An adaptive neutrosophic large neighborhood search and its application

Jun Jiang [1], Fangwei Zhang [2,*], Yiying Pan [1], Rui Wang [1], and Zhu Han [1]

1 School of Navigation and Shipping, Shandong Jiaotong University, Weihai China
E-mail: jiang19560628511@163.com, yiyingpan02@163.com, ruiwang27@163.com, 3098843861@qq.com
2 School of International Business, Shandong Jiaotong University, Weihai China
E-mail: fangweirzhang@163.com

* Correspondence: fangweirzhang@163.com

Abstract: An adaptive large neighborhood search (ALNS) lacks a clear measure for assessing the improvement degree in the new solution, which causes fuzziness and uncertainty in the operator score. To solve the aforementioned problem, the main innovation of this study is to propose an adaptive neutrosophic large neighborhood search (ANLNS). Specifically, the main work is as follows. Firstly, the number of times each operator scores are quantified by constructing NSs, thereby analyzing algorithm performance and preventing the idealization of scores. Secondly, a novel neutrosophic utility function and score function are proposed to assign an appropriate score for the operator within a reasonable interval. Finally, the effectiveness and robustness of the proposed ANLNS is validated by the capacitated vehicle routing problem benchmarks with three varying scales and comparative analyses. The compared results indicate that the proportion of best solutions for ANLNS are 50%, 100%, and 37.5%, which significantly highlight the robustness and reliability of the proposed algorithm when the degree of destruction is 0.3, 0.5, and 0.7, respectively. Meanwhile, the proposed ANLNS is efficient and flexible, providing a novel method for addressing other situation optimization problems.

Keywords: Neutrosophic sets; Utility function; Adaptive neutrosophic large neighborhood search; Capacitated vehicle routing problem

1. Introduction

As the complexity and particularity of contemporary work environments, the efficacy of solutions and the rationality of algorithm workflows are of concern dramatically. Therefore, it is noted that an increasing number of heuristic algorithms have been widely applied and researched, such as adaptive large neighborhood search (ALNS) [1]. Usually, the six primary sections of ALNS process are initialization, roulette wheel selection (RWS), destroy operators, repair operators, adaptive weight adjustment, and acceptance and stopping criteria [2]. Especially, in the adaptive weight adjustment stage, the ALNS only focuses on whether the new solution is better than the best solution or current solution. If this condition is met, the operator increases the fixed score. However, this established scoring mechanism does not evaluate the specific improvement degree in the new solution, which makes it difficult to assign the appropriate score for each operator and exhibits fuzziness and uncertainty.

Based on the aforementioned problems, this study mainly focuses on how to assign an appropriate score for each operator. Thereafter, this study proposes an adaptive neutrosophic large neighborhood search (ANLNS) algorithm to better assign the operator score. Specifically, the main contributions are as follows.

Firstly, this study keeps track of the number of times that each operator scores under four different situations, and quantifies these number of times by constructing neutrosophic sets (NSs). Among them, NSs facilitate the ANLNS to comprehensively analyze operator performance and enhance the ability to evaluate actual contributions of each operator.

Secondly, this study proposes a neutrosophic utility function based on neutrosophic sets and a score function. This combination of functions not only better measures the algorithm's preference for different operators but also assign a suitable score within a reasonable interval. Moreover, the scores are rational, reducing the existence of subjective factors.

Finally, this study verifies the robustness and validity of ANLNS through the capacitated vehicle routing problem (CVRP) benchmark instances. Additionally, it provides a novel method to address the CVRP by applying ANLNS.

The rest of this study is organized as follows. Section 2 introduces the literature review of ALNS, NSs, and CVRP. Section 3 indicates some definitions of ALNS and NSs, introduces a mathematical model on CVRP, and proposes the ANLNS. Section 4 provides a study case and proves its validity and robustness of the proposed ANLNS. In Section 5, this study presents some conclusions and insights.

2. Literature review

For convenience, this section reviews previous studies on ALNS, NSs, and CVRP, providing an overview of existing deficiencies that need attention.

ALNS was proposed by Pisinger and Ropke [3] to address the vehicle routing problem (VRP) under various transportation conditions. Subsequently, scholars have carried out extensive expansion and in-depth discussion on the ALNS in recent decades. For instance, Cai et al. [4] employed a hybrid heuristic algorithm that integrates ALNS and tabu search to address the electric vehicle relocation problem. Then, aiming at a bi-objective green VRP, Amiri et al. [5] presented an approach integrating three multi-objective solution methods with the ALNS. In the same year, by enhancing the ALNS algorithm, Fathollahi-Fard et al. [6] resolved the general quadratic allocation problem efficiently. Chen et al. [7] designed a new ALNS to effectively address the routing of delivery robots. Simultaneously, Rolim et al. [8] constructed a new earliness-tardiness extension of the ALNS, which contained several insertion operators and an embedded procedure.

Meanwhile, NSs offer a powerful framework for effectively analyzing and solving uncertain, fuzzy, and contradictory information. In these regards, some scholars have delved into the application of NSs. For example, Rashno et al. [9] designed a new clustering algorithm based on NSs, effectively addressing boundary and outlier points by defining data indeterminacy and optimizing a cost function through gradient descent methods. Then, Saeed et al. [10] extended and proposed the neutrosophic hypersoft graph to hand neutrosophic hypersoft information. Later, Ye [11] extended the application of neutrosophic (indeterminate fuzzy) multivalued sets and achieved great advancements in teaching quality assessment of teachers. Then, Smarandache [12] reviewed the definitions and properties of six new topologies introduced between 2019 and 2022. Barbosa and Smarandache [13] introduced the Pura Vida Neutrosophic Algebra based on neutrosophic numbers, which apply to the Neutrosophic matrices and vectors. Smarandache [14] designed a novel SuperHyperSoft Set and Fuzzy Extension SuperHyperSoft Set, which provides examples to present these applications. Smarandache [15] applied the n-th PowerSet of a set to describe the organizational architecture in the real world more comprehensively and discussed SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction, and Neutrosophic SuperHyperStructure. Smarandache [16] identified nine groundbreaking topologies, providing a comprehensive overview and delving into current trends and future challenges. Al-Hijjawi and Alkhazaleh [17] proposed a concept of the possibility of neutrosophic hypersoft sets and gave the basic operations and characteristics. To better understand Neutrosophic Numbers' operation in the neutrosophic statistics framework, Smarandache [18] innovatively presented the Appurtenance Equation and Inclusion Equation and gave the solution method. Mishra et al. [19] proposed an integrated multi-attribute decision-making methodology using single-valued NSs to evaluate and prioritize energy storage technologies.

Besides, CVRP is NP-hard. CVRP has made some improvements and extensions on the basis of the traditional traveling salesman problem [20]. For example, Vidal [21] implemented a simple and open-source Hybrid Genetic Search algorithm, which specially contained a new neighborhood for the CVRP. Then, Rezaei et al. [22] proposed solving the CVRP, combining a refined imperialist competitive algorithm with a hybrid genetic search algorithm for enhancing optimization. After that, Jiang et al. [23] designed an evolutionary algorithm for solving CVRP based on the relevance matrix. At the same time, Wang et al. [24] effectively improved the genetic algorithm and successfully solved the multi-objective CVRP with workload balancing. Simultaneously, Chi and He [25] addressed the pickup CVRP with three-dimensional loading constraints by improving a traditional branch-and-price algorithm.

Based on the aforementioned, it indicates that many improved ALNS algorithms satisfy some practical work requirements. However, the issues of fuzziness and uncertainty need to be resolved in the ALNS algorithm. Specifically, the traditional scoring mechanism of the ALNS is to subjectively assign a fixed score to the operator, failing to effectively reflect the specific improvement degree in the new solution and leading to uncertainty and ambiguity in the operator score. Although NSs are widely applied, they have not yet been integrated into ALNS, which provides a novel direction for research. Aiming at the above problems, this study proposes the integration of NSs into the ALNS to better assign operator scores and reflect their performance.

3. Materials and methods

For convenience, this section briefly introduces the basic definitions of ALNS and NSs, and basic problem description on CVRP.

3.1. Preliminaries

Definition 1. [26] The ALNS algorithm flow refers to the pseudo-code of Algorithm 1.

ALGORITHM 1 ALNS.

1: **Input**: the initial solution x;
2: $x_{current} \leftarrow x$; $x_{best} \leftarrow x_{current}$;
3: $q = \mu \cdot C_N$; $t = 0$;
4: **While** $t \leq max\_item$ **do**
5: $x_s \leftarrow x_{current}$;
6: $x' \leftarrow$ select a destroy operator by using the RWS, and remove q requests from x_s;
7: $x_{new} \leftarrow$ select a repair operator by using the RWS, and insert q requests into x';
8: **If** $f(x_{new}) < f(x_{best})$ **then**
9: $x_{best} \leftarrow x_{new}$; $x_{current} \leftarrow x_{new}$; increase the score σ_1 for the utilized operators;
10: **Else if** $f(x_{new}) < f(x_{current})$ **then**
11: $x_{current} \leftarrow x_{new}$; increase the score σ_2 for the utilized operators;
12: **Else if** the simulated annealing criteria is met, **then**
13: $x_{current} \leftarrow x_{new}$; increase the score σ_3 for the utilized operators;
14: **Else**
15: increase the score σ_4 for the utilized operators;
16: **End if**
17: update scores and weights of all operators;
18: $t \leftarrow t + 1$;
19: **End while**
20: **Output**: the best solution x_{best}.

Here, Line 1 indicates that the ALNS selects the initial feasible solution x as the input parameter. Line 2 indicates that the current solution $x_{current}$ is assigned the value of x, followed by setting the best solution x_{best} equal to $x_{current}$. Line 3 indicates that q requests are removed from the solution, and the start iteration number is set to 0. Among them, the parameter C_N represents the size of the solution, and the parameter μ denotes the degree of destruction to which requests are removed from the solution. Line 5 indicates that a solution x_s is a copy of the $x_{current}$. Lines 6-7 execute the RWS to select the operators that construct a new solution x_{new}. Apply lines 8-11 to update x_{best} and $x_{current}$. If x_{new} is better than x_{best}, x_{best} and $x_{current}$ replace x_{new}. Otherwise, if x_{new} is better than $x_{current}$, $x_{current}$ replaces x_{new}. Then, if the operators produce x_{new} that is worse than $x_{current}$, accept x_{new} using the simulated annealing acceptance criterion (lines12-16). Line 17 performs the update of the operator scores and weights. Finally, the algorithm outputs x_{best} in line 20.

Definition 2. [27, 28] Let X be a space of points (objects), and each element of X is represented by x. A truth-membership function $T_S(x)$, an indeterminacy-membership function $I_S(x)$, and a falsity-membership function $F_S(x)$ are used to describe a NS S in X. Then, use $S = \{\langle x, T_S(x), I_S(x), F_S(x)\rangle | x \in X\}$ as a representation of S. Functions $T_S(x)$, $I_S(x)$, and $F_S(x)$ are real standard or non-standard subsets of $]0^-, 1^+[$, denoted by $T_S(x): X \to]0^-, 1^+[$, $I_S(x): X \to]0^-, 1^+[$, and $F_S(x): X \to]0^-, 1^+[$ respectively. There is no restriction on the sum of $T_S(x)$, $I_S(x)$, and $F_S(x)$, resulting in $0^- \leq T_S(x) + I_S(x) + F_S(x) \leq 3^+$. For simplicity, if X has only one element, S is expressed as $S = \langle T_S(x), I_S(x), F_S(x)\rangle$ and can be called a neutrosophic number.

Example: The student's performance grade is considered a neutrosophic number to effectively represent a student's performance grade in a subject: ⟨7.2,7.5,7.8⟩. The student's grade in the truth-membership degree is 7.2. The student's grade in the indeterminacy-membership degree is 7.5. The student's grade in the falsity-membership degree is 7.8.

3.2. Problem description

The CVRP was defined on a connected graph $G = (V, E)$. Among them, $V = \{V_0, V_1, \ldots, V_{n+1}\}$ is a vertex set and $E = \{(V_i, V_j) | V_i, V_j \in V, V_i \neq V_j\}$ is an arc set. Each arc represents the distance between two vertices. The vertex V_0 represents the depot, and the other vertices represent the customer point. Each vehicle has the same loading capacity and carries the same kind of goods, that is, constitutes a group of homogeneous fleets. Each vehicle begins its route at the depot and ultimately returns to the depot after completing the distribution task. The demand for goods by all customers is known, with each customer being served by exactly one vehicle. The loading capacity and total distance of each vehicle are not allowed to exceed the maximum loading capacity and maximum distance. The CVRP aims at minimizing total cost, which, in practical terms, translates to reducing the overall transportation distance. Then, the CVRP is expressed as follows:

$$\min f = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{v=1}^{K} d_{ij} x_{ijv} \tag{1}$$

$$s.t.$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij} x_{ijv} \le T_v, 1 \le v \le K; \tag{2}$$

$$\sum_{v=1}^{K}\sum_{j=1}^{N} x_{0jv} = \sum_{v=1}^{K}\sum_{j=1}^{N} x_{j0v} \le 1; \tag{3}$$

$$\sum_{v=1}^{K}\sum_{j=1}^{N}\sum_{i=0}^{N} x_{ijv} = 1; \tag{4}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} x_{ijv} M_i \le Q_v, v \in \{1, \ldots, K\}; \tag{5}$$

$$\sum_{v=0}^{K}\sum_{i=0}^{N} x_{ijk} \le K, \quad for \quad i = 0, \tag{6}$$

where Eq. (1) indicates an objective function of CVRP. Eq. (2) indicates that each vehicle follows the maximum traveling distance. Eq. (3) indicates that each vehicle begins and ends in the depot. Eq. (4) makes sure each customer only accesses one vehicle. Eq. (5) indicates the constraint of maximum loading capacity. Eq. (6) indicates that the number of available routes after leaving the depot, which is no more than K. x_{ijv} is a 0-1 binary variable, where $x_{ij} = 1$ indicates that vehicle v travels from customer i to customer j, and 0 otherwise. d_{ij} represents the distance from customer i to customer j, N indicates the total number of customers served, K represents the maximum number of vehicles utilized, Q_v represents the maximum loading constraint for vehicle v, M_i represents customer i demand, T_v represents the maximum distance traveling of vehicle v. Meanwhile, Figure 1 illustrates an instance of CVRP to enhance understanding.

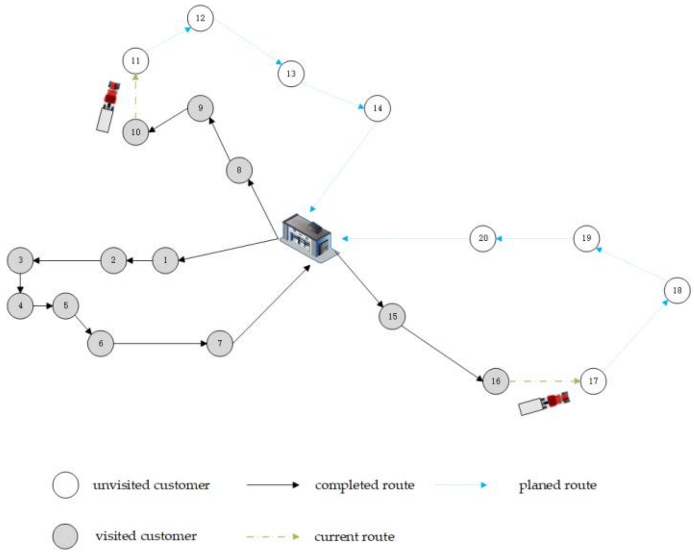


Figure 1. Illustration of the capacitated vehicle routing problem.

3.3. Proposed approach

This study introduces the concrete steps of the ANLNS algorithm, which is applied to solve the CVRP. The initial solution stage is expressed in subsection 3.3.1. Then, the general principles of the RWS are explained in subsection 3.3.2. The destroy operators are introduced in subsection 3.3.3. The repair operators are introduced in subsection 3.3.4. Furthermore, NSs and neutrosophic utility functions are used to assign an appropriate score for the actual contribution of the used operator in subsection 3.3.5. The acceptance and stopping criteria are introduced in subsection 3.3.6.

3.3.1. Initial solution

A high-quality initial feasible solution has a crucial impact on the final best solution, which aids the algorithm rapidly to converge, prevents falling into the local best solution, and reduces computation time. Due to its simplicity, efficiency, and scalability, this study chooses the greedy insertion to generate the high-quality initial feasible solution. Before initialization, customers N with different demands are known. In general, under the constraint of the maximum loading capacity Q_v of vehicle v, the greedy insertion algorithm chooses the customer nearest to the current customer as the next direction for transportation. According to reference [29], the main step for the initial solution is described in Algorithm 2.

ALGORITHM 2 Initial solution.

1: **Input**: a set of customers N, the maximum loading capacity Q_v;
2: create an empty set of routes P;
3: create an empty boole set V of uninvited customers;
4: create an empty current route R and set the start point as the depot;
5: **Repeat**
6: insert one customer into the R, all constraints are suitable;

7: set the status of the visited customer as true in the V;
8: **If** the total demand of customers in the R exceeds Q_v **then**
9: add the depot to the tail of R, put R into the P, and set R empty again;
10: return to step 4;
11: **Else**
12: return to step 6;
13: **End if**
14: **Until** all customers are served in the N as well and all access status is true in the V, respectively;
15: get the set of routes P as the initial solution x;
16: **Output**: the initial feasible solution x.

3.3.2. Selection of operators

The combination of different destroy operators and repair operators not only enhances the diversity of solutions but also balances the relationship between exploration and exploitation effectively. For each iteration, this study has k operators with weight $\omega_i, i \in \{1, \dots, k\}$, and uses the RWS to choose the operators based on the probability φ. Thereafter, the φ of the jth operator is given as

$$\varphi = \frac{\omega_j}{\sum_{i=1}^{k} \omega_i}. \tag{8}$$

Noting the destroy operator is selected independently before the repair operator. Specially, before the algorithm starts, the weight of each operator is initialized to 1.

3.3.3. Removal operators

This study describes three types of destroy operators in detail, namely random removal, Shaw removal, and worst removal. For each iteration, these heuristics remove the number of customers $q = \mu \cdot N$ from the solution x_s. Among them, N indicates the total number of customers in $x_{current}$. Next, the removed customers are stored in set D.

1. Random removal

The main principle is to remove q customers from x_s randomly. Next, these removed customers are stored in set D.

2. Shaw removal

The Shaw removal as a new local search to address the VRP. The core idea of Shaw removal is to remove the customer that is similar in geographical location, demand, or being visited at similar times [30-31]. To do so, a relatedness function $R(i, j)$ is defined to better measure the similarity of customer i and customer j. Therefore, the function is defined as

$$R(i, j) = \alpha d_{ij} + \delta \left| q_i - q_j \right|, \tag{9}$$

where d_{ij} presents the distance of customer i and customer j as the first term of $R(i,j)$. $|q_i - q_j|$ is the second term of $R(i,j)$. Meanwhile, the parameters q_i and q_j denote the demand of customer i and customer j, respectively. The above two terms are weighted using weights α, and δ, respectively. According to reference [32], the entire Shaw removal process is expressed by Algorithm 3. In particular, the parameter P_1 is added to realize the process of selection randomly, thereby increasing the diversity of solutions.

ALGORITHM 3 Shaw removal.

Function Shaw removal (solution x_s, randomized selection parameter P_1)
  create an empty set D to store the removed customers;
  a randomly selected customer ψ from $x_{current}$;
  assume that the set D contains the customer ψ;
  while $|D| < q$ **do**
    a randomly selected customer RN from D;
    a set L containing the similarity $R(i,j)$ of RN with all customers from x_s not in D;
    sort L in ascending order according to the values of $R(i,j)$;
    generate a random number γ, where $\gamma \in [0,1]$;
    remove customer $L[\gamma^{P_1}|L|]$ from L and insert it into D;
  End while
End function

3. Worst removal

The fundamental principle of the worst removal is to select and remove the customer that brings high costs. At the beginning, let $\Delta_i = f(x_s) - f_{-i}(x_s)$ be the cost of customer i. Among them, $f_{-i}(x_s)$ is the cost for x_s after removing customer i. Note that the high Δ_i equals the longest distance from customer i to customer j in this study. Therefore, get the cost Δ_i for each customer and sort it in descending order. Subsequently, according to the parameter q, the worst removal uses the randomized selection parameter P_2 to remove customers and store them in set D. The procedure of worst removal is referred to the reference [32]. Meanwhile, the main steps are shown in Algorithm 4.

ALGORITHM 4 Worst removal.

Function Worst removal (solution x_s, randomized selection parameter P_2)
  create an empty set D to store the removed customers;
  record the total distance of solution $x_{current}$ as $f(x_s)$;
  while $q > 0$ **do**
    calculate the cost Δ_i of each customer in x_s;
    a set L containing the cost Δ_i of each customer in x_s;
    sort L in ascending order according to the values of Δ_i;
    generate a random number γ, where $\gamma \in [0,1]$;
    remove customer $L[\gamma^{P_2}|L|]$ from L and insert it into D;
    $q \leftarrow q - 1$;
  End while
End function

3.3.4. Insertion operators

In this subsection, each removed customer reinserts the broken x_s to construct the new solution. The insertion operators are regret insertion and best insertion, respectively. Then, the repair work is not completed until all removed customers are inserted in each iteration.

1. Regret insertion

The main work of this operator is to reinsert the removed customer into the lowest insertion cost position so that there is not most regretful. According to the reference [33], the regret insertion calculates the lowest insertion cost Δ_1 to determine the corresponding optimal position in route j_1. Then, the second lowest insertion cost Δ_2 follows the same approach to do so. For each user, the regret value is calculated as $RC = \Delta_2 - \Delta_1$. After that, the high regret value for customer i is selected for reinsert operation, aiming to realize minimization for the objective cost.

2. Best insertion

As indicated by reference [34], the best insertion position is selected by using the best insertion strategy, which aims to generate the lowest insertion cost for the removed customer. Hence, at each iteration, record the insertion cost Δ_i that the removed customers i reinsert each position of all routes in the broken solution. Then, find the lowest insertion cost to determine the best insertion position.

3.3.5. Adaptive neutrosophic weight adjustment

As described in Algorithm 1, the scores for each operator are assigned by σ_1, σ_2, σ_3, or σ_4, respectively in the four different situations. Among them, these scores are integers satisfying $\sigma_1 > \sigma_2 > \sigma_3 > \sigma_4 > 0$. However, according to the quality of x_{new}, the ALNS simply increases the same score for different operators in each scoring situation, rather than assigning the score corresponding to the specific improvement degree of x_{new}. To do so, this study introduces NSs, designing the neutrosophic utility function and the score function to assign the appropriate score for each operator. Besides, the adaptive neutrosophic weight adjustment follows the basic principle of reference [30].

The process of ANLNS is divided into many segments. Each segment iterates 100 times. The score of all operators equals 0 before each segment runs. During the segment, let NN_1, NN_2, NN_3, and NN_4 be the number of times the operator scores under the four different scoring levels. Specially, this way applies to each operator in the segment, as shown in Table 1.

Table 1. The number of times for operators with four different situations.

Parameter	Score level	Description
NN_1	σ_1	If x_{new} is better than x_{best}. Then, NN_1 increases 1.
NN_2	σ_2	If x_{new} is worse than x_{best} and better than $x_{current}$. Then, NN_2 increases 1.
NN_3	σ_3	If x_{new} is worse than $x_{current}$, x_{new} is accepted with a certain probability. Then, NN_3 increases 1.
NN_4	σ_4	If x_{new} is worse than $x_{current}$, x_{new} is unacceptable. Then, NN_4 increases 1.

Therefore, to analyze the uncertainty of operator performance, the neutrosophic number $NN = \langle N_T, N_I, N_F \rangle$ is constructed by using NN_1, NN_2, NN_3, and NN_4. For convenience, N_T, N_I, and N_F are defined as

$$N_T = \frac{\lambda NN_1 + NN_2}{\lambda NN_1 + \sum_{i=2}^{4} NN_i}, N_I = \frac{NN_3}{\lambda NN_1 + \sum_{i=2}^{4} NN_i}, N_F = \frac{NN_4}{\lambda NN_1 + \sum_{i=2}^{4} NN_i}, \tag{10}$$

where N_T indicates the truth-membership degree that x_{new} is better than x_{best} or $x_{current}$ in the segment of ANLNS. N_I indicates the indeterminacy-membership degree that x_{new} is worse than $x_{current}$, but x_{new} is accepted with a certain probability in the segment of ANLNS. N_F indicates the falsity-membership degree that x_{new} is unacceptable in the segment of ANLNS. And, λ is a correction factor. Subsequently, in order to express the degree to which ANLNS prefers the operator, inspired by reference [35], this study constructed a novel neutrosophic utility function by introducing NN into the quadratic utility function. Thus, the neutrosophic utility function is obtained as follows:

$$U = \left(N_T + N_F\right)\left(N_T - N_F + 1\right)/2. \tag{12}$$

It is noteworthy that the parameters NN_1, NN_2, NN_3, and NN_4 have the upper limits of the scores level, which are σ_1, σ_2, σ_3, and σ_4 respectively. Due to the constraints of the conditions, the lower limits of the scores for each parameter are also σ_2, σ_3, σ_4, and 0 respectively. In order to assign an appropriate score, four different score intervals $(\sigma_2, \sigma_1]$, $(\sigma_3, \sigma_2]$, $(\sigma_4, \sigma_3]$, and $(0, \sigma_4]$ are redistributed to correspond to the four different scoring situations respectively. Therefore, the new score function for each operator is obtained as

$$\sigma_c^* = \begin{cases} U\left(\sigma_c - \sigma_{c+1}\right) + \sigma_{c+1}, c \in \{1,2,3\}, \\ U\left(\sigma_c - 0\right) + 0, \qquad c = 4, \end{cases} \tag{13}$$

where σ_c^* indicates the new score of the operator in the cth scoring situations. $\sigma_c - \sigma_{c+1}$ indicates the size of the score interval consisting of the score σ_c and σ_{c+1}. In particular, σ_{c+1} is equal to 0 when $c = 4$. The neutrosophic utility value U serves as a weight, which influences the magnitude of the score change that the operator has within the score interval. After 100 iterations of segment s, the weight for all operators i is updated to apply in segment $s + 1$, and the $\omega_{i,s+1}$ is defined as

$$\omega_{i,s+1} = \omega_{i,s}\left(1-\phi\right) + \phi\frac{\eta_i}{\tau_i}, \tag{14}$$

where, $\phi \in (0,1)$ is a reaction factor. η_i is the total score obtained by operators i in segment s. τ_i indicates the utilized times for the operator i in the segment s.

3.3.6. Acceptance and stopping criteria

In ANLNS, if $f(x_{new}) > f(x_{current})$, the simulated annealing acceptance criterion is adopted. x_{new} with a certain probability $e^{-(f(x_{new})-f(x_{current}))/T}$ is accepted. T is the current temperature, which equals the start temperature T_0 before this segment starts. With the continuous iteration in each segment, T is decreased with the cooling rate $\beta \in (0,1)$ such that $T = T \cdot \beta$. It is increasingly difficult for ANLNS to accept the probability of poor solutions. To ensure the continuous convergence of the algorithm, the setting of T_0 is extremely critical. T_0 is $-t \cdot |f(x_{current})|/lnT_p$, where t is the start temperature control parameter. lnT_p expresses that if x_{new} is worse than $x_{current}$ at T_0, x_{new} is accepted with a probability of T_p%.

4. Results and Discussion

For convenience, this section contains four subsections, which are detailed as follows. Subsection 4.1 introduces details about a set of benchmark instances. Subsection 4.2 gives the setting parameter of ANLNS. Subsection 4.3 applies the CVRP benchmark instances to test the effectiveness and practicability of ANLNS. Subsection 4.4 provides a comparative analysis to verify the robustness and validity of ANLNS.

4.1. Benchmark

In this subsection, to verify the effectiveness of the proposed ANLNS, this study selected representative instances C1-C3, C6-C8, C12, and C14 in reference [36] as case studies. More specifically, these instances are categorized into three scales based on the number of customers served. Among them, instances C1 and C6 are classified as small-scale, each serving 50 customers. Instances C2 and C7 are classified as medium-scale, each serving 75 customers. Finally, instances C3, C8, C10, C12, and C14 are classified as large-scale, each serving 100 customers. Then, instances C1, C2, C3, and C12 are constrained by the loading capacity. In contrast, instances C6, C7, C8, and C14 are subject to additional constraints, including the maximum traveling distance and drop time. Meanwhile, the customer demands vary for each instance. Furthermore, instances C1-C3, C6-C8, C12, and C14 are designed based on the customer distribution, where instances C1-C3, and C6-C8 are randomly distributed, and instances C12 and C14 are clustered distributed. In detail, the information for CVRP benchmark instances please refers to Table 2.

Table 2. The information for CVRP benchmark instances.

Instances	Number of customers	Loading capacity	Maximum traveling distance	Drop time
1	50	160	999999	0
2	75	140	999999	0
3	100	200	999999	0
6	50	160	200	10
7	75	140	160	10
8	100	200	230	10
12	100	200	999999	0
14	100	200	1040	90

4.2. Parameters setting

The experimental results are presented for ANLNS in the CVRP benchmark. The ANLNS is coded in Python with the PyCharm. The experiments are all conducted on a Windows 11 environment, using an Intel Core i7-14650HX CPU clocked at 2.2 GHz with 16 GB RAM.

To optimize the algorithm's performance, this study defines several sets of algorithm parameters anticipated to bring high quality solutions, and these parameters are tested on several instances to keep a balance between solution quality and runtime. This study identifies that the randomness parameters of Shaw removal and worst removal, the maximum number of iterations, and the start temperature, have a great impact on the performance of the algorithm. Therefore, this study tunes these parameters, and the remaining parameters are adopted as reference [30]. Then, the operating parameters of ANLNS are explained in Table 3.

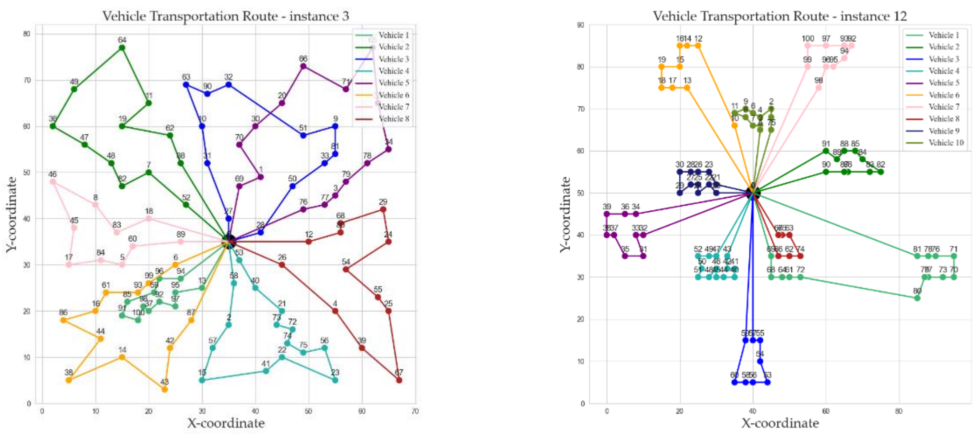
Table 3. The operating of parameters of ANLNS

Description	Symbol	Numerical value
the maximum number of iterations	max_item	1000
the degree of destruction	μ	0.5
the weighted parameter of distance item for $R(i,j)$	α	0.75

in Shaw removal		
the weighted parameter of loading capacity item for $R(i,j)$ in Shaw removal	δ	0.1
the randomness parameters of Shaw removal	P_1	4
the randomness parameters of the worst removal	P_2	4
adaptive adjustment weight parameter	σ_1, σ_2, σ_3, σ_4, ϕ	30, 19, 13, 9，0.5
correction factor	λ	0.5
start temperature control parameter	t	0.07
cooling rate	β	0.99975
acceptable probability	T_p	0.5

4.3. Case application

Each instance varies in characteristics. Specifically, instances C3 and C12 have identical customer and demand of 100 and 200, respectively, and the distribution of customers is different in both instances. Therefore, this study selects the above instances to represent large-scale and more complex situations, demonstrating the applicability and the solving capability of ANLNS in diverse situations. The vehicle transportation routes of instances 3 and 12 are shown in Figure 2. Meanwhile, this study displays the vehicle transportation routes for the remaining instances in Appendix A.



(a) (b)

Figure 2. (**a**) The vehicle transportation route of instance; (**b**) The vehicle transportation route of instance 12.

It is observed from Figure 2 that the transportation routes using the ANLNS are clear. Meanwhile, Figure 2 highlights that the number of vehicles in instances 3 and 12 is as low as 8 and 10, respectively. Therefore, under varying scales of transportation situations, it is found that the vehicle transportation routes planned by ANLNS are frequently effective. By applying the formula $(C_{actual}/C_{max}) \cdot 100\%$ to calculate the vehicle loading rate of instances 3 and 12, as shown in Figures 3-4. Then, Figures 3-4 indicate that almost all of the loading rate exceeds 0.75. The loading rate of each vehicle in instance 3 is as high as 99.50%, 98.00%, 97.50%, 84.50%, 99.50%, 99.50%, 67.50%, and 83.00%, respectively. The loading capacity rate of each vehicle in instance 12 is as high as 100.00%, 85.00%, 100.00%, 80.00%, 100%, 100.00%, 95.00%, 75.00%, 85.00%, and 85.00%, respectively. On the other hand, Figures 3-4 reveal that the no-load travel distance is reduced and the efficiency of transportation is improved through ANLNS's excellent goods allocation. Then, the total transportation distance is greatly reduced. In summary, it is observed from Figures 2-4 that ANLNS has significant potential in route planning and goods allocation.

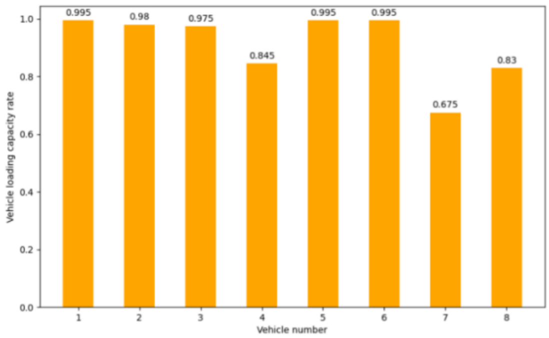


Figure 3. The vehicle loading rate in instance 3.

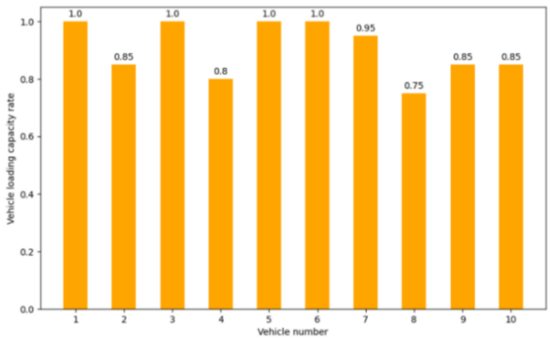
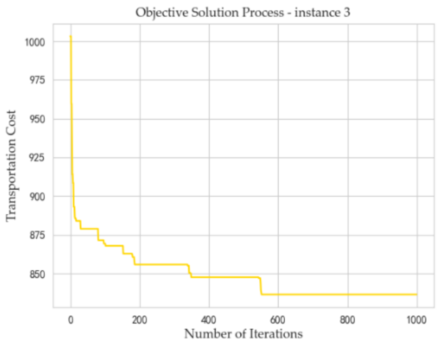
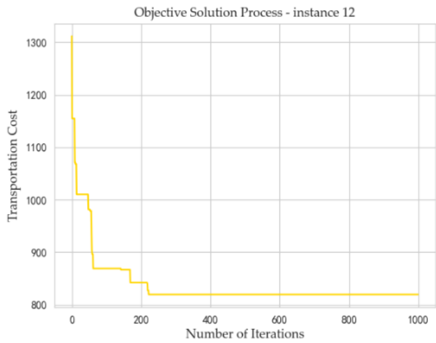


Figure 4. The vehicle loading rate in instance 12.

Figure 5 shows that ANLNS converges quickly after about 450 and 200 iterations, respectively. In the initial stage, ANLNS has a high convergence speed. Therefore, ANLNS effectively converges to the best solution of the ideal area. The current transportation routes are effective and realistic.



(a)



(b)

Figure 5. (**a**) The convergence behavior of instance 3; (**b**) The convergence behavior of instance 12.

4.4. Comparative analysis

For convenience, the results of experiments for ANLNS, ALNS, and Akpunar et al. [37] are compared and analyzed to verify the robustness and reliability of the ANLNS. Subsequently, this study uses the degree of destruction, which is 0.3, 0.5 and 0.7, respectively. For all instances, each algorithm runs ten times, and each time contains 1000 iterations. Tables 3-5 below show the comparison results where tables include the benchmark labels in the first column. Within the "ANLNS", "ALNS", and "Akpunar et al.' Hybrid ALNS" columns, the sub-column "BS" represents the best solution from ten algorithm runs. The sub-columns "AS" and "WS" display the average and worst solutions from the same ten runs. The "SD" denotes the standard deviation. Additionally, select the best solutions for each instance from ANLNS, ALNS, and Akpunar et al.' Hybrid ALNS, and highlight them in bold within the BS column.

Table 3. Compared results at a degree of destruction of 0.3.

Instance	ANLNS				ALNS				Akpunar et al.' Hybrid ALNS			
	BS	AS	WS	SD	BS	AS	WS	SD	BS	AS	WS	SD
C1	**527.94**	533.49	539.90	4.78	528.14	534.82	544.90	4.79	528.14	536.04	546.75	5.80
C2	**863.42**	877.80	886.29	7.51	869.69	882.87	892.87	7.70	864.34	877.45	887.28	6.79
C3	836.43	842.35	854.02	5.02	**830.48**	840.44	849.96	5.56	831.41	838.13	844.65	4.14
C6	526.22	531.81	541.11	4.84	529.43	538.19	541.11	3.16	**524.61**	535.20	542.47	5.74
C7	**857.94**	875.40	884.83	7.86	864.95	882.90	897.10	8.40	866.34	877.86	895.60	7.43
C8	836.20	839.73	843.45	2.16	832.62	840.76	846.19	4.46	**831.98**	839.92	850.89	4.69
C12	819.97	829.08	838.66	6.72	825.38	846.16	874.18	14.62	**819.56**	825.79	844.19	7.75
C14	**819.56**	833.95	854.38	13.93	826.07	843.98	858.30	9.45	819.60	835.39	864.22	14.68

Table 4. Compared results at a degree of destruction of 0.5.

Instance	ANLNS				ALNS				Akpunar et al.' Hybrid ALNS			
	BS	AS	WS	SD	BS	AS	WS	SD	BS	AS	WS	SD

C1	**529.43**	538.34	543.05	3.57	539.22	542.06	547.67	2.64	535.00	540.48	543.44	2.42
C2	**860.12**	875.46	886.01	7.46	868.98	885.40	898.86	8.24	865.83	876.76	896.02	8.95
C3	**835.72**	849.00	856.70	6.24	844.98	853.66	861.84	4.99	844.26	850.06	857.26	3.68
C6	**528.14**	538.96	545.64	5.25	537.00	543.63	555.33	4.91	**528.14**	538.57	543.31	4.91
C7	**863.95**	877.02	886.75	6.54	870.75	886.62	901.49	8.68	868.51	876.25	886.96	6.83
C8	**836.55**	848.16	856.40	5.48	849.72	854.57	860.73	3.44	841.07	849.57	858.58	6.08
C12	**819.60**	821.36	822.95	1.04	821.92	831.38	847.46	8.23	**819.60**	821.23	823.81	1.37
C14	**819.60**	822.59	829.57	2.79	821.15	831.69	852.10	10.01	**819.60**	821.00	824.21	1.23

Table 5. Compared results at a degree of destruction of 0.7.

Instance	ANLNS				ALNS				Akpunar et al.' Hybrid ALNS			
	BS	AS	WS	SD	BS	AS	WS	SD	BS	AS	WS	SD
C1	541.31	543.74	548.76	2.74	542.51	547.73	560.77	5.195	**540.47**	545.77	557.29	4.83
C2	**875.95**	907.00	934.81	16.88	884.84	910.26	935.67	16.82	886.17	906.78	929.12	10.95
C3	**837.69**	853.62	865.09	7.82	849.51	857.18	868.53	5.39	840.93	851.23	861.35	5.69
C6	1072.97	1087.61	1102.90	8.14	1067.83	1103.59	1114.37	12.83	**533.63**	542.38	547.90	3.84
C7	1377.23	1404.67	1421.62	12.48	1400.63	1417.16	1436.12	11.70	**875.03**	911.76	941.99	18.00
C8	**536.13**	544.35	552.51	4.77	539.46	544.51	548.28	2.49	842.63	851.91	860.34	5.83
C12	869.94	907.80	929.44	16.07	882.15	917.41	938.27	17.72	**820.48**	821.66	823.71	1.03
C14	846.84	853.05	863.83	4.56	839.02	856.23	870.43	9.09	**819.97**	821.55	823.25	0.87

Tables 3-5 conclude that the best solution obtained by ANLNS is better than the expected result. In the experimental results from 8 instances under three different degrees of destruction, ANLNS obtains 15 best solutions. Specifically, when the degree of destruction is 0.3, 0.5, and 0.7, ANLNS considerably obtains 4(50.0%), 8(100.0%), and 3(37.5%) best solutions respectively. Under the same conditions, the total number of best solutions for ALNS, and Akpunar et al.' Hybrid ALNS is 1 and 11, respectively. Therefore, Tables 3-5 indicate that ANLNS has a robust performance advantage. Meanwhile, it shows that ANLNS possesses a good global search capability and diversifies the search.

In addition, tables 3-5 show the average solution and standard deviation obtained by each algorithm in all instances. it is observed that the average solution and standard deviation obtained by ANLNS are better than the other two algorithms. Even if some instances show that the standard deviation of ANLNS is volatile. However, as a whole, it shows that the ANLNS has a stable and strong ability to handle complex and diverse instances of different scales.

Table 6 below shows the execution time of the ANLNS, ALNS, and Akpunar et al.' Hybrid ALNS algorithm in seconds. Among them, the "DT" indicates the execution time of the algorithm under the degree of destruction is 0.3. The "DF" indicates the execution time of the algorithm under the degree of destruction is 0.5. The "DS" indicates the execution time of the algorithm under the degree of destruction is 0.7.

Table 6. Compared execution time at different degrees of destruction.

Instance	ANLNS			ALNS			Akpunar et al.' HybridALNS		
	DT	DF	DS	DT	DF	DS	DT	DF	DS
C1	9.93	16.05	19.72	9.96	15.98	19.73	10.96	17.07	21.23
C2	23.32	36.82	38.46	23.82	38.77	46.53	25.67	41.80	39.84
C3	62.32	111.5	139.93	62.84	112.26	118.14	68.22	117.48	146.32
C6	9.83	15.74	422.45	10.05	15.67	42.52	11.18	29.50	34.86
C7	23.82	38.23	935.93	23.8	39.88	98.47	26.47	57.93	39.53
C8	63.36	110.62	19.72	63.25	112.02	260.32	124.41	116.96	148.18
C12	56.60	97.62	37.32	57.87	100.43	238.24	106.86	102.96	134.65
C14	56.45	97.79	141.52	57.28	99.12	224.61	61.85	211.02	290.83

Table 6 expresses that ANLNS is more efficient than ALNS and Akpunar et al.' Hybrid ALNS since the run time of ANLNS is significantly smaller than ALNS. Meanwhile, the running time by ANLNS for instance 12 with degree of destruction 0.3 is 56.60 seconds, while the corresponding ALNS takes 57.87 seconds and Akpunar et al.' Hybrid ALNS takes 106.86 seconds. Even if the instance scale and degree of destruction keep increasing, in some cases, ANLNS may be slightly worse than ALNS and Akpunar et al.' Hybrid ALNS. But on the whole, the result shows that the proposed ANLNS is competitive with ALNS and Akpunar et al.' Hybrid ALNS in CVRP.

4.5. Discussion

The traditional ALNS and improved ALNS provide a way of dealing with some practical work. However, these studies do not consider ALNS existing in the fuzziness and uncertainty of assessing and scoring operators. Therefore, this study proposes a novel ANLNS algorithm to solve the above problem. The following details are provided.

Firstly, the NS, which keeps track of the number of times each operator scores in different situations, is integrated into the framework of ALNS. This has been of great help to the ANLNS in evaluating each operator's performance and actual contribution. Secondly, the neutrosophic utility function and score function are proposed in this study, which is helpful to assign scores reasonably within a reasonable interval. Finally, in this study, ANLNS tested three representative instances of varying scales selected from a set of CVRP benchmark instances.

Compared with previous ALNS-related studies, this study found that the ANLNS algorithm not only showed a more detailed and accurate evaluation effect on operator performance but also effectively measured the algorithm's preference for different operators and reduced the existence of subjective factors. Subsequently, in the results section, the experimental results and comparative analysis emphasize the robustness and reliability of ANLNS for solving optimization problems.

However, ANLNS exhibits limitations in exhaustively exploring the solution space. Therefore, future work includes continuing to evaluate whether these methods are retained in the ANLNS algorithm and considering the introduction of artificial intelligence to effectively improve the algorithm.

5. Conclusion

As described before, in the adaptive weight adjustment stage, the score of operators exists fuzziness and uncertainty. To address this problem, this study proposes an ANLNS algorithm. The main contributions are as follows.

Firstly, this study records the number of times that the operator scores under four different situations, and constructs NSs to quantify these number of times. The application of NSs facilitates the ANLNS to analyze and evaluate operator performance more comprehensively.

Secondly, this study proposes a novel neutrosophic utility function and score function. Then, the proposed function evaluates the ANLNS's preference and assigns the appropriate score for different operators.

Finally, this study verifies a solving ability of ANLNS through CVRP. Moreover, the provided best solution offers a novel way for solving CVRP and other situation optimization problems.

In summary, the effect of this study is that ANLNS can adjust the operator weight to a relatively reasonable level, and make the ANLNS workflow more reasonable and efficient. However, the limitation of this study is that the ANLNS has not thoroughly explored the solution space to find potential high-quality solutions. Therefore, in the future, this research will consider the solution space of exploration and exploitation for potential best solutions. Therefore, combined with the decision tree and attention mechanisms, the research would provide a reasonable, effective, and practical algorithm.

Funding: Fangwei Zhang's work is supported by the Natural Science Foundation of Shandong Province (ZR2021MG003).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

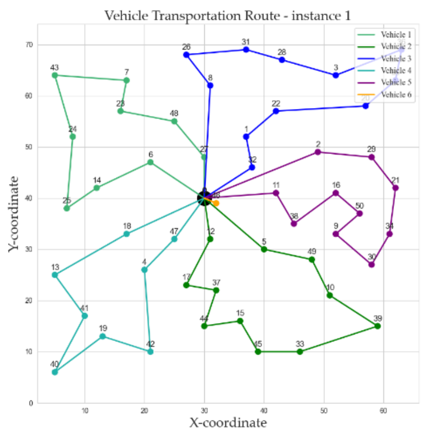


Figure A1

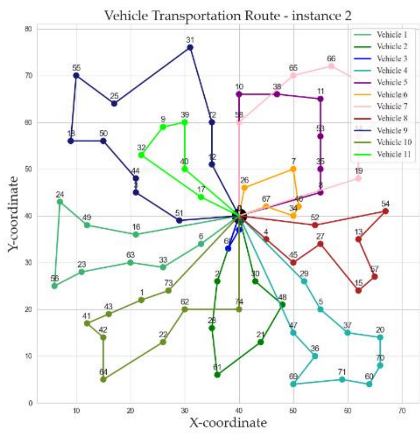


Figure A2

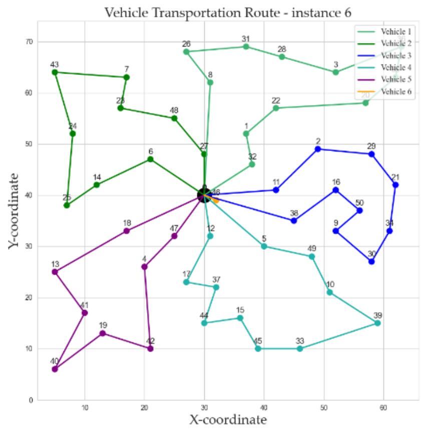


Figure A3

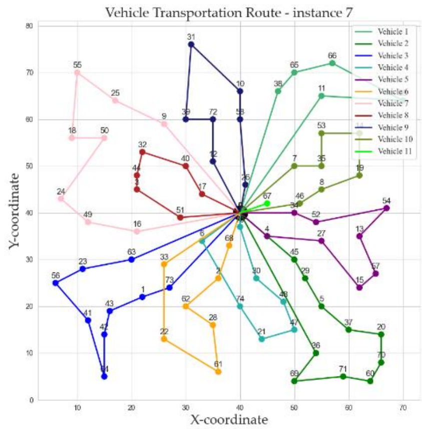


Figure A4

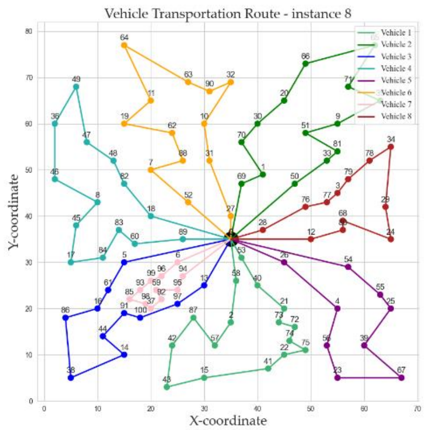


Figure A5

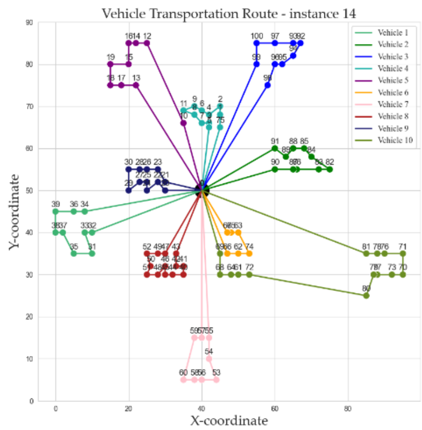


Figure A6

References

1. Demir, E.; Bektaş, T.; Laporte, G.; An adaptive large neighborhood search heuristic for the pollution-routing problem. *European journal of operational research*, **2012**, 223(2), 346-359.
2. Dang, Q.V.; Pham, K.; Design of a footwear assembly line using simulation-based ALNS. *Procedia CIRP*, **2016**, 40, 596-601.
3. Pisinger, D.; Ropke, S.; A general heuristic for vehicle routing problems. *Computers & operations research*, **2007**, 34(8), 2403-2435.
4. Cai, L.; Wang, X.; Luo, Z.X.; Liang, Y.J.; A hybrid adaptive large neighborhood search and tabu search algorithm for the electric vehicle relocation problem. *Computers & Industrial Engineering*, **2022**, 167, 108005.
5. Amiri, A.; Amin, S.H.; Zolfagharinia, H.; A bi-objective green vehicle routing problem with a mixed fleet of conventional and electric trucks: Considering charging power and density of stations. *Expert Systems with Applications*, **2023**, 213, 119228.
6. Fathollahi-Fard, A.M.; Wong, K.Y.; Aljuaid, M.; An efficient adaptive large neighborhood search algorithm based on heuristics and reformulations for the generalized quadratic assignment problem. *Engineering Applications of Artificial Intelligence*, **2023**, 126, 106802.

7. Chen, C.; Demir, E.; Huang, Y.; An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European journal of operational research*, **2021**, 294(3), 1164-1180.
8. Rolim, G.A.; Nagano, M.S.; de Athayde Prata, B.; Formulations and an adaptive large neighborhood search for just-in-time scheduling of unrelated parallel machines with a common due window. *Computers & Operations Research*, **2023**, 153, 106159.
9. Rashno, E.; Minaei-Bidgoli, B.; Guo, Y.; An effective clustering method based on data indeterminacy in neutrosophic set domain. *Engineering Applications of Artificial Intelligence*, **2020**, 89, 103411.
10. Saeed, M.; Rahman, A.U.; Arshad, M.; Dhital, A.; A novel approach to neutrosophic hypersoft graphs with properties. *Neutrosophic Sets and Systems*, **2021**, 46, 336-355.
11. Ye, J.; MCGDM approach using the weighted hyperbolic sine similarity measure of neutrosophic (indeterminate fuzzy) multivalued sets for the teaching quality assessment of teachers. *Neutrosophic Sets and Systems*, **2022**, 48(1), 1.
12. Smarandache, F.; New Types of Topologies and Neutrosophic Topologies. *Neutrosophic Systems with Applications*, **2023**, 1, 1-3. https://doi.org/10.61356/j.nswa.2023.1
13. Barbosa, R.P.; Smarandache, F.; Pura Vida Neutrosophic Algebra. *Neutrosophic Systems with Applications*, **2023**, 9, 101-106. https://doi.org/10.61356/j.nswa.2023.68
14. Smarandache, F.; Foundation of the SuperHyperSoft Set and the Fuzzy Extension SuperHyperSoft Set: A New Vision. *Neutrosophic Systems with Applications*, **2023**, 11: 48-51. https://doi.org/10.61356/j.nswa.2023.95
15. Smarandache, F.; SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions. *Neutrosophic Systems with Applications*, **2023**, 12, 68-76. https://doi.org/10.61356/j.nswa.2023.115
16. Smarandache, F.; Foundation of Revolutionary Topologies: An Overview, Examples, Trend Analysis, Research Issues, Challenges, and Future Directions. *Neutrosophic Systems with Applications*, **2023**, 13, 45-66. https://doi.org/10.61356/j.nswa.2024.125
17. Al-Hijjawi, S.; Alkhazaleh, S.; Possibility Neutrosophic Hypersoft Set (PNHSS). *Neutrosophic Sets and Systems*, **2023**, 53(1), 7.
18. Smarandache, F.; Foundation of Appurtenance and Inclusion Equations for Constructing the Operations of Neutrosophic Numbers Needed in Neutrosophic Statistics. *Neutrosophic Systems with Applications*, **2024**, 15, 16-32. https://doi.org/10.61356/j.nswa.2024.1513856
19. Mishra, A.R.; Pamucar, D.; Rani, P.; Shrivastava, R.; Hezam, M. I.; Assessing the sustainable energy storage technologies using single-valued neutrosophic decision-making framework with divergence measure. *Expert Systems with Applications*, **2024**, 238, 121791.
20. Garey, M.R.; Johnson, D.S. *Computers and intractability: A guide to the theory of NP-completeness*, 1st ed.; W.H. Freeman and Company: New York, 1979; pp. 340.
21. Vidal, T.; Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, **2022**, 140, 105643.
22. Rezaei, B.; Guimaraes, F.G.; Enayatifar, R.; Haddow, P.C.; Combining genetic local search into a multi-population Imperialist Competitive Algorithm for the Capacitated Vehicle Routing Problem. *Applied Soft Computing*, **2023**, 142, 110309.
23. Jiang, H.; Lu, M.; Tian, Y.; Qiu, J.; Zhang, X.; An evolutionary algorithm for solving capacitated vehicle routing problems by using local information. *Applied Soft Computing*, **2022**, 117, 108431.
24. Wang, C.; Ma, B.; Sun, J.; A co-evolutionary genetic algorithm with knowledge transfer for multi-objective capacitated vehicle routing problems. *Applied Soft Computing*, **2023**, 148, 110913.
25. Chi, J.; He, S.; Pickup capacitated vehicle routing problem with three-dimensional loading constraints: Model and algorithms. *Transportation Research Part E: Logistics and Transportation Review*, **2023**, 176, 103208.
26. Wang, X.; Liang, Y.; Tang, X.; Jiang, X.Y.; A multi-compartment electric vehicle routing problem with time windows and temperature and humidity settings for perishable product delivery. *Expert Systems with Applications*, **2023**, 233, 120974.

27. Smarandache, F. *A Unifying Field in Logics: Neutrsophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability* 3rd ed.; Publisher: Am. Res. Press, American, 2007; pp. 233.
28. Srinivas, S.; Prabakaran, K.; Optimization of Single-valued Triangular Neutrosophic Fuzzy Travelling Salesman Problem. *Neutrosophic Sets and Systems*, **2023**, 60(1), 26.
29. Xue, S.Q.; Song, R.; He, S.W.; An, J.Y.; Wang, Y.M.; An improved adaptive large neighborhood search algorithm for the heterogeneous customized bus service with multiple pickup and delivery candidate locations. *Journal of Advanced Transportation*, **2022**, 2022. https://doi.org/10.1155/2022/1679469
30. Shaw, P. A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 1997, 46.
31. Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. Principles and Practice of Constraint Programming - CP98, Berlin, Heidelberg, Germany, 1998; Maher, M., Puget, JF, Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, Germany; pp. 417-431.
32. Ropke, S.; Pisinger, D.; An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, **2006**, 40(4), 455-472.
33. Shi, Y.; Liu, W.; Zhou, Y.; An adaptive large neighborhood search based approach for the vehicle routing problem with zone-based pricing. *Engineering Applications of Artificial Intelligence*, **2023**, 124, 106506.
34. Masson, R.; Lehuédé, F.; Péton, O.; The dial-a-ride problem with transfers. *Computers & Operations Research*, **2014**, 41, 12-23.
35. Johnstone, D.J.; Lindley, D.V.; Elementary proof that mean–variance implies quadratic utility. *Theory and Decision*, **2011**, 70, 149-155.
36. Christofides, N.; Mingozzi, A.; Toth, P. The Vehicle Routing Problem. In *Combinatorial Optimization*, 1st ed.; Christofides, N., Mingozzi, A., Toth, P., Sandi, C., Eds.; Wiley: Chichester, UK, 1997; Volume 10, pp. 325–338.
37. Akpunar, Ö.Ş.; Akpinar, Ş.; A hybrid adaptive large neighbourhood search algorithm for the capacitated location routing problem. *Expert systems with applications*, **2021**, 168, 114304.

Received: March 10, 2024. Accepted: July 26, 2024