Application Of The Ant Colony Algorithm For Solving The Fuzzy Traveling Salesman Problem

Eugene Ivohin[1,*], Oleksiy Oletsky[2], Konstantin Yushtin[1], Valeriy Gavrilenko[3] and Maksym Boguslavskyi[3]

[1] *Taras Shevchenko National University of Kyiv, Ministry of Education of Ukraine, 60 Volodymyrska Str., Kyiv, 01033, Ukraine*

[2] *National University of Kyiv-Mohyla Academy, Ministry of Education of Ukraine, 2 Skovorody Str., Kyiv, 04070, Ukraine*

[3] *National Transport University, 1 M. Omelianovycha-Pavlenka Str., Kyiv, 01010, Ukraine*

Abstract

The traveling salesman problem (TSP) is a classical combinatorial optimization problem that involves finding the shortest or fastest route among a set of cities. To formalize the uncertainty and imprecision in input data, often caused by subjective evaluations of the travel time intervals, this paper employs fuzzy numbers. The form of these fuzzy numbers is based on a Gaussian-like approach. This work examines the specifics of applying the ant colony optimization (ACO) algorithm and proposes an approach for its optimal use. The impact of the algorithm's parameters on the quality of the approximated best solution is analyzed. The problem is illustrated with numerical examples involving a sufficiently large number of cities in the transportation network.

Keywords

fuzzy traveling salesman problem, ant colony optimization method, trapezoidal fuzzy numbers, defuzzification, performance evaluation

1. Introduction

One of the promising directions in scientific and practical research of social and information processes is based on the use of mathematical methods that incorporate the principles of natural decision-making mechanisms. Swarm intelligence is a relatively new technological approach formalized through the analysis of the social behavior of animals and insects. In particular, observations of ants have led to the development of several methods and techniques, the most studied and successful of which is the general optimization method known as *Ant Colony Optimization* (ACO). The simulation of the self-organization of an ant colony forms the basis of ant optimization algorithms—a new and promising method of natural computation. Other natural prototypes for optimization methods include the behavior of dragonflies (*Dragonfly Algorithm*, DA), bees (*Bee Algorithm*, BA), termites (*Termite Algorithm*), fish (*Fish Swarm Algorithm*, FSO), and wolves (*Wolf Pack Algorithm*, WSA).

The implementation of swarm intelligence refers to the method of solving various optimization problems using a group of agents that interact with each other based on simple rules, which guide the complex behavior of the entire system. Regarding its use in optimization techniques, the main advantage is the ability to find global optima in problems with a large number of parameters and constraints, as well as flexibility, scalability, distributed computation capabilities, and fault

8th International Scientific and Practical Conference Applied Information Systems and Technologies in the Digital Society AISTDS'2024, 2024, October 1, Kyiv, Ukraine

[*] Corresponding author.

ivohin@univ.kiev.ua (E. Ivohin); oletsky@ukr.net (O. Oletsky); gkons@univ.kiev.ua (K. Yushtin); vvgavrilenko1953@gmail.com (V. Gavrilenko); maxbogus09@gmail.com (M. Boguslavskyi)

0000-0002-5826-7408 (E. Ivohin); 0000-0002-0553-5915 (O. Oletsky); 0009-0001-9881-2343 (K. Yushtin); 0000-0001-9682-4204 (V. Gavrilenko); 0009-0000-1264-133X (M. Boguslavskyi)

tolerance. Systems based on swarm intelligence allow for the rapid discovery of efficient solutions under conditions of dynamic parameter changes, agent failures, and without requiring centralized control limitations.

Let us consider the application of swarm intelligence for solving optimization problems using ACO [1]. The first version of the algorithm in own Ph.D. work was proposed by Marco Dorigo in 1996 [2]. Ant algorithms have been widely used by scientists since the mid-1990s [3]. To date, good results have been achieved with ant optimization in solving complex combinatorial problems such as truck route optimization, graph coloring, the quadratic assignment problem, network schedule optimization, calendar planning problems, and others [4, 5]. Ant algorithms are particularly effective in online optimization of processes in distributed non-stationary systems, such as traffic distribution problems in telecommunications networks [6].

An ant colony can be viewed as a multi-agent system in which each agent (ant) operates autonomously according to defined rules. The behavior of each agent is governed by simple probabilistic rules. This principle aligns with the behavior of ants in the real world, where they work together to build nests, forage for food, and protect the colony. Studies [7, 8] have shown that based on the primitive behavior of individual agents, the collective system's behavior allows for achieving optimal results for various classes of problems.

The idea of the algorithm is based on the behavior of an ant colony, which finds a path to food that is close to optimal. The core of ant behavior is self-organization - a set of dynamic mechanisms by which the system achieves a global goal as a result of interactions at a low level.

Multiple interactions are implemented in the form of sequential iterative route searches conducted simultaneously by several ants. Each ant starts its movement randomly when leaving the nest in search of food. It is assumed that each agent does not follow a predetermined path or direction. This exploratory behavior allows ants to explore a wide area around the nest. A key aspect of ant behavior is their ability to leave chemical traces—pheromones—along their path. These pheromones serve as signals for other ants, indicating that the path has already been explored and is in use. When an ant finds food, it returns to the nest, leaving a pheromone trail that helps other ants find the path to the food. The amount of pheromone deposited by the ant on each segment of the route is inversely proportional to the length of that segment. The shorter the path found by the ant during the search, the more pheromone will be deposited on the corresponding segments (graph edges).

It should be noted that the use of only positive feedback leads to the rapid (premature) convergence of the algorithm, meaning that all ants will follow the same suboptimal route. To prevent oversaturation of the paths, pheromones evaporate over time, which implements negative feedback. This allows ants to adapt to changes in the environment (for example, the appearance of new food sources or obstacles on the paths).

Thanks to the mechanisms of inter-agent interaction, the system self-organizes, allowing individual ants to use optimal paths to food sources while enabling efficient solutions to resource allocation and search problems without centralized control.

From a mathematical perspective, the ACO model is described through the following basic components related to the behavior of both individual ants and the system as a whole: mechanisms for path formation; pheromone deposition to mark the traversed paths; pheromone evaporation; rules for an ant's path selection.

One of the problems for which a solution based on the ant colony algorithm can be proposed is the traveling salesman problem [9]. This study considers the adaptation of the ACO algorithm to solve the traveling salesman problem with fuzzy parameters of movements at the stages of the transport network, determined by fuzzy trapezoid numbers. The implementation of components of self-organizing behavior of ants for optimizing the route of the traveling salesman is proposed and computational experiments are carried out to determine the effectiveness of the developed method.

2. Task statement

The standard formulation of the traveling salesman problem is to choose the shortest (in length or time) closed path on a network of cities that passes through all cities exactly once. The number of possible options is $(n-1)!$, and if the stages of the route are symmetric, the number of unique routes is $(n-1)!/2$.

The problem is to find the optimal traveling salesman path, the determination of which requires a very large amount of computational resources, which leads to the need to use approximate algorithms such as ACO.

His In real logistics problems, the concept of the duration or cost of travel between individual points of the transport network cannot be fixed; they are determined approximately, often with the influence of the subjective factor based on estimates of time periods or the cost of movement along the route sections. This leads to the need to take into account uncertainty, its formalization based on different methods. One of the approaches used in this case is to involve fuzzy numbers and implement means of manipulating them.

Let's consider the formulation of the traveling salesman problem with a vaguely specified duration of movements at the stages of the transport and logistics network. In this case, it is necessary to find a cyclic permutation of the numbers of the cities to be visited by the traveling salesman, according to which the time spent will be minimal, taking into account the restriction on visiting each of the points no more than once. The mathematical formulation of the fuzzy problem of the traveling salesman can be written as follows: it is necessary to minimize, taking into account some method of comparing fuzzy numbers, the objective function

$$\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n}\tilde{t}_{ij}x_{ij}\,, \tag{1}$$

where the time costs for moving between points are given in the form of a matrix $\tilde{T}=\{\tilde{t}_{ij}\}$, $i,j=\overline{1,n}$, with elements in the form of fuzzy numbers [10], and the possible paths of movement between cities are determined by the matrix *X*, provided that the constraints are met

$$\begin{gathered}\sum\nolimits_{i=1}^{n}x_{ij}=1 \text{ for all } j=1,2,...,n\,,\\ \sum\nolimits_{j=1}^{n}x_{ij}=1 \text{ for all } i=1,2,...,n\,,\\ i-j+nx_{ij}\le n-1\,,\ 1\le i\ne j\le n\,,\\ x_{ij}=0 \text{ or } 1 \text{ for all } i,j=1,2,...,n\,.\end{gathered} \tag{2}$$

For the software implementation, the diagonal elements $\tilde{t}_{ii}$ of the matrix $\tilde{T}$ should be set to large positive numbers to ensure that the solution provides values $x_{ii}=0$ for all $i=1,2,....,n$. The fuzzy travel durations $\tilde{t}_{ij}$ between arbitrary cities $i,j=\overline{1,n}$, will be represented as trapezoidal fuzzy numbers.

Definition. A trapezoidal fuzzy number $\tilde{A}$ [11] is an ordered quadruple of real numbers $(a_1,a_2,a_3,a_4), a_1\le a_2\le a_3\le a_4$, for which the membership function $\mu_{\tilde{A}}(x)$ is defined as follows:

$$\mu_{\tilde{A}}(x)=\begin{cases}\dfrac{x-a_1}{a_2-a_1}, if\ \ a_1\le x\le a_2,\\ 1, \text{if} a_2\le x\le a_3,\\ \dfrac{a_4-x}{a_4-a_3}, if\ \ a_3\le x\le a_4.\end{cases} \tag{3}$$

If a Gaussian-based approach with corresponding characteristics is applied to the representation of a trapezoidal fuzzy number, then in the generalized case, the trapezoidal fuzzy number can be represented in a slightly different form:

$$\tilde{A}=(a_1,a_2,a_3,a_4)=([a_2,a_3],\alpha,\beta)=(m,w,\alpha,\beta), \tag{4}$$

where $m=\frac{a_2+a_3}{2}$ is the midpoint, $w=\frac{a_3-a_2}{2}$ is the half-width of the plateau, and the coefficients $\alpha=a_2-a_1$ and $\beta=a_4-a_3$ define the left and right distribution of the fuzzy number $\tilde{A}=(a_1,a_2,a_3,a_4)$. To operate with fuzzy numbers, it is necessary to define operations based on the description provided above. The midpoint is taken as the simple arithmetic mean of the plateau boundaries, and the left and right distributions are considered according to the lattice rule, where for arbitrary real numbers a,b we set $a\cup b=\max\{a,b\}$ and $a\cap b=min\{a,b\}$.

Then, for arbitrary trapezoidal fuzzy numbers $\tilde{A}=(m(\tilde{A}),w(\tilde{A}),\alpha_1,\beta_1)$ and $\tilde{B}=(m(\tilde{B}),w(\tilde{B}),\alpha_2,\beta_2)$ the operations of addition, subtraction, multiplication, and division can be defined, which in the general case are denoted by the symbol $\circ$:

$$\begin{gathered}\tilde{A}\circ\tilde{B}=(m(\tilde{A})\circ m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)\\=(m(\tilde{A})\circ m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\end{gathered} \tag{5}$$

Finally, we have:

$$\begin{gathered}\tilde{A}+\tilde{B}=(m(\tilde{A})+m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)\\=(m(\tilde{A})+m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)),\\\tilde{A}-\tilde{B}=(m(\tilde{A})-m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)\\=(m(\tilde{A})-m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)),\\\tilde{A}\times\tilde{B}=(m(\tilde{A})\times m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)\\=(m(\tilde{A})\times m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)),\\\tilde{A}\div\tilde{B}=(m(\tilde{A})\div m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)\\=(m(\tilde{A})\div m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\end{gathered} \tag{6}$$

To perform the operations of comparison and ranking of fuzzy numbers, we will use the method based on the median average value. In other words, if for each $\tilde{A}=(a_1,a_2,a_3,a_4)\in F(R)$ a ranking function $\mathfrak{R}:F(R)\to R$ with a median average value $\mathfrak{R}(\tilde{A})=\left[\left(\frac{a_2+a_3}{2}\right)+\left(\frac{\beta-\alpha}{4}\right)\right]$, is defined, then for any two trapezoidal fuzzy numbers $\tilde{A}=(a_1,a_2,a_3,a_4)$ and $\tilde{B}=(b_1,b_2,b_3,b_4)$ we have the following possible comparison options:

$$\begin{gathered}\text{- } \tilde{A}\succ\tilde{B} \text{ if and only if } \mathfrak{R}(\tilde{A})>\mathfrak{R}(\tilde{B});\\\text{- } \tilde{A}\prec\tilde{B} \text{ if and only if } \mathfrak{R}(\tilde{A})<\mathfrak{R}(\tilde{B});\\\text{- } \tilde{A}\approx\tilde{B} \text{ if and only if } \mathfrak{R}(\tilde{A})=\mathfrak{R}(\tilde{B}).\end{gathered} \tag{7}$$

The process of handling fuzzy numbers involves a defuzzification stage - converting a fuzzy result into a crisp (numerical) value. This is a crucial step in the methodology of applying fuzzy logic, especially in fuzzy control and fuzzy business logic tasks, where fuzzy solutions need to be transformed into specific events or numerical values. There are various defuzzification methods, the most common of which are the Center of Gravity (CoG) or centroid method, the mean of maxima method, and the maximum method. For comparison of research results, we will use the

Center of Gravity method. In this method, the defuzzification point is calculated as the center of gravity of the fuzzy set by the formula:

$$CoG = \frac{\int_{a_1}^{a_4} x \cdot \mu(x) dx}{\int_{a_1}^{a_4} \mu(x) dx}, \tag{8}$$

where x are the points of fuzzy number support, and $\mu(x)$ is the membership degree of each point.

3. Algorithm for finding the traveling salesman route

Path formation in the ASO model is described using a graph. Let $G=(V,E)$ represent a graph, where V – is the set of m vertices, and E is the set of edges. Each edge $(i,j)\in E$ is associated with two key parameters: the travel time t_{ij} along edge(i,j) (typically proportional to the path length D_{ij}) and the pheromone intensity τ_{ij} on edge (i,j), $i,j=m$. The preservation of pheromone levels is a key process for inter-agent interaction among ants. The pheromone intensity on edge (i,j) is updated based on the experience of the ants that have traversed this edge. The pheromone level is often updated using the following formula:

$$\tau_{ij}(s+1)=(1-\rho)\cdot\tau_{ij}(s)+\Delta\tau_{ij}(s), \tag{9}$$

where ρ is the pheromone evaporation coefficient, and $0<\rho<1$, a $\Delta\tau_{ij}(s)$ is the amount of pheromone deposited by the ants on edge (i,j) during iteration s, $s=0,1,2,...$

Pheromone evaporation decreases the intensity of pheromones on all edges, allowing the system to forget previous (possibly suboptimal) paths and adapt to changes. This prevents premature convergence to local optima. The selection of each step in the ant's path is based on probabilistic rules. When, at iteration s ant k is at vertex i, it chooses the next vertex j with a certain probability $P_{ij}^{k}(s)$, which depends on the pheromone intensity and visibility η_{ij}, the latter being inversely proportional to the path length $\eta_{ij}=1/D_{ij}$:

$$P_{ij}^{k}(s)=\frac{(\tau_{ij}(s))^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l\in J_i^k}(\tau_{il}(s))^{\alpha}(\eta_{il})^{\beta}}, ifj\in J_i^k,\ P_{ij}^{k}(s)=0, ifj\notin J_i^k, \tag{10}$$

where $\tau_{ij}(s)$ is the pheromone intensity on edge (i,j), $\eta_{ij}=1/D_{ij}$ is the visibility (inversely proportional to the path length), and α and β that control the influence of pheromone intensity and visibility, respectively. The sum in the denominator is calculated over all available edges (the set of available vertices J_i^k from vertex i). If $\alpha=0$, the most likely transition will be to the nearest cities. In classical optimization theory, this corresponds to the so-called greedy algorithm. If $\beta=0$, only pheromone reinforcement operates, which leads to the rapid termination of the algorithm as all ants converge on the same suboptimal solution. Note that the sum of all transition probabilities from city i over all possible options from the set J_i^k at iteration s equals 1:

$$\sum_{j\in J_i} P_{ij}^{k}(s)=1. \tag{11}$$

The main objective of the ant algorithm is to minimize the path length L, which is the sum of the lengths of the edges along the ant's path:

$$L=\sum_{(i,j)\in P} D_{ij}, \tag{12}$$

where P denotes the set of edges that make up the ant's path.

Let us consider the implementation scheme of the four main components of the self-organizing behaviour of ants during the optimization of the traveling salesman route. Sequentially implementing the iterative steps that replicate the procedure of finding the route by each ant, we obtain the functioning scheme of the ant algorithm for solving the fuzzy traveling salesman problem. The transition of an ant from city i to city j at iteration s of the algorithm depends on three components: the tabu list, visibility, and the virtual pheromone trail. The tabu list A_i^k is a list of cities that ant k has already visited before reaching vertex i , and revisiting them is forbidden. This list grows as the ant progresses along the route and is cleared at the beginning of each iteration of the algorithm. Let J_i^k denote the list of cities that ant k, currently at city i, still needs to visit. It is clear that the union of the lists A_i^k and J_i^k gives the set of all cities specified in the traveling salesman problem.

At the first iteration $s=0$, the amount of pheromones on each path is equal to $\tau_{ij}(0)=0$. At the start of the iteration, a set of ants is generated at each vertex of the graph.

During the iteration, each ant acts independently. In the first city i , the tabu list A_i^k for ant k consists of the city where the ant is currently located $A_i^k=\{i\}$. Then, each ant probabilistically selects the next city for movement based on the given travel time to the nearest cities using formula (1). For this, a random number generator will be used.

After selecting the next city j, this city is added to the tabu list $A_j^k=\{i,j\}$. In the next step, the ant selects the following city to move to, and so on, until the last city in the route. If the route cannot be closed, the ant is considered invalid until the next iteration. After the iteration is completed, pheromone evaporation is calculated for each possible edge. Then, for each ant that successfully completes its route after iteration s, the route duration is calculated, and for each edge used in the route, pheromone is added in an amount inversely proportional to the route duration for each ant $\Delta\tau_{ij}(s)=\dfrac{Q}{L_k}$, where L_k is the duration of the successful route of each ant k, and edge $(i,j)\in L_k$ belongs to the route of ant k, Q is a tunable parameter, the value of which is chosen to be on the same order as the length of the optimal route [12].

Thus, with properly chosen parameter values for α and β, we gradually obtain improved results with each iteration.

4. Results of numerical calculations

Let us proceed to the experimental section. As part of the computational experiment, we will calculate and compare the solution to the usual traveling salesman problem obtained by the enumeration method and the solutions to the corresponding fuzzy problem using the proposed approach.

Let us consider the traveling salesman problem on a transport network with a given travel time (Fig. 1, the travel time between individual nodes is given by the values placed on the corresponding edges) [13].

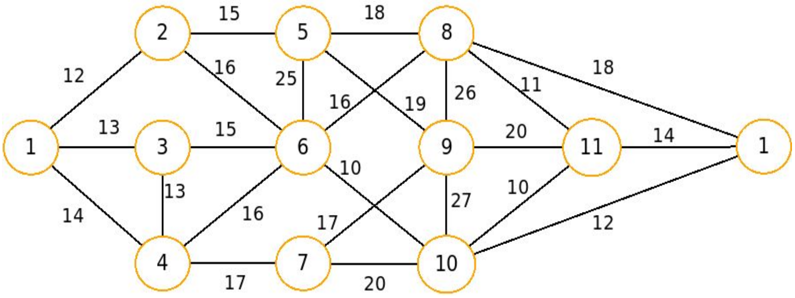

Figure 1: An example of a transport network in the traveling salesman problem

The optimal solution to the traveling salesman problem (1), (2) with a time criterion for the route duration is determined by the following sequence:

$$1 \to 2 \to 6 \to 10 \to 11 \to 8 \to 5 \to 9 \to 7 \to 4 \to 3 \to 1, \quad (13)$$

which is 156 units [13].

In the first stage, the ACO algorithm was adapted to solve the traveling salesman problem with fuzzy travel durations between cities. For this, the problem was fuzzified using trapezoidal fuzzy numbers (Fig.2).

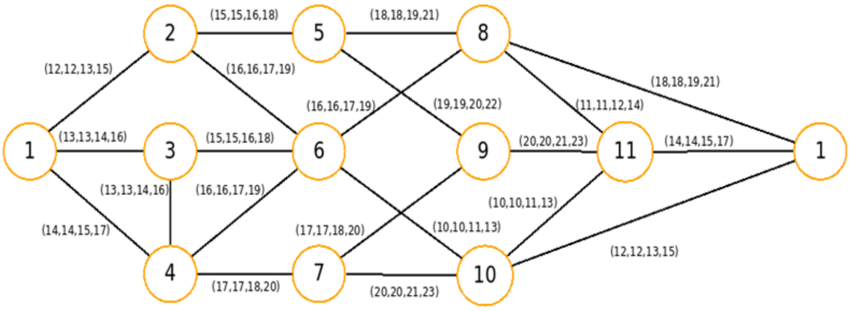

Figure 2: Schematic example of the traveling salesman problem with fuzzy travel durations

The ranking of routes with fuzzy durations was performed using the CoG method for continuous distribution.

For this problem, numerical calculations of the traveling salesman route were conducted using the ACO algorithm and the exhaustive search method. The application of the ASO algorithm to the traveling salesman problem in the specified configuration is characterized by a high convergence speed and ensures the best result with fairly conservative ASO parameters (see Table 1).

Table 1
ACO parameters for solving the TSP shown in Fig. 1.

α	β	Q	V_{vap}	Iteration
0.1	-1	20	0.05	4
0.1	-2	20	0.05	3
0.1	-4	20	0.05	2
0.15	-1	20	0.05	5
0.2	-1	20	0.05	5
0.25	-1	20	0.05	7
0.3	-1	20	0.05	8

In the “iteration” column, the first iteration that contains the best solution with the highest probability is indicated.

The optimal solution to the problem defines the duration of the traveling salesman route as a fuzzy trapezoidal number $(156,156,167,189)$, with the result generally being achieved by the 3rd or 4th iteration, and its defuzzified value based on the CoG method is 167.92 units.

Further experiments with ASO were conducted to assess the quality of the obtained result, taking into account various numbers of cities in the transportation network. During the numerical calculations, the genetic algorithm was used to solve the TSP with random placement of $n = 16,17,18$ cities on a two-dimensional 200x200 plane, where the travel durations between network cities were determined by the average fuzzy time duration, assumed to be proportional to the distance between the corresponding cities. Three series of experiments were conducted, with the number of iterations equal to: four times the number of cities (Table 2), twice the number of cities (Table 3), and the number of cities (Table 4). The number of ants in each case was equal to the number of cities. For each experiment, simulations were conducted to vary the parameters β and α. A comparison of the best result obtained for the specified number of iterations with the best result obtained by exhaustive search is presented as percentages in Tables 2, 3, and 4. As can be seen from the results, when the number of iterations is equal to the number of cities, the algorithm yields results within 5% of the best solution. Increasing the number of iterations improves the results. The variation of the parameter β within the range from -4 to -1 and the parameter α within the range from 0.1 to 0.3 qualitatively impacts the results, bringing them closer to 5% deviation from the best result obtained by exhaustive search.

Table 2
The number of iterations is four times the number of cities

V_{vap}	α	β	Q	Number of cities, quality		
				16	17	18
0.05	0.15	-1	20	103.72%	104.31%	104. 0%
0.05	0.15	-1.5	20	103.56%	103.81%	104.28%
0.05	0.15	-2	20	103.67%	103.73%	104.19%
0.05	0.15	-2.5	20	103.75%	103.93%	103.97%
0.05	0.15	-3	20	103.68%	104.18%	104.27%
0.05	0.15	-3.5	20	103.97%	104.21%	104.30%
0.05	0.15	-4	20	103.88%	104.11%	104.33%
0.05	0.1	-3	20	103.77%	104.04%	104.64%
0.05	0.15	-3	20	103.68%	103.78%	103.87%
0.05	0.2	-3	20	103.66%	103.71%	103.98%
0.05	0.25	-3	20	104.02%	104.10%	104.19%
0.05	0.3	-3	20	104.08%	104.11%	104.32%

Table 3
The number of iterations equals twice the number of cities

V_{vap}	α	β	Q	Number of cities, quality		
				16	17	18
0.05	0.15	-1	20	104.04%	104.50%	104.74%
0.05	0.15	-1.5	20	103.94%	104.48%	104.78%
0.05	0.15	-2	20	103.98%	104.23%	104.81%
0.05	0.15	-2.5	20	103.99%	104.17%	104.26%
0.05	0.15	-3	20	103.97%	103.99%	104.56%
0.05	0.15	-3.5	20	104.10%	104.26%	104.38%
0.05	0.15	-4	20	103.86%	104.36%	104.55%
0.05	0.1	-3	20	103.88%	104.11%	104.58%
0.05	0.15	-3	20	103.87%	103.91%	104.56%
0.05	0.2	-3	20	104.01%	104.40%	104.47%
0.05	0.25	-3	20	104.12%	104.28%	104.83%
0.05	0.3	-3	20	104.34%	104.44%	104.72%

Table 4
The number of iterations equals the number of cities

V_{vap}	α	β	Q	Number of cities, quality		
				16	17	18
0.05	0.15	-1	20	104.75%	104.92%	104.96%
0.05	0.15	-1.5	20	104.61%	105.06%	105.20%
0.05	0.15	-2	20	104.42%	104.45%	104.69%
0.05	0.15	-2.5	20	104.35%	104.69%	105.11%
0.05	0.15	-3	20	104.54%	104.57%	104.75%
0.05	0.15	-3.5	20	104.38%	104.39%	104.71%
0.05	0.15	-4	20	104.11%	105.12%	105.15%
0.05	0.1	-3	20	104.37%	104.50%	105.13%
0.05	0.15	-3	20	104.34%	104.47%	104.75%
0.05	0.2	-3	20	104.31%	104.77%	104.83%
0.05	0.25	-3	20	104.45%	104.70%	104.83%
0.05	0.3	-3	20	104.47%	104.96%	105.05%

For comparison with the results obtained by other methods, computational experiments were conducted using a genetic algorithm (GA) [14] with the use of an improved mutation scheme and population diversity increase related to the PNP (Pick Near point) group [15] and without using PNP. Similar versions of the fuzzy traveling salesman problem with 16, 17, 18 cities were considered.

The table 5 shows the data of the averaged quality indicators of the obtained results of solving the traveling salesman problem based on the genetic algorithm with modeling of a transport network map consisting of $n=16,17,18$ cities (without and with PNP, respectively). The efficiency indicators of calculations using the ACO method were compared with the time indicators of solving the problem using the genetic algorithm (Table 6).

Table 5
The results of numerical calculations based on GA

Number of cities	GA without PNP, deviation from the optimal solution by, %	GA with PNP, deviation from the optimal solution by, %
16	5.11	4.94
17	5.22	5.05
18	5.51	5.25

Table 6
The efficiency indicators of GA and ACO

Number of cities	Genetic Algorithm, sec	Algorithm Colony Optimization, sec
16	1.191	1.212
17	1.452	1.512
18	1.516	1.553

As can be seen from tables 5 and 6, the deviations of the obtained results from the optimal also do not exceed 5.5%, and the time costs for finding a solution do not differ significantly [16]. Thus, it can be concluded that the use of the ACO algorithm allows solving fuzzy traveling salesman problems, while maintaining the accuracy and efficiency of the desired solutions.

5. Conclusion

The results of the studies conducted in this paper demonstrate the application of the *Ant Colony Optimization* (ACO) algorithm to solve the traveling salesman problem with fuzzy travel time along the transport network. Trapezoidal fuzzy numbers are used to formalize the travel time values. A transformation of fuzzy number representation based on a Gaussian-like approach is used, and operations on fuzzy values are defined. The features of the ACO algorithm are considered and an approach for its application is proposed. The effectiveness of using this algorithm for finding locally optimal solutions in traveling salesman problems with a sufficiently large number of cities in the transport network is shown. Estimates of the deviation of the duration of the obtained solutions from the duration of the optimal route are given.

Computational experiments are conducted that show a significant influence of the ant colony algorithm parameters on the accuracy of approximate solutions to the fuzzy traveling salesman problem. The numerical examples provided also demonstrate the influence of the number of iterations on the accuracy of the results obtained.

A conclusion is made about the efficiency of using the ASO algorithm for solving fuzzy traveling salesman problems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Dorigo, G.A. Di Caro, L.M. Gambardella, Ant Algorithms for Discrete Optimization. Artificial Life, 5 (2) (1999), 137-172. doi: 10.1162/106454699568728.

[2] M. Dorigo, Optimization, Learning and Natural Algorithms. Ph.D. Thesis. Politecnico di Milano, Italian. 1992. URL: https://scholar.google.com/citations?view_op=view_citation&hl=en&user= PwYT6EMAAAAJ&citation_for_view=PwYT6EMAAAAJ:XWcFhoZvYIAC.

[3] T. Stützle, H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems, 16 (8) (2000), 889-914. doi: https://doi.org/10.1016/S0167-739X(00)00043-1.

[4] M. Dorigo, Ant Colony Optimization. MIT Press. 2004. doi: https://doi.org/10.7551/mitpress/1290.001.0001.

[5] S. Bouamama, C. Blum, J. Fages, An algorithm based on ant colony optimization for the minimum connected dominating set problem. Applied Soft Computing, 80 (2019), 672-686. doi: https://doi.org/10.1016/j.asoc.2019.04.028.

[6] M. Dorigo, G.A. Di Caro, AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research, 9 (1998), 317-365. doi: 10.1613/jair.530.

[7] J. Kennedy, R.C. Eberhart, Swarm Intelligence. Morgan Kaufmann. 2001. ISBN 978-1-55860-595-4. 2001.

[8] D. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters. Journal of Franklin Institute, 346 (4) (2009), 328-348. doi: 10.1016/j.jfranklin.2008.11. 003.

[9] W.L. Winston, Introduction to Mathematical Programming: Applications and Algorithms. Tomson/Brooks Cole. 2004. URL: https://itslearningakarmazyan.wordpress.com/wp-content/uploads/2015/09/operation-research-aplications-and-algorithms.pdf

[10] L.A. Zadeh, Fuzzy sets. Information and Control, 8 (1965), 338-353.

[11] L. Yang, X. Wang, Z. He, S. Wang, J. Lin, Review of Traveling Salesman Problem Solution Methods, in: Proceedings of 18th International Conference, Bio-Inspired Computing: Theories and Applications, Changsha, China, December 15–17, 2023. Part I. Communications in Computer and Information Science, 2024, pp. 3-16.

[12] D. Dubois, H. Prade, The mean value of a fuzzy number. Fuzzy Sets and Systems, 24(3) (1987), 279–300.

[13] E.V. Ivohin, V.V. Gavrylenko, K.E. Ivohina, On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method. Radio Electronics, Computer Science, Control, 3 (2023), 141-147. doi: https://doi.org/10.15588/1607-3274-2023-3-14.

[14] L.D. Chambers. Practical Handbook of Genetic Algorithms. CRC Press, 2019.

[15] B.A. AlSalibi, V.B. Jelodar, I.A. Venkat. Comparative Study between the NearestNeighbor and Genetic Algorithms: A revisit to the Traveling Salesman Problem. International Journal of Computer Science and Electronics Engineering (IJCSEE), 1 (1) (2013), 34-38.

[16] E.V. Ivohin, K.E. Yushtin. About one approach to solving the fuzzy traveling salesman problem based on the genetic algorithm. Bulletin of Taras Shevchenko National University of Kyiv. Physical and Mathematical Sciences, 78(1), (2024), 140–151.