Neutrosophic Sets and Systems

Volume 52 Article 47

12-1-2022

Neutrosophic Genetic Algorithm for solving the Vehicle Routing Problem with uncertain travel times

Rafael Rojas-Gualdron
Universidad de Investigación y Desarrollo, Barrancabermeja, rrojas7@udi.edu.co

Florentin Smarandache
University of New Mexico, smarand@unm.edu

NSS
Neutrosophic Sets and Systems
{Special Issue: Neutrosophic in Latin America, progress and perspectives}, Vol. 52, 2022
University of New Mexico

Neutrosophic Genetic Algorithm for solving the Vehicle Routing Problem with uncertain travel times

Rafael Rojas-Gualdron [1] and Florentin Smarandache [2]

[1] Department of Industrial Engineering, Universidad de Investigación y Desarrollo, Barrancabermeja, Colombia;
E-mail: rrojas7@udi.edu.co
[2] Mathematics Department, University of New Mexico, 705 Gurley Ave., Gallup, NM 87301, USA
E-mail: smarand@unm.edu

Abstract: The Vehicle Routing Problem (VRP) has been extensively studied by different researchers from all over the world in recent years. Multiple solutions have been proposed for different variations of the problem, such as Capacitive Vehicle Routing Problem (CVRP), Vehicle Routing Problem with Time Windows (VRP-TW), Vehicle Routing Problem with Pickup and Delivery (VRPPD), among others, all of them with deterministic times. In the last years, researchers have been interested in including in their different models the variations that travel times may experience when exposed to all kind of phenomena, mainly vehicle traffic. This article addresses the VRP from this perspective, proposing the design and implementation of a genetic algorithm based on neutrosophic theory for calculating the fitness function of each route, considering the variability and uncertainty present in travel times. A deterministic genetic algorithm is also implemented with the average travel times to compare it with the neutrosophic algorithm using simulation. As conclusion, a deterministic algorithm does not necessarily generate the best solution in the real world, full of uncertainty. Also, the quantification of uncertainty using neutrosophic theory can be used in route planning, opening a broad and interesting field of research for future investigations.

Keywords: Vehicle Routing Problem (VRP); Neutrosophic Theory; Uncertain Travel Times; Stochastic Vehicle Routing Problem (SVRP); Genetic Algorithm

1. Introduction

One of the characteristics that most affect the proper functioning of supply chains today is the variability and uncertainty present in their transportation systems [1], mainly due to three causes: firstly, the high complexity of its large-scale processes, secondly, to the occurrence of complex and random traffic phenomena, and, thirdly, to the susceptibility of being affected by external and unpredictable factors.

It is a common practice in companies to face the challenge of planning routes with many customers and demands using the empirical knowledge of their most experienced workers [2]. However, they face exceptionally tough working conditions due to the increasing number of routes and diversity of demands. For this reason, it is necessary to develop decision support systems that facilitate this task and allow planning efficient routes in short times applicable to the daily operation of companies.

It should also be considered that the flow of traffic on urban roads in developing countries is characterized by being heterogeneous, due to the large number of buses on its main roads [3]. This condition makes it especially difficult to model traffic behavior in its most populated cities and, particularly complex, to estimate vehicle travel times through its streets.

Among the external factors that affect the planning of the routes and cause the calculations to move away from reality, generating an increase in costs, we can consider the unpredictable behavior of some users, the various weather conditions that may occur, some of them extreme, in addition to the errors in the measurement or estimation of the travel times of each one of the routes [4].

Stochastic vehicle routing problems (SVRP) arise whenever some elements of the problem are random [5]. The most common cases found in the literature can be classified into three large groups: vehicle routing problems with stochastic demand, vehicle routing problems with stochastic travel times, or problems in which both conditions are present.

SVRPs differ from their deterministic counterpart in several fundamental aspects, mainly in a different solution concept, which results in much more complex solution methodologies, which are often considered computationally intractable, since only relatively small instances can be solved optimally and the difficulty programmers face when designing and evaluating good heuristics for this kind of problem.

For the scope of this article, the authors focus on vehicle routing problems with stochastic travel times, mainly affected by unpredictable traffic behavior over time. This uncertainty is mainly due to two possible causes: the first, the probability of occurrence of traffic accidents and, the second, the variability of the demand of the transport networks, which generates phenomena such as peak hours or hours of high flow of vehicles and off-peak hours or hours with low vehicle traffic. These factors are the reason why the travel time between any two nodes in the transport network is considered stochastic [6].

A widely used tool to treat this stochasticity is the information available through historical data that is collected and analyzed to be converted into models or probability distributions of the random variables available, for our case study, of the travel times of each one of the arcs of the network [7]. In a pure SVRP, these distributions are available, and the route optimization process is static and is performed only once, without considering the changes that occur during the route.

Among the different solutions found in the literature, we can cite branch and price algorithms [8], [9], simulation-based heuristics [10], simheuristics [11], [12], adaptive local search algorithms, ALNS, [13], scatter search [14], tabu search [15], iterative local search, ILS [16], particle swarm optimization, PSO [17], genetic algorithms [18], non-dominated sorting genetic algorithms, NSGA [19] and memetic algorithms [20], [21].

For the scope of this article, the authors focus on the difficulty of modeling traffic, due to its complexity and randomness, especially on last-mile route planning. Authors based on the open-source data published by Uber Movement of the historical travel times to get the minimum, maximum and average values for travel times between two nodes. Also, a function is proposed to convert this data into a triangular neutrosophic number [22] to make use of the score function to compare two neutrosophic numbers. This score function is used to calculate the fitness function of the proposed genetic algorithm. Finally, the results are compared and a good behavior of the neutrosophic algorithm is observed in stochastic scenarios.

2. The proposed algorithm

The methodology proposed for the development of this research consists of the following phases: the definition of the structure of the data available to carry out the study, the design of a function to convert the available data into neutrosophic triangular numbers, the definition of a fitness function based on a neutrosophic score function to compare two triangular neutrosophic numbers, the definition of the parameters and the design of the genetic algorithm, the implementation of a deterministic genetic algorithm to solve the VRP problem using the mean values of the travel times to compare it with the neutrosophic algorithm and finally the simulation through the generation of scenarios to test the performance of both algorithms.

2.1. Definition of the structure of the data

This research is based on the open-source data provided by Uber Movement [23] on the history of travel times between two nodes, in our case, in the city of Bogotá, Colombia. The data structure that we are going to use includes the minimum, maximum and mean value of the data history of each route for the period between March 1 and March 31, 2020.

2.2. Design of a function to convert the data to neutrosophic triangular number

Once the travel times between a pair of nodes are obtained as a triplet [a, b, c] where a is the minimum value of the interval, b is the mean value and c the maximum value of the interval, we must proceed to convert this triplet into a neutrosophic triangular number $\langle [a, b, c], (T, I, F) \rangle$. For this purpose, the authors propose the use of the following function:

$$TNS\ number = \langle [a, b, c], (T, I, F) \rangle$$

where:

$$\Delta_T = \min\begin{pmatrix} b-a \\ c-b \end{pmatrix}$$

$$\Delta_F = \max\begin{pmatrix} b-a \\ c-b \end{pmatrix}$$

$$T = \begin{bmatrix} 1-\frac{\Delta_T}{b}, & \Delta_T < b \\ 0, & otherwise \end{bmatrix}$$

$$I = \begin{bmatrix} \frac{c-a}{b}, & c-a < b \\ 1, & otherwise \end{bmatrix}$$

$$F = \begin{bmatrix} \frac{\Delta_F}{b}, & \Delta_F < b \\ 1, & otherwise \end{bmatrix}$$

2.3. Definition of the fitness function

To calculate the fitness function, we rely on the score function used to compare two neutrosophic triangular numbers.

Definition 1 [24]**:** (Comparison of any two random TNS numbers). Let $\hat{r}^{NS} = \langle [\hat{r}_T, \hat{r}_I, \hat{r}_P], (T_{\hat{r}}, I_{\hat{r}}, F_{\hat{r}}) \rangle$ be a TNS number, and then the score function is defined as follows:

$$s(r^{NS}) = \frac{1}{12} \cdot [\hat{r}_T + 2 \cdot \hat{r}_I + \hat{r}_P] \cdot [2 + T_{\hat{r}} - I_{\hat{r}} - F_{\hat{r}}]$$

Let $\hat{r}^{NS} = \langle [\hat{r}_T, \hat{r}_I, \hat{r}_P], (T_{\hat{r}}, I_{\hat{r}}, F_{\hat{r}}) \rangle$ and $\hat{s}^{NS} = \langle [\hat{s}_T, \hat{s}_I, \hat{s}_P], (T_{\hat{s}}, I_{\hat{s}}, F_{\hat{s}}) \rangle$ be two arbitrary TNSNs, the ranking of $\hat{r}^{NS}$ and $\hat{s}^{NS}$ by score function is defined as follows:

if $s(\hat{r}^{NS}) \succ s(\hat{s}^{NS})$ then $\hat{r}^{NS} \succ \hat{s}^{NS}$
if $s(\hat{r}^{NS}) \prec s(\hat{s}^{NS})$ then $\hat{r}^{NS} \prec \hat{s}^{NS}$
if $s(\hat{r}^{NS}) \approx s(\hat{s}^{NS})$ then $\hat{r}^{NS} \approx \hat{s}^{NS}$

The value of the fitness function of each individual is equal to the sum of the score function of the neutrosophic triangular number that represents the distance between each pair of nodes of the route.

2.4. Definition of parameters and design of the genetic algorithm

2.4.1. Individual

Each individual is represented by a chromosome of n+2 positions, where n is the number of customers that are visited on the route. For example, for 10 clients, the chromosome would have 12 positions, one for each of the clients represented by 1, 2, 3..., n plus a zero at the beginning and at the end of the chromosome that represent the deposit from which the route starts and where the route finishes as seen on figure 1.

0	3	1	5	6	9	2	4	8	10	7	0

Figure 1. Chromosome for an individual with 10 clients.

2.4.2. Parent selection

The selection is carried out by a selection function, proportional to the fitness function, in which each individual has a probability of being selected as a parent that is proportional to the value of its fitness function. Because for the VRP the best individual is the one with the lowest value of the fitness function, the probability of being selected will be inversely proportional to their fitness.

2.4.3. Crossover function

The crossover function generates two random positions from the first parent to cut a chromosome fragment between the two positions. That fragment is then inserted at a random location within the second parent. Finally, the values that are repeated within the chromosome are eliminated from this resulting chromosome to generate the child. Figure 2 shows an example of the crossover function.

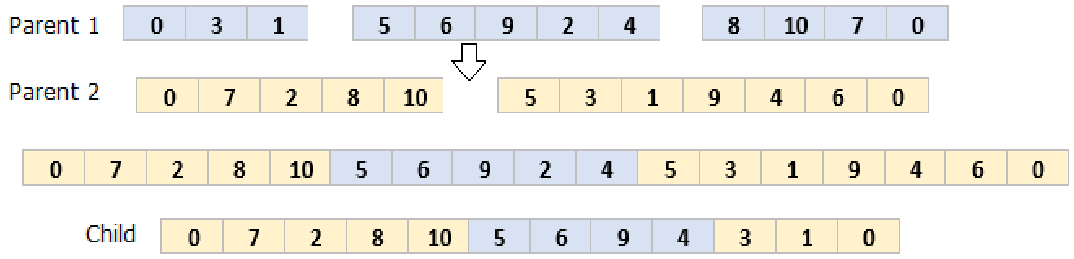

Figure 2. Example of the crossover function.

2.4.4. Mutation function

The mutation probability was set at 5 percent to diversify the individuals and avoid falling into a local optimum. If an individual is chosen for the mutation, two chromosome positions are randomly selected, and their values are swapped. Figure 3 shows an example of a mutation.

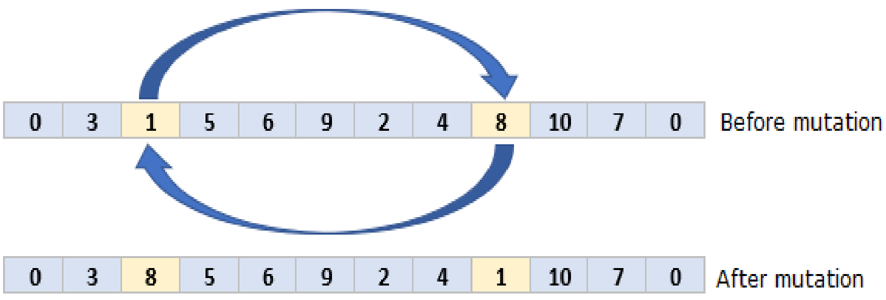

Figure 3. Example of the mutation function.

2.4.5. Pseudocode

Step 1. Generate random population
Step 2. Calculate fitness
Step 3. Update the incumbent
Step 4. For the number of generations:
    Step 4.1. Parent selection
    Step 4.2. Crossover function
    Step 4.3. Mutation function
    Step 4.4. Calculate fitness
    Step 4.5. Update the incumbent

2.5. Design of the deterministic genetic algorithm

To test the results of the proposed neutrosophic algorithm, we are going to implement a deterministic algorithm, which will have the same design as the neutrosophic algorithm with the difference that its fitness function will be equal to the sum of the mean travel times of each pair of nodes.

2.6. Simulation

In order to simulate the behavior of the algorithm in the real world we are going to randomly generate one hundred scenarios based on the data we have. To randomly generate the travel times between any pair of nodes, we need to fit the data we have to a probability distribution. For our case, the probability distribution that is closest to the behavior of the travel times and to the data we have is the triangular distribution, for which we need the values of the maximum, minimum, and mode. The maximum and minimum values are available in the data; however, the mode is not. To calculate the mode, we rely on the following equation applicable to a triangular distribution:

$$Mean = \frac{Maximum + Mode + Minimum}{3}$$

Bearing in mind that we have the value of the mean in the data, we only have to get the mode from the equation to be able to calculate it for the travel times between each pair of nodes.

$$Mode = 3 \cdot Mean - Maximum - Minimum$$

Having the values of maximum, minimum and mode we proceed to randomly generate the travel times between each pair of nodes using the triangular distribution. One hundred scenarios are generated, and the best solution obtained by the neutrosophic algorithm is compared against the best solution obtained by the deterministic algorithm to evaluate the results and draw conclusions.

3. Results

To test the performance of the algorithm, eleven locations will be selected from the Uber Movement database. Ten will be the customers and will be numbered from 1 to 10 and one will be the supplier and will have the number zero. The values in seconds for the maximum value of the range of travel times between each location can be seen in table 1.

Table 1. Maximum value for the range of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0	0	2217	1076	1965	2780	2518	3402	1344	2071	1778	2242
1	1945	0	1732	1505	2586	2926	3635	2283	1339	1037	2000
2	1490	1425	0	719	2095	1719	3692	1371	1231	855	1187
3	2125	1367	1141	0	2323	2758	2376	950	889	984	737
4	2486	2668	2200	1945	0	784	1015	2968	1798	1419	2511
5	1984	1986	1715	1542	773	0	1690	2544	1341	907	2113
6	3086	3558	3491	2517	1273	1583	0	3743	2754	2299	2901
7	1236	2503	1219	1278	3055	2723	3103	0	1762	1823	1058
8	2051	1194	1873	927	1954	2240	2963	1770	0	1009	1552
9	1016	998	809	819	2000	1964	2870	1597	542	0	1362
10	2064	2647	1894	1229	3900	3676	4547	1318	2301	2507	0

The values in seconds for the mean value of the range of travel times between each location can be seen in table 2.

Table 2. Mean value for the range of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0	0	1604	739	1333	1938	1768	2448	922	1491	1228	1550
1	1392	0	1194	1046	1705	1930	2432	1587	942	685	1354
2	896	971	0	481	1379	1147	2351	837	824	546	734
3	1370	989	699	0	1471	1857	1582	578	574	600	405
4	1742	1821	1475	1341	0	342	633	1979	1203	888	1706
5	1459	1393	1199	1089	327	0	988	1726	928	602	1450
6	2234	2576	2376	1865	798	991	0	2487	1889	1533	2164
7	805	1743	711	775	2098	1902	2134	0	1191	1208	585
8	1419	824	1198	600	1215	1435	1856	1171	0	615	976
9	726	688	507	550	1129	1108	1792	1016	364	0	877
10	1375	1905	1194	669	2658	2529	3008	724	1412	1684	0

The values in seconds for the mean value of the range of travel times between each location can be seen in table 3.

Table 3. Minimum value for the range of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0	0	1160	507	904	1350	1241	1761	632	1073	847	1071
1	995	0	823	726	1124	1272	1626	1102	662	452	916
2	538	661	0	321	907	765	1496	510	551	348	453
3	882	715	428	0	931	1250	1052	351	370	365	222
4	1220	1242	988	924	0	149	394	1319	804	555	1158
5	1072	976	837	768	138	0	577	1170	642	399	994
6	1617	1864	1616	1381	500	620	0	1652	1295	1021	1613
7	524	1213	414	469	1490	1328	1467	0	804	800	323
8	981	568	766	387	755	919	1162	774	0	374	613
9	518	473	317	369	637	624	1118	646	244	0	564
10	915	1370	752	364	1811	1739	1989	397	866	1131	0

Using the function defined in section 2.2, we proceed to calculate the degrees of truth, indeterminacy, and falsity of each triplet. The degrees of truth of each of the travel times between each pair of nodes can be seen in table 4.

Table 4. Degree of truth for the values of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0		0,723	0,686	0,678	0,697	0,702	0,719	0,685	0,720	0,690	0,691
1	0,715		0,689	0,694	0,659	0,659	0,669	0,694	0,703	0,660	0,677
2	0,600	0,681		0,667	0,658	0,667	0,636	0,609	0,669	0,637	0,617
3	0,644	0,723	0,612		0,633	0,673	0,665	0,607	0,645	0,608	0,548
4	0,700	0,682	0,670	0,689		0,436	0,622	0,666	0,668	0,625	0,679
5	0,735	0,701	0,698	0,705	0,422		0,584	0,678	0,692	0,663	0,686
6	0,724	0,724	0,680	0,740	0,627	0,626		0,664	0,686	0,666	0,745

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
7	0,651	0,696	0,582	0,605	0,710	0,698	0,687		0,675	0,662	0,552
8	0,691	0,689	0,639	0,645	0,621	0,640	0,626	0,661		0,608	0,628
9	0,713	0,688	0,625	0,671	0,564	0,563	0,624	0,636	0,670		0,643
10	0,665	0,719	0,630	0,544	0,681	0,688	0,661	0,548	0,613	0,672	

The degrees of indeterminacy of each of the travel times between each pair of nodes can be seen in table 5.

Table 5. Degree of indeterminacy for the values of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0		0,659	0,770	0,796	0,738	0,722	0,670	0,772	0,669	0,758	0,755
1	0,682		0,761	0,745	0,857	0,857	0,826	0,744	0,719	0,854	0,801
2	1,000	0,787		0,827	0,861	0,832	0,934	1,000	0,825	0,929	1,000
3	0,907	0,659	1,000		0,946	0,812	0,837	1,000	0,904	1,000	1,000
4	0,727	0,783	0,822	0,761		1,000	0,981	0,833	0,826	0,973	0,793
5	0,625	0,725	0,732	0,711	1,000		1,000	0,796	0,753	0,844	0,772
6	0,658	0,658	0,789	0,609	0,969	0,972		0,841	0,772	0,834	0,595
7	0,884	0,740	1,000	1,000	0,746	0,733	0,767		0,804	0,847	1,000
8	0,754	0,760	0,924	0,900	0,987	0,921	0,970	0,851		1,000	0,962
9	0,686	0,763	0,970	0,818	1,000	1,000	0,978	0,936	0,819		0,910
10	0,836	0,670	0,956	1,000	0,786	0,766	0,850	1,000	1,000	0,817	

The degrees of falsity for each of the travel times between each pair of nodes can be seen in table 6.

Table 6. Degree of falsity for the values of travel times between each pair of nodes.

CUSTOMER	0	1	2	3	4	5	6	7	8	9	10
0		0,382	0,456	0,474	0,434	0,424	0,390	0,458	0,389	0,448	0,446
1	0,397		0,451	0,439	0,517	0,516	0,495	0,439	0,421	0,514	0,477
2	0,663	0,468		0,495	0,519	0,499	0,570	0,638	0,494	0,566	0,617
3	0,551	0,382	0,632		0,579	0,485	0,502	0,644	0,549	0,640	0,820
4	0,427	0,465	0,492	0,450		1,000	0,603	0,500	0,495	0,598	0,472
5	0,360	0,426	0,430	0,416	1,000		0,711	0,474	0,445	0,507	0,457
6	0,381	0,381	0,469	0,350	0,595	0,597		0,505	0,458	0,500	0,341
7	0,535	0,436	0,714	0,649	0,456	0,432	0,454		0,479	0,509	0,809
8	0,445	0,449	0,563	0,545	0,608	0,561	0,596	0,512		0,641	0,590
9	0,399	0,451	0,596	0,489	0,771	0,773	0,602	0,572	0,489		0,553
10	0,501	0,390	0,586	0,837	0,467	0,454	0,512	0,820	0,630	0,489	

Now with the data of the travel times converted to neutrosophic triangular numbers we proceed to execute the genetic algorithm for the proposed example. The best solution obtained by the neutrosophic genetic algorithm with a fitness of 3456.4 can be seen in figure 4.

0	2	1	9	4	6	5	8	3	10	7	0

Figure 4. Best individual for the neutrosophic genetic algorithm.

The next step is to execute the deterministic genetic algorithm using the average values of the travel times between each pair of nodes. The best individual, with a fitness of 8508, for the deterministic genetic algorithm, can be seen in figure 5.

0	2	4	6	5	9	1	8	3	10	7	0

Figure 5. Best individual for the deterministic genetic algorithm.

Afterwards, the travel times between each pair of nodes are simulated for one hundred scenarios. Then the travel time of each of the best routes obtained by both the neutrosophic algorithm and the deterministic algorithm is calculated. The results can be seen in figure 6.

The neutrosophic algorithm obtained a shorter total travel time than the deterministic algorithm in 52 of the 100 simulated scenarios, while the deterministic algorithm obtained a shorter total travel time in 48 of the 100 simulated scenarios. The graph of the percentage of success of each of the algorithms can be seen in figure 7.

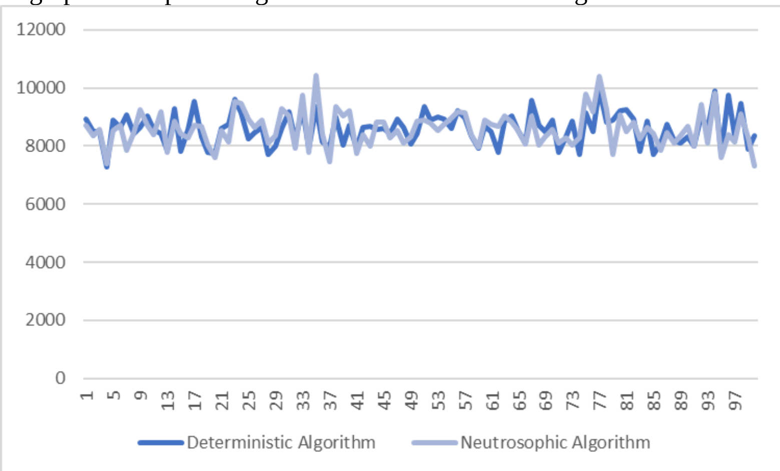

Figure 6. Travel times for both, the neutrosophic algorithm and the deterministic algorithm in the simulation.

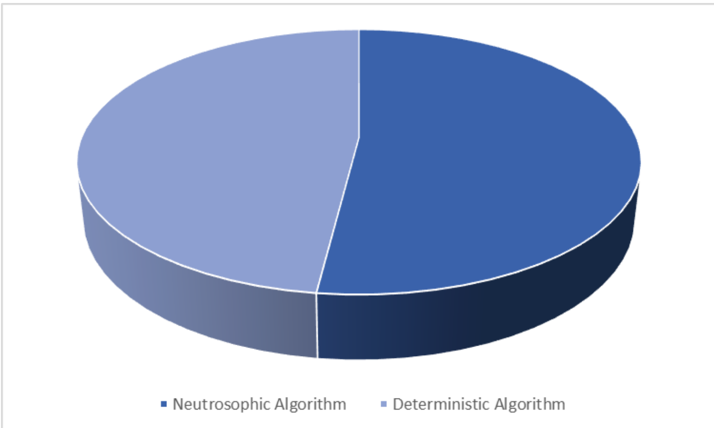

Figure 7. Percentage of successes of each algorithm.

4. Conclusions

Throughout this investigation we were able to verify the importance of including the uncertainty present in the data in the models we use for route planning. Working with mathematical models that only consider deterministic values can lead to generating solutions that in real life will not have the same behavior that was observed when solving the model.

The good performance of the neutrosophic theory was once again verified to work with values subject to different types of uncertainty, since it allows quantifying this uncertainty in order to take it into account in mathematical models and, as in this case, in the algorithms used to solve complex problems.

Even though the simulation was based on fitting the data to a triangular distribution, which greatly reduces the uncertainty present in them, the algorithm presented good results for the problem we were studying. This research is expected to open a new branch in neutrosophic research by combining metaheuristics with neutrosophic theory to solve complex problems that are also subject to high levels of uncertainty.

Funding: "This research received no external funding".

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

References

1. D. Kubek, P. Więcek, and K. Chwastek, "The Impact of Short Term Traffic Forecasting on the Effectiveness of Vehicles Routes Planning in Urban Areas," Transp. Res. Procedia, vol. 18, no. June, pp. 172–179, 2016, doi: 10.1016/j.trpro.2016.12.024.
2. X. Yu and K. Liu, "A spatial decision support system for large scale vehicle routing," 2009 Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2009, vol. 3, pp. 444–449, 2009, doi: 10.1109/ICMTMA.2009.270.
3. Z. Yao, B. Zhao, L. Qin, Y. Jiang, B. Ran, and B. Peng, "An efficient heterogeneous platoon dispersion model for real-time traffic signal control," Phys. A Stat. Mech. its Appl., vol. 539, p. 122982, 2020, doi: 10.1016/j.physa.2019.122982.
4. D. Kubek, "Zastosowanie optymalizacji odpornej w problematyce wyznaczania tras pojazdom: charakterystyka oraz kierunki bada," Logistyka, no. 3, pp. 2654–2663, 2015.
5. M. Gendreau, G. Laporte, and R. Séguin, "Stochastic vehicle routing," Eur. J. Oper. Res., vol. 88, no. 1, pp. 3–12, 1996, doi: 10.1016/0377-2217(95)00050-X.
6. J. Gao, "Model and algorithm of vehicle routing problem with time windows in stochastic traffic network," 2010 Int. Conf. Logist. Syst. Intell. Manag. ICLSIM 2010, vol. 2, pp. 848–851, 2010, doi: 10.1109/ICLSIM.2010.5461065.
7. U. Ritzinger, J. Puchinger, and R. F. Hartl, "A survey on dynamic and stochastic vehicle routing problems," Int. J. Prod. Res., vol. 54, no. 1, pp. 215–231, 2016, doi: 10.1080/00207543.2015.1043403.
8. G. Laporte, F. Louveaux, and H. Mercure, "Vehicle routing problem with stochastic travel times," Transp. Sci., vol. 26, no. 3, pp. 161–170, 1992, doi: 10.1287/trsc.26.3.161.
9. B. Rostami, G. Desaulniers, F. Errico, and A. Lodi, "Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times," Oper. Res., vol. 69, no. 2, pp. 436–455, 2021.
10. Z. Wang and L. Lin, "A Simulation-based algorithm for the capacitated vehicle routing problem with stochastic travel times," J. Appl. Math., vol. 2013, 2013, doi: 10.1155/2013/127156.
11. D. Guimarans, O. Dominguez, J. Panadero, and A. A. Juan, "A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times," Simul. Model. Pract. Theory, vol. 89, pp. 1–14, 2018, doi: 10.1016/j.simpat.2018.09.004.
12. J. Panadero, A. A. Juan, A. Freixes, C. Serrat, M. Grifoll, and M. Dehghanimohamamdabadi, "A SIMHEURISTIC FOR THE UNMANNED AERIAL VEHICLE SURVEILLANCE-ROUTING PROBLEM WITH STOCHASTIC TRAVEL TIMES AND RELIABILITY CONSIDERATIONS," in 2019 WINTER SIMULATION CONFERENCE (WSC), 2019, pp. 1883–1893.
13. D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, "The time-dependent vehicle routing problem with soft time windows and stochastic travel times," Transp. Res. Part C Emerg. Technol., vol. 48, pp. 66–83, 2014, doi: 10.1016/j.trc.2014.08.007.
14. T. Zhang, W. A. Chaovalitwongse, and Y. Zhang, "Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries," Comput. Oper. Res., vol. 39, no. 10, pp. 2277–2290, 2012, doi: 10.1016/j.cor.2011.11.021.
15. X. Li, P. Tian, and S. C. H. Leung, "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm," Int. J. Prod. Econ., vol. 125, no. 1, pp. 137–145, 2010, doi: 10.1016/j.ijpe.2010.01.013.
16. E. Messaoud, "A chance constrained programming model and an improved large neighborhood search algorithm for the electric vehicle routing problem with stochastic travel times," Evol. Intell., no. 0123456789, 2021, doi: 10.1007/s12065-021-00648-0.
17. Z. Shao, S. Gao, and S. Wang, "A Hybrid Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Stochastic Travel Time," in FUZZY INFORMATION AND ENGINEERING, VOL 1, 2009, vol. 54, pp. 566–574.
18. B. Xie and S. An, "Applying genetic algorithm to vehicle routing problem with stochastic travel times," Dyn. Contin. Discret. Impuls. Syst. A-MATHEMATICAL Anal., vol. 13, no. 2, S, pp. 693–697, 2006.

19. J. Oyola, “The capacitated vehicle routing problem with soft time windows and stochastic travel times,” Rev. Fac. Ing. Univ. Pedagog. Y Tecnol. Colomb., vol. 28, no. 50, pp. 19–32, 2019, doi: 10.19053/01211129.v28.n50.2019.8782.
20. A. Gutierrez, L. Dieulle, N. Labadie, and N. Velasco, “A multi population memetic algorithm for the vehicle routing problem with time windows and stochastic travel and service times,” IFAC-PapersOnLine, vol. 49, no. 12, pp. 1204–1209, 2016, doi: 10.1016/j.ifacol.2016.07.673.
21. D. M. Miranda, J. Branke, and S. V. Conceição, “Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time,” Appl. Soft Comput. J., vol. 70, pp. 66–79, 2018, doi: 10.1016/j.asoc.2018.05.026.
22. S. I. A. Aal, M. M. A. Abd Ellatif, and M. M. Hassan, “Two ranking methods of Single Valued Triangular Neutrosophic Numbers to rank and evaluate Information Systems Quality,” Neutrosophic Sets Syst., vol. 19, pp. 133–143, 2018.
23. Uber, “Uber Movement,” 2022. https://movement.uber.com/.
24. R. Kumar, S. A. Edaltpanah, S. Jha, S. Broumi, and A. Dey, “Neutrosophic shortest path problem,” Neutrosophic Sets Syst., vol. 23, pp. 5–15, 2018.

Received: September 15, 2022. **Accepted:** October 31, 2022