths can be found: $P_{\min[0]} = P_{(134789)}$.

Step 3: Update pheromone levels:

1) The concentration of all pheromones was reduced by 5% , and $C_i = 1*(1-0.05) = 0.95, i \in [1,85]$.

2) The pheromone concentration of path $P_{\min_{[0]}}$ increases by 10% ; $C_{(145789)} = 0.95*1.1 = 1.045$.

Step 4: Determine the ratio $per_{(p_{\min[0]})} = \frac{c_{(p_{\min[0]})}}{\sum_{i=0}^{n} c_i} = \frac{1.045}{0.95*84+1.045} = 0.013$ of the pheromone level of path $P_{\min_{[0]}}$ to the total pheromone level and calculate the product $N_{p_{\min[0]}} = round(N * per_{(p_{\min[0]})}) = round(10 * 0.013) = ($ for the next path $P_{\min_{[0]}}$; that is, it is not necessary to specify the selection path $P_{(134789)}$, but only to select another 10 paths randomly. Subsequent, Steps 2 and 3 were repeated. Finally, the local optimal solution converges to $P_{(1369)}$. The information on the local optimal solution generated during iteration is shown in Table VII.

TABLE VII

LOCAL OPTIMAL SOLUTION PRODUCED BY EACH ITERATION

Times	Minpath	SPeng	Pheromone	N (min)
1	$p_{(124789)}$	0.92810	1.04500	0
2	$p_{(13479)}$	0.93987	0.99275	0
3	$p_{(12479)}$	0.91868	0.94311	0
4	$p_{(1369)}$	0.89035	0.89595	0
5	$p_{(124589)}$	0.93906	0.85116	0
⋮	⋮	⋮	⋮	⋮
160	$p_{(12479)}$	0.91868	0.00109	0
161	$p_{(12479)}$	0.91868	0.00114	0
162	$p_{(1245789)}$	0.94736	0.00029	0
163	$p_{(1369)}$	0.89035	0.00139	0
164	$p_{(1369)}$	0.89035	0.00143	1
⋮	⋮	⋮	⋮	⋮
220	$p_{(1369)}$	0.89035	0.01758	9
221	$p_{(1369)}$	0.89035	0.01837	9
222	$p_{(1369)}$	0.89035	0.01920	9
223	$p_{(1369)}$	0.89035	0.02006	9
224	$p_{(1369)}$	0.89035	0.02097	10

The comparison of Tables IV and VII indicates that the shortest path $P_{(1369)}$ can be calculated by using two different score functions and the number of iterations are similar. It is clear that when using the ant colony algorithm to solve the SPP, the complexity of the algorithm is not directly related to the score function of the interval valued NN. The convergence processes for the local optimal solution calculated by two score functions are shown in Fig. 9:

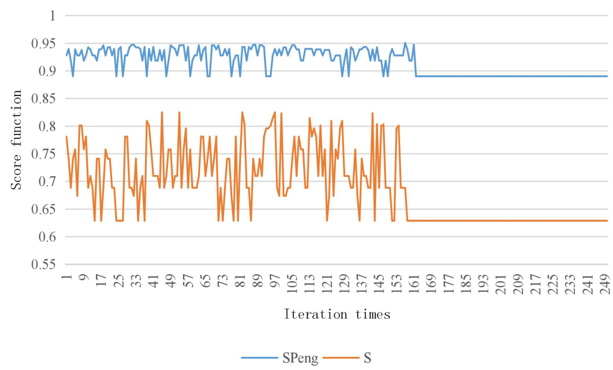

FIGURE 9. Convergence comparison chart with different score functions.

In Fig. 9, the abscissa represents the number of iterations, and the ordinate represents the score function of the local optimal solution obtained during iteration. The curve S represents the convergence process for the local optimal solution when the score function is as proposed in reference [19], and the curve SPeng represents the convergence process for the local optimal solution when the score function proposed in reference [31] is used. The figure shows that when using different score functions to seek convergence, the final shortest path score function is inconsistent, S converges to 0.62900, SPeng converges to 0.89035, and the paths they represent are both $P_{(1369)}$ when the different score functions are iterative. The local optimal solution converged after approximately 160 cycles, indicating that the ant colony algorithm used to solve the SPP of the interval valued neutrosophic graph is stable.

VII. CONCLUSION

In this study, we have developed an ant colony algorithm for solving the SPP on a network with interval valued neutrosophic edge distances, and the effectiveness of the algorithm is proved by implementing it using an example. Thereafter, the ant colony algorithm is compared with the Bellman and Dijkstra algorithms. The comparison proves that the ant colony algorithm has better time complexity. Further, we studied the effect of different parameter settings on the convergence process of the ant colony algorithm. Finally, we used different score functions to solve the SPP and obtained a consistent optimal solution, thereby proving the stability of the ant colony algorithm. Although the time cost of the ant colony algorithm is linear, it must undergo a sufficient number of iterations to obtain the optimal solution. Therefore, for the SPP of a simple neutrosophic graph with a few nodes, the ant colony algorithm is not as efficient as the classic Dijkstra algorithm. In the future, we will apply our method to the multi-source SPP of complex neutrosophic graphs.

REFERENCES

[1] F. Smarandache, *A Unifying Field in Logics. Neutrosophy: Neutrosophic Probability, Set and Logic*. Rehoboth, DE, USA: American Research Press, 1999, pp. 7–8.

[2] M. Ali *et al.*, "Complex Neutrosophic Set," *Neural. Comput. Appl.*, vol. 27, no. 1, pp. 1-18, Jan. 2016.

[3] S. Broumi *et al.*, "Single valued neutrosophic graphs," *J. New Theory.*, vol. 10, no. 2, pp. 86-101, Feb. 2016.

[4] S. Broumi *et al.*, "An introduction to bipolar single valued neutrosophic graph theory," *Appl. Mech. Mater.*, vol. 841, no. 6, pp. 184-191, Jun. 2016.

[5] S. Broumi *et al.*, "Shortest path problem under bipolar neutrosphic setting," *Appl. Mech. Mater.*, vol. 859, pp. 59-66, Dec. 2016.

[6] R. K. Ahuja *et al.*, "Faster algorithms for the shortest path problem," *J. Acm.*, vol. 37, no. 2, pp. 213-223, Apr. 1990.

[7] C. D. Yang *et al.*, "On bends and lengths of rectilinear paths: a graph theoretic approach," *Int. J. Comput. Geom. Appl.*, vol. 2, no. 1, pp. 61-74, Jan. 1992.

[8] P. S. Yu *et al.*, "The potential of fuzzy multi-objective model for rainfall forecasting from typhoons," *Nat. Hazards.*, vol. 34, no. 2, pp. 131-150, Feb. 2005.

[9] C. S. Chen *et al.*, "Typhoon event-based evolutionary fuzzy inference model for flood stage forecasting," *J. Hydrol.*, vol. 490, no. 20, pp. 134-143, May. 2013.

[10] R. Tan *et al.*, "Multi-Attribute Decision-Making Method Based on Prospect Theory in Heterogeneous Information Environment and Its Application in Typhoon Disaster Assessment," *Int. J. Comput. Int. Sys.*, vol. 12, no. 2, pp. 881-895, Jul. 2019.

[11] J. Buckley *et al.*, "Fuzzy shortest path problem. Monte Carlo methods in fuzzy optimization," *Stud. Fuzziness Soft Comput.*, vol. 222, pp. 191-193, Nov. 2007.

[12] Y. Deng *et al.*, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment," *Appl. Soft Comput.*, vol. 12, no. 3, pp. 1231-1237, Mar. 2012.

[13] S. S. Biswas *et al.*, "An algorithm for extracting intuitionistic fuzzy shortest path in a graph," *Appl. Comput. Intell. Soft Comput.*, vol. 2013, no. 1, pp. 1-5, Jan. 2013.

[14] F. Smarandache. "Neutrosophic Set is a Generalization of Intuitionistic Fuzzy Set, Inconsistent Intuitionistic Fuzzy Set (Picture Fuzzy Set, Ternary Fuzzy Set), Pythagorean Fuzzy Set (Atanassov's Intuitionistic Fuzzy Set of second type), q-Rung Orthopair Fuzzy Set, Spherical Fuzzy Set, and n-HyperSpherical Fuzzy Set, while Neutrosophication is a Generalization of Regret Theory, Grey System Theory, and Three-Ways Decision (revisited)," *J. New Theory.*, vol. 29, pp. 1-35, Nov. 2019.

[15] J. Ye *et al.*, "Multiple-attribute decision-making method under a single-valued neutrosophic hesitant fuzzy environment," *Appl. Comput. Intell. Soft Comput.*, vol. 24, no. 1, pp. 23-36, Jan. 2015.

[16] J. Peng *et al.*, "Multi-valued Neutrosophic Sets and its Application in Multi-criteria Decision-making Problems," *Neutrosophic Sets Syst.,* vol. 10, no. 1, pp. 3-17, Jan. 2015.

[17] J. Ye *et al.*, "Trapezoidal neutrosophic set and its application to multiple attribute decision-making," *Neural. Comput. Appl.*, vol. 26, no. 5, pp. 1157-1166, Jul. 2015.

[18] H. Nancy *et al.*, "An improved score function for ranking neutrosophic sets and its application to decision making process," *Int. J. Uncertain Quantif.*, vol. 6, no. 5, pp. 377-385, May. 2016.

[19] S. Ridvan *et al.*, "Maximizing deviation method for neutrosophic multiple attribute decision making with incomplete weight information," *Neural. Comput. Appl.*, vol. 27, no. 7, pp. 2017-2029, Jul. 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990912, IEEE Access

[20] Deli *et al*., "A novel defuzzification method of SV-trapezoidal neutrosophic numbers and multi-attribute decision making: a comparative analysis," *Soft Comput*., vol. 23, no. 4, pp.12529-12545, Apr. 2019.

[21] S. Broumi *et al*., "Computing minimum spanning tree in interval valued bipolar neutrosophic environment," *Int. J. Model. Optim*., vol. 7, no. 5, pp. 300-304, Oct. 2017.

[22] H, Hu *et al*., "On solving the quadratic shortest path problem," *Informs J. Comput*., vol. 99, no. 8, pp. 1-30, Aug. 2017.

[23] Y, Zhang *et al*., "Robust Shortest Path Problem With Distributional Uncertainty," *IEEE T. Intell. Transp*., vol. 99, no. 6, pp. 1-11, Jun. 2017.

[24] S. Broumi *et al*., "Shortest path problem on single valued neutrosophic graphs," presented at the *ISNCC-2017,* Marrakech, Morocco, May. 16-18, 2017.

[25] X, Peng *et al*., "Algorithms for interval neutrosophic multiple attribute decision-making based on MABAC, similarity measure, and EDAS," *Int. J. Uncertain Quantif*., vol. 7, no. 5, pp. 395-421 May. 2017.

[26] F. Smarandache, *A Unifying Field in Logics. Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability (Fifth Edition)*. Rehoboth, USA: American Research Press, 2005, pp. 123–135.

[27] H. B. Wang *et al*., "Single valued neutrosophic sets," *Tech. Sci. Appl. Math*., vol. 20, no. 10, pp. 10-14, Oct. 2012.

[28] Deli *et al*., "A ranking method of single valued neutrosophic numbers and its applications to multi-attribute decision making problems," *Int. J. Mach. Learn Cybern*., vol. 8, no. 4, pp. 1309-1322, Apr. 2017.

[29] S. Broumi, *New Trends in Neutrosophic Theor Appl*. Surapati Pramanik: Pons edition, 2018, pp. 403–434.

[30] S. Broumi *et al*., "Shortest path problem under interval valued neutrosophic setting. Software Guide," *J. Fundam. Appl. Sci*., vol. 10, no. 4S, pp. 168-174 May. 2018.

[31] E. Bolturk *et al*., "A novel interval-valued neutrosophic AHP with cosine similarity measure," *Soft Comput*., vol. 22, no. 3, pp. 4941-4958, Mar. 2018.

[32] H.-B. Wang, *Interval neutrosophic sets and logic: theory and applications in computing*. Hexis, Phoenix, USA: Wadsworth, 2005, pp. 123–135.

[33] P, Biswas *et al*., "Distance measure based MADM strategy with interval trapezoidal neutrosophic numbers," *Neutrosophic Sets Syst.,* vol. 19, no. 10, pp. 40-46, Oct. 2018.

[34] Deli *et al*., "Expansions and reductions on neutrosophic classical soft set," *Süleyman Demirel Univ. J. Nat. Appl. Sci*., vol. 22, no. 9, pp. 478-486, Sep. 2018.

[35] Deli *et al*., "Operators on single valued trapezoidal neutrosophic numbers and SVTN-group decision making," *Neutrosophic Sets Syst.,* vol. 22, pp. 131-151, Nov. 2018.

[36] Deli *et al*., "Some weighted geometric operators with SVTrN-numbers and their application to multi-criteria decision making problems," *J. Intell. Fuzzy Syst*., vol. 32, no. 1, pp. 291-301, Jan. 2017.

[37] M. A. Basset *et al*., "A hybrid neutrosophic multiple criteria group decision making approach for project selection," *Cogn. Syst. Res*., vol. 57, pp. 216-227, Oct. 2019.

[38] M. A. Basset *et al*., "Neutrosophic AHPDelphi Group decision making model based on trapezoidal neutrosophic numbers," *J. Amb. Intel. Hum. Comp*., vol. 9, no. 7, pp. 1427-1443, Jul. 2017.

[39] R. Kumar *et al*., "Neutrosophic shortest path problem," *Neutrosophic Sets Syst.,* vol. 23, pp. 5-15, Dec. 2018.

[40] S. Broumi *et al*., "Application of Dijkstra Algorithm for Solving Interval Valued Neutrosophic Shortest Path Problem," *Comput. Intell-Us*., vol. 1, no. 1, pp. 1-6, Jan. 2017.

[41] R. P. Tan *et al*., "Solving methods for the shortest path problem based on trapezoidal fuzzy neutrosophic numbers," *Control Decis*., vol. 34, no. 4, pp. 851-960, Apr. 2019.

[42] S. Broumi, D. Nagarajan, A. Bakali, M. Talea, F. Smarandache, M. Lathamaheswari, "The shortest path problem in interval valued trapezoidal and triangular neutrosophic environment," *Complex Intell. Syst*., vol. 5, pp. 391-402, Feb. 2019.

[43] S. Broumi, A. Dey, M. Talea, A. Bakali, F. Smarandache, D. Nagarajan, M. Lathamaheswari, R Kumar, "Shortest path problem using Bellman algorithm under neutrosophic environment."" *Complex Intell. Syst*., vol. 5, pp. 409-416, Feb. 2019.

[44] A. Chakraborty. "Application of Pentagonal Neutrosophic Number in Shortest Path Problem," *Int. J. Neutrosophic Sci*., vol. 3, no. 1, pp. 21-28, Jan. 2020.

[45] F. Smarandache. "NeutroAlgebra is a Generalization of Partial Algebra," *Int. J. Neutrosophic Sci*., vol. 2, no. 1, pp. 8-17, Jan. 2020.

[46] P. Schweizer. "Uncertainty: two probabilities for the three states of neutrosophy," *Int. J. Neutrosophic Sci*., vol. 2, no. 1, pp. 18-26, Jan. 2020.

[47] S. A. Edalatpanah. “A Direct Model for Triangular Neutrosophic Linear Programming,” *Int. J. Neutrosophic Sci.*, vol. 1, no. 1, pp. 19-28, Jan. 2020.

[48] D. P. Qiao *et al*., "Ageneral overview on ant colony algorithm and its application," *Softw. Guide*, vol. 116, no. 12, pp. 217–221, Dec. 2017.