Received: 18 October 2022 | Revised: 22 February 2023 | Accepted: 4 May 2023

DOI: 10.1002/net.22159

SPECIAL ISSUE ARTICLE

WILEY

Solving the time capacitated arc routing problem under fuzzy and stochastic travel and service times

Xabier A. Martin[1] | **Javier Panadero[2]** | **David Peidro[3]** | **Elena Perez-Bernabeu[1]** | **Angel A. Juan[1]**

[1]Department of Applied Statistics and OR, Universitat Politècnica de València, Alcoy, Spain
[2]Department of Management, Universitat Politècnica de Catalunya, Barcelona, Spain
[3]Department of Management, Universitat Politècnica de València, Alcoy, Spain

Correspondence
Javier Panadero, Department of Management, Universitat Politècnica de Catalunya, 647 Av. Diagonal, 08028 Barcelona, Spain.
Email: javier.panadero@upc.edu

Funding information
Spanish Ministry of Science, Grant/Award Number: PID2019-111100RB-C21/AEI/10.13039/501100011033; Barcelona Council and the "la Caixa" Foundation under the framework of the Barcelona Science Plan 2020-2023, Grant/Award Number: 21S09355-001; Generalitat Valenciana, Grant/Award Number: PROMETEO/2021/065.

Abstract

Stochastic, as well as fuzzy uncertainty, can be found in most real-world systems. Considering both types of uncertainties simultaneously makes optimization problems incredibly challenging. In this paper we propose a fuzzy simheuristic to solve the Time Capacitated Arc Routing Problem (TCARP) when the nature of the travel time can either be deterministic, stochastic or fuzzy. The main goal is to find a solution (vehicle routes) that minimizes the total time spent in servicing the required arcs. However, due to uncertainty, other characteristics of the solution are also considered. In particular, we illustrate how reliability concepts can enrich the probabilistic information given to decision-makers. In order to solve the aforementioned optimization problem, we extend the concept of simheuristic framework so it can also include fuzzy elements. Hence, both stochastic and fuzzy uncertainty are simultaneously incorporated into the CARP. In order to test our approach, classical CARP instances have been adapted and extended so that customers' demands become either stochastic or fuzzy. The experimental results show the effectiveness of the proposed approach when compared with more traditional ones. In particular, our fuzzy simheuristic is capable of generating new best-known solutions for the stochastic versions of some instances belonging to the *tegl*, *tcarp*, *val*, and *rural* benchmarks.

KEYWORDS

arc routing problem, fuzzy techniques, metaheuristics, optimization, simheuristics, simulation-optimization, uncertainty

1 | INTRODUCTION

Many real-world challenges in different industries can be modeled as optimization problems and then analyzed using simulation or optimization methods. These simulation or mathematical models are intended to include all the key elements and their relationships [37]. The most important purpose of these models is to support decision-making in real-world scenarios, and uncertainty is one of the main factors to be considered when defining the model. According to [64], uncertainty can be represented by probability distributions, fuzzy sets, interval models, or convex models. The causes of uncertainty can range from incomplete information to complexity, subjectivity, or conflicting evidence [72]. For this reason, different ways of dealing with uncertainty are employed. For example, in Liu et al. [39], random variables and simulation models were used to study stochastic models. This paper discusses a stochastic and fuzzy version of the Time Capacitated Arc Routing Problem (TCARP), which is an extended version of the CARP. The CARP is a routing problem where the goal is to visit a set of required arcs or edges in a usually incomplete network. In addition, it is usual to consider some vehicle capacity constraints that limit the volume each

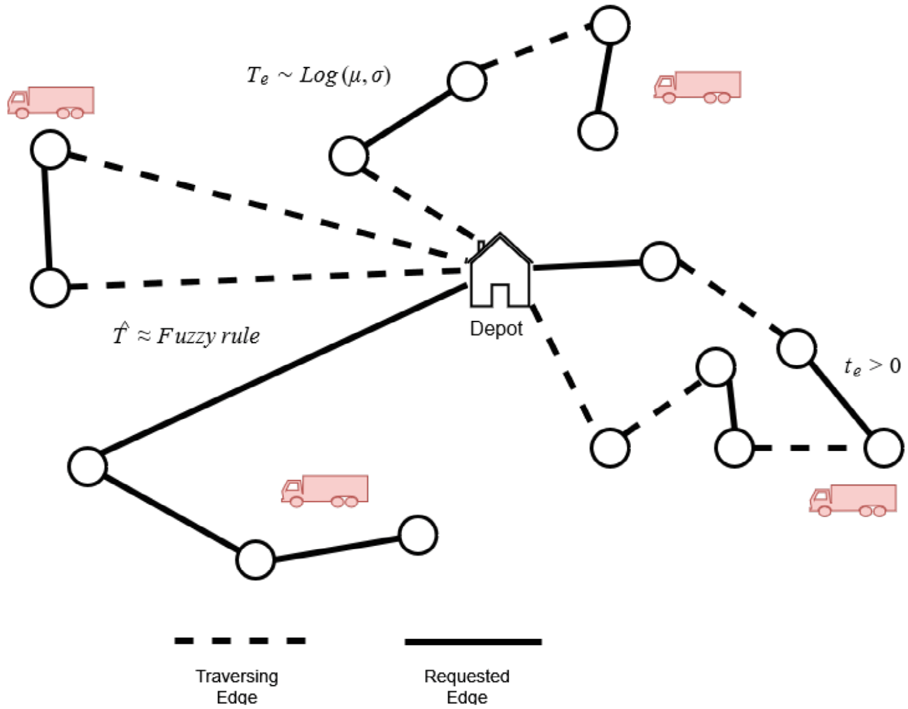

FIGURE 1 The capacitated arc routing problem with stochastic and fuzzy travel times.

vehicle can carry. CARPs are NP-hard, so it is difficult to find optimal solutions in short computing times as the network size increases. Therefore, research has focused on identifying good lower bounds and near-optimal solutions [65]. A further specialization of the CARP includes the case where the vehicle capacity constraint is a time limitation rather than a volume constraint, which is the case of the TCARP.

As explained in Chica et al. [11], a simheuristic approach allows for the integration of simulation and metaheuristics in order to optimize systems under conditions of stochastic uncertainty. Many papers have recently employed simheuristics to solve stochastic optimization problems in different areas, as in finance [49] or transportation [32]. Rabe et al. [57] describe a simheuristic framework for dealing with systems involving mild stochastic uncertainty, which is a special case of uncertainty [69]. Yet, another type of uncertainty is the fuzzy one [19]. This paper considers both types of uncertainty, stochastic and fuzzy, so the simheuristic framework is extended to include fuzzy uncertainty. We propose a fuzzy simheuristic algorithm to solve the TCARP, where the travel times can be deterministic, stochastic, or fuzzy in nature. Therefore, the objective is to minimize the total time spent in servicing the required arcs. Figure 1 shows a schematic representation of a TCARP problem with different uncertainty scenarios.

The aim of this work is to solve the TCARP problem considering different uncertainty scenarios for each of the edges, either stochastic or fuzzy, within reasonably short computational times. Since no instances of the TCARP considering time-based random and fuzzy capacities have been found in the literature, we have extended the existing benchmarks already employed in Keenan [29]. The structure of the rest of the paper is as follows: first, Section 2 reviews the general concepts and works related to the methodology shown in this paper. Some of the related work available in the literature is presented in Section 3. Our approach to solving the TCARP is described in Section 4. A series of computational experiments are described in Section 5. Section 6 discusses the results, and deterministic solutions under uncertainty are analyzed and compared to solutions found using the proposed fuzzy simheuristic approach. Finally, Section 7 highlights the main outcomes of this work and points out some future research.

2 | BACKGROUND CONCEPTS

This section reviews the background concepts and works related to the methodology presented in this paper, starting with the concepts of fuzzy systems and following with simheuristics.

2.1 | Overview of fuzzy concepts

According to Pishvaee et al. [56], uncertainty in data can be classified into two categories: (i) uncertainty due to randomness; and (ii) uncertainty caused by a lack of details (epistemic uncertainty). The first category is randomness, which is caused by the random nature of parameters. If sufficiently historical data on the distribution function of the parameters is accessible, stochastic methods are mostly accurate. Nevertheless, in the absence of previous data, the distribution function of the unknown parameter cannot be properly calculated. The second category of ambiguity is epistemic, which results from a lack of knowledge or information about uncertain variables.

Fuzzy set theory [67] is one of the widely used and acceptable tools to deal with non-probabilistic uncertainty [35]. While fuzzy sets and systems have many applications, their use in rule-based systems has demonstrated their significance as a robust design approach [44]. A fuzzy inference system (also known by many other names, such as fuzzy logic system, rule-based fuzzy logic system, fuzzy expert system, or simply fuzzy system), is a computational system that uses fuzzy logic to make decisions or predictions based on uncertain input information. It is a mathematical model that mimics the reasoning process of a human expert in a particular domain. A fuzzy inference system consists of three main components:

- Fuzzification: The process of converting crisp input values into fuzzy sets. This is done by assigning membership degrees to each input value according to a predefined set of membership functions.
- Inference engine: The process of applying a set of fuzzy rules to the fuzzy input sets to infer a fuzzy output. The inference engine uses a set of if-then rules to make decisions based on the input information.
- Defuzzification: The process of converting the fuzzy output values into crisp values. This is typically done by calculating the center of gravity of the fuzzy output set, which corresponds to the crisp value most likely to represent the output.

The conventional fuzzy sets, also called type-1 fuzzy sets (T1FS), generalize the traditional set concept by allowing the degree of membership to be any value between 0 and 1 [67]. A T1FS, denoted A, in a universe of discourse X can be defined as:

$$A = \{(x, \mu_A(x)) \,|\, \forall x \in X\}, \tag{1}$$

in which $0 \le \mu_A(x) \le 1$. The value of $\mu_A(x)$ is called the degree of membership, or membership grade, of x in A. If $\mu_A(x) = 1$ or $\mu_A(x) = 0\ \forall x \in X$, then the fuzzy set A reduces to a crisp set. In the real world, this fuzzy approach corresponds to many situations where it is difficult to decide, in an unambiguous manner, if something belongs to a specific class [8]. The Inference engine is the component that applies the fuzzy rules to the fuzzified input values to infer the fuzzy output values. A fuzzy rule has the structure "IF x is A THEN y is B," in which "x is A" is called the rule's *antecedent* and "y is B" is called the rule's *consequent*. A and B are linguistics values defined by fuzzy sets on universes of discourse X and Y, respectively. The Inference engine matches the fuzzified input values with the antecedent (premise) parts of the fuzzy rules. Once the antecedent parts of the fuzzy rules have been matched against the fuzzified input values, the inference engine uses a method called "fuzzy implication" to combine the premise and consequent (conclusion) parts of the rules. It typically involves taking the minimum (or "and" operation) of the premise parts and the maximum (or "or" operation) of the conclusion parts. The result is a fuzzy set representing the inferred value of the output variable, which is passed to the defuzzification component to convert it into a crisp output value. T1FS systems have been successfully applied in various fields, such as automatic control, data classification, decision analysis, expert systems, forecasting, robotics, pattern recognition, etc. [25].

According to [44] and [45], there are different sources of uncertainties in a T1FS system:

- Uncertainty about the meanings of the words used in the rules (i.e., words mean different things to different people).
- Uncertainty about the consequent used in a rule (especially when knowledge is extracted from a group of experts who do not all agree).
- Uncertainty about the measurements that activate the fuzzy system (maybe noisy).
- Uncertainty about the data used to tune the parameters of a fuzzy system (may also be noisy).

Such uncertainties lead to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions [28]. T1FS systems, like the ones mentioned above, cannot directly handle such uncertainties because their membership functions are totally crisp [9, 45].

The concept of a type-2 fuzzy set (T2FS) was introduced by Zadeh [68] as an extension of the concept of T1FS. Such T2FS are fuzzy sets whose membership grades themselves are T1FS. Basically, a T2FS is a set in which we also have uncertainty about the membership function. They are very useful when it is difficult to determine an exact membership function for a fuzzy set. Hence, they are useful for incorporating uncertainties [28]. A T2FS provides additional degrees of freedom for modeling uncertainties compared to T1FS, and it can achieve a higher degree of approximation in modeling real-world problems [8]. Following Mendel and John[45], a T2FS, $\tilde{A}$, can be defined as:

$$\tilde{A} = \left\{((x, u),\ \mu_{\tilde{A}}(x, u)) \,|\, \forall x \in X,\ \forall u \in J_x,\ J_x \subseteq [0, 1]\right\}, \tag{2}$$

where $0 \le \mu_{\tilde{A}}(x, u) \le 1$ is the type-2 membership function, J_x is the primary membership of $x \in X$, which is the domain of the secondary membership function $\mu_{\tilde{A}}(x)$. A T2FS is characterized by a bounded region called the footprint of uncertainty (FOU). It is the union of all primary membership functions, that is:

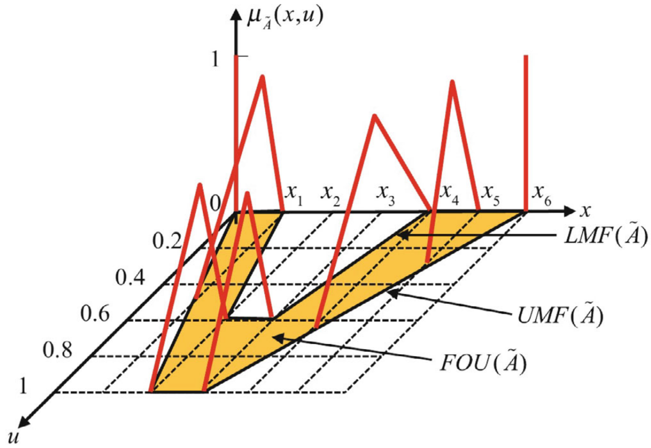

FIGURE 2 Example of a type-2 fuzzy set.

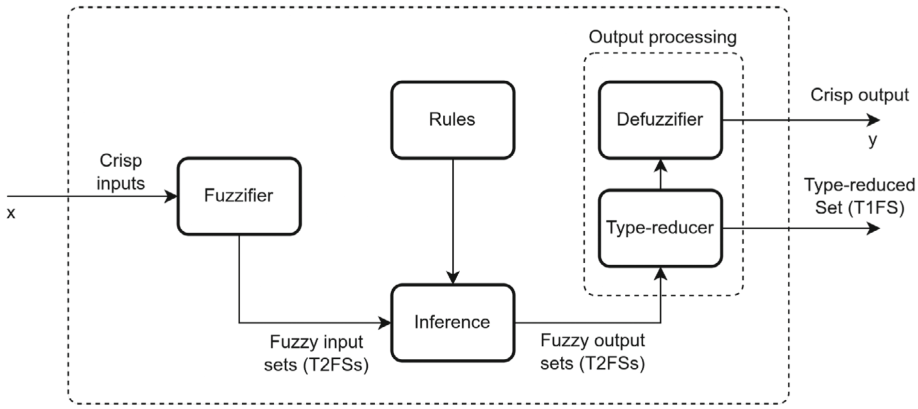

FIGURE 3 Type-2 fuzzy inference system.

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x. \tag{3}$$

The FOU is bounded by an upper membership function $\overline{\mu}_{\tilde{A}}(x)$ (UMF) and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$ (LMF), which are both T1FS membership functions [46]. The FOU is very useful since it not only focuses our attention on the uncertainties associated with a specific type-2 membership function, but it also provides a very convenient verbal description of the entire domain of support for all the secondary grades of a type-2 membership function [45]. An example of a T2FS, which includes the concepts of FOU, UMF, and LMF, is depicted in Figure 2.

A T2FS system also consists of fuzzy *if-then* rules defined as: "IF x is $\tilde{A}$ THEN y is $\tilde{B}$." We can say that a T2FS system is a generalization of a conventional T1FS system in the sense that uncertainty is not only limited to the linguistic variables but also is present in the definition of the membership functions [8, 58]. In the T2FS system (see Figure 3), as in the T1FS system, crisp inputs are first fuzzified into fuzzy input sets that then activate the inference block. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to compute unions and intersections of type-2 fuzzy sets, as well as compositions of type-2 relations. The main difference between T2FS and T1FS systems is the extra step called type-reduction in the output processing. The type-reducer transforms a T2FS into a T1FS, and then the defuzzifier map that set into a number (as in the T1FS systems). Multiple approaches and algorithms are available in the literature for converting a T2FS output into a crisp number (including direct defuzzification methods without type-reduction). For a detailed explanation of the type-reduction methods and algorithms available, the reader is referred to [44] and [8]. T2FS systems have been applied in many engineering areas, showing their ability to perform better than T1FS systems when facing dynamic uncertainties [17, 23, 38, 41]. Uncertainty built into the definition of a T2FS system improves system resilience against noise and handling of unknown data [58]. Due to the higher computational complexity of a T2FS system compared to a T1FS one [47, 66], the most widely used T2FS to represent system uncertainties is interval type-2 fuzzy set (IT2FS). An IT2FS is a special case of T2FS where all of its secondary grades in Equation (2) equal 1. An Interval IT2FS system is a compromise between the uncertainty modeling capabilities of a general T2FS and the computational inexpensiveness of a T1FS [35]. In this paper, we focus on IT2FS systems.

2.2 | Overview of simheuristics

The term simheuristics comes from a mixture of the words simulation and metaheuristics. Simulation is a technique that makes it possible to reproduce the performance of a given system over a period of time, from an initial situation, according to the input data, and gather information about the system as output data [37]. Faulin et al. [16] state that this approach is well known for modeling complex problems with stochastic variables and, consequently, analyzing its behavior, performance, and reliability. Since simulation only provides information about the performance of the system, when solving problems under stochastic uncertainty, simulation-optimization approaches have been used. They involve optimizing a stochastic objective function subject to stochastic constraints [4]. For more detail on simulation-optimization, please refer to reviews, such as those written by [3], [40], and [60], among others.

Simheuristics is a combination of simulation—which does not lead to any type of optimization—and heuristic-based optimization—which uses the previous simulation stage to guide the algorithm during the search for near-optimal solutions. These simheuristic algorithms are especially useful when dealing with large-scale and NP-hard optimization problems under stochastic uncertainty. The metaheuristic algorithm provides a set of different solutions tested by the simulation. Then, each event is modeled by the best-fit probability distribution, and the results produced by the simulation provide feedback to the metaheuristic process. This feedback might contain useful information to improve the search for the best stochastic solutions. After meeting a stop condition, a limited set of "elite" solutions are sent to a more intensive simulation process in order to obtain accurate estimates of their behavior under a stochastic scenario. The outputs of the simulation process include not only central statistics but also dispersion measures in order to evaluate the risk of the candidate solutions [11]. Glover et al. [18] explained how to combine simulation and optimization and proved that this combination was able to find optimal or near-optimal solutions in a few minutes to problems that previously took days or even months to solve. Fuzzy simheuristics extend the simheuristic concept by allowing us to consider fuzzy uncertainty as well. In general, fuzzy simheuristics are designed to cope with stochastic optimization problems with the following structure:

$$\text{optimize } f(s) = \Phi(s, X, Y), \tag{4}$$

subject to:

$$Pr\left(g_i(s, X, Y) \le l_i(X, Y)\right) \le t_i \qquad \forall i \in I, \tag{5}$$

$$Pr\left(g_j(s, X, Y) > l_j(X, Y)\right) \le t_j \qquad \forall j \in J, \tag{6}$$

$$g_k(s) \le t_k \qquad \forall k \in K, \tag{7}$$

$$s \in S, \tag{8}$$

where s is an element of the solution space S, X is a vector of random variables, and Y is a vector of fuzzy variables. The objective function in Equation (4), $\Phi(s, X, Y)$, is a stochastic-fuzzy measure to be optimized (minimized or maximized). Equations (5) and (6) present probabilistic constraints. For example, the probability that the time delay in solution s, $g_j(s, X, Y)$, exceeds a certain limit, $l_j(X, Y)$, is within a user-defined threshold. Equation (7) is a typical deterministic constraint.

3 | RELATED WORK

This section offers a brief review on the applications of fuzzy logic and simheuristics to solve optimization problems with uncertainties. It also reviews recent work on the CARP.

3.1 | Fuzzy and simheuristic approaches in transportation problems

Although there is previous research dealing with different vehicle routing problems under fuzzy uncertainty (mainly considering fuzzy demands, travel times, and time-windows constraints), fuzzy set-based techniques have rarely been used in the CARP and their variants. Wang and Wen [63] investigated a Chinese Postman Problem (CPP) with time window constraints in a fuzzy environment while considering two application cases. The first case considered the arrival time as uncertain and a fuzzy relation was used to describe the degree of possibility. The second case considered the upper and lower bounds of time constraints as fuzzy numbers. Although the authors do not present a solution method, they suggested the use of an ant-colony optimization algorithm. A multi-objective CPP was presented by Majumder et al. [42] using the uncertainty theory as a framework. The authors proposed a model with two goals: maximizing the total expected net profit produced by the post person from a tour and minimizing the total expected travel time of the tour. They considered uncertain variables like allowance, travel expense, and travel time. One of the methods used to solve the multi-objective model was the fuzzy programming method proposed by [71]. The fuzzy credibility theory is the basis for the study by Babaee Tirkolaee et al. [5]. Assuming fuzzy demands on the necessary links, these authors addressed a multitrip CARP in solid waste management and coped with the unknown amount of waste produced in urban sites. The authors proposed an ant-colony optimization metaheuristic and evaluated the applicability of the recommended methodology using a real-world case study from Iran. All of the aforementioned works used T1FS systems in their approaches.

As mentioned before, T2FS systems have been used in many engineering problems. However, applications of the T2FS systems to routing and transportation problems are still scarce. Tadić et al. [59] solved the selection of recycling center locations using a two-objective genetic algorithm. The optimization criteria are presented as total distance, and the overall suitability index is represented by interval triangular type-2 fuzzy numbers. Zandieh and Ghannadpour [70] minimized the transportation risks and travel costs of a hazardous materials (HazMat) routing problem, which was solved using an evolutionary algorithm. An interval type-2 fuzzy logic controller is used to estimate risk, including population density, vehicle load, link length, and time of day as inputs for the system. Cengiz Toklu [10] developed a new method integrating the interval type-2 fuzzy TOPSIS method with a savings criterion. They presented a numerical example related to a distribution problem for humanitarian aid. The interval type-2 fuzzy TOPSIS provides the benefit of identifying safer routes by considering the various dangers that might occur during a disaster. Men et al. [43] presented a HazMat Capacitated Vehicle Routing Problem (CVRP) under uncertainty. To identify a set of routes with the lowest transportation risk, they proposed a model with trapezoidal interval type-2 fuzzy variables in the objective function. Population density near the accident site is considered a fuzzy variable due to the mobility of the population. They employed chance-constrained programming and a simulated annealing algorithm to solve the equivalent deterministic model.

Simheuristics have been effectively used in several application fields, including production planning and scheduling [1, 14, 48, 62], portfolio selection [53], supplier selection [27], telecommunication networks [2], facility location [52], defense [36] and transportation systems [24, 54, 61]. In order to solve the CARP with stochastic demand, Gonzalez-Martin et al. [21] presented a simheuristic approach combining Monte Carlo simulation with the RandSHARP metaheuristic. Keenan et al. [32] proposed a strategic oscillation simheuristic algorithm for the TCARP, considering stochastic demands and service times. Dealing simultaneously with stochastic and fuzzy uncertainty we can find the works by Oliva et al. [50] and Tordecilla et al. [61]. The former proposed a fuzzy simheuristic approach for the stochastic and fuzzy version of the Team Orienteering Problem (TOP). They considered a more realistic scenario in which the reward for any customer can be given by a deterministic value, a random variable following a probability distribution, or a fuzzy function represented using a T1FS system. In the latter, the authors solve a CVRP and a TOP involving stochastic and fuzzy variables. Again, their simheuristic is based on a T1FS system, a multistart metaheuristic, and a Monte Carlo simulation. To the best of our knowledge, there is no previous research in the literature addressing the CARP with both stochastic and fuzzy uncertainty, with the latter being modeled using a T2FS system as proposed in this paper.

3.2 | The stochastic capacitated arc routing problem

The CPP is the simplest form of the CARP, in which the objective is to visit a set of connecting edges or arcs. Some additional considerations, such as time or volume constraints, can be taken into account [12]. The graph of the CARP can be defined as $G = (V, E)$, with V being the set of vertices including the depot d, and E being the set of edges or arcs in a graph. The Rural Post Problem (RPP) is a different version of the CPP in which some arcs or edges require to be visited and some others do not. In this case, the objective is to find a minimum cost tour visiting all the required arcs and edges [51]. The CARP was initially introduced by Golden and Wong [20]. This type of problem is NP-hard, so it is not easy to find optimal solutions as the size of the underlying network grows [65]. A list of the best exact algorithms proposed for CARP can be found in Pecin and Uchoa [55], who propose a new branch-and-cut-and-price algorithm capable of solving small- and medium-sized CARP benchmark sets. The deterministic version of the problem occasionally misses out on critical real-world factors that arise in transportation and distribution-logistics issues. This occurs when input data or graph properties cannot be adequately modeled in a static or deterministic environment. The CARP can then be formulated within the frameworks of robust optimization and stochastic programming De Maio et al. [15].

If a time constraint is added, then the TCARP will consider the time constraint instead of the volume one. In this case, each edge or arc has a nonnegative travel time representing the time it takes to traverse it. The demand for an edge or arc is based on the time that indicates the service time. This service time also includes the travel times. Each of the available vehicles has a positive time capacity. When an edge traversed by a route is not served, it is referred to as the "deadhead" of that route. The main goal of the TCARP is finding a set of routes that minimizes the total time used for servicing all the required edges given that each edge is only serviced once. The total time includes the time associated with traversing deadhead edges as well as the time servicing the required edges. Heuristics have been used for solving rural postal delivery problems [31]. Lower bounds based on graph theory for the TCARP were tested in Keenan [30]. A more general CARP with Deadheading Demand (CARPDD) was also studied by Bartolini et al. [7], while Kirlik and Sipahioglu [34] developed lower bounds and an exact algorithm based on cut-and-column generation and branch-and-price. TCARP can also be seen as a particular case of CARPDD in which the arc demands and the objective function have the same time units. For a comprehensive review of the literature De Maio et al. [15] summarize the methodologies used in each particular case to model ARPs and the methods used to solve the problem either exactly or using heuristics.

4 | EXTENDING THE SIMHEURISTIC FRAMEWORK WITH FUZZY TECHNIQUES

Algorithm 1 depicts the main characteristics of our fuzzy-simheuristic methodology (Figure 4). First, a feasible initial solution *initSol* is generated using the constructive savings heuristic proposed in De Armas et al. [13] (TSHARP heuristic). In order to generate a solution, the variables of the problem are previously replaced by their expected values. Once this initial solution is generated, it is copied into *baseSol* and *bestSol*. Next, during the second stage, an iterated local search (ILS) metaheuristic is applied to improve the initial solution by iteratively exploring the search space. The ILS is a relatively simple metaheuristic that

Algorithm 1. Fuzzy simheuristic algorithm

```
t_elapsed ← 0
eliteSols ← Ø
initSol ← TSHARPHeuristic(inputs)
initSol ← localSearch(initSol)
simulation(initSol, q_short)
baseSol ← initSol
bestSol ← baseSol
insert(eliteSols, baseSol)
while (t_elapsed < t_max) do % Iterated local search
    newSol ← perturbation(baseSol)
    newSol ← localSearch(newSol)
    Δ ← cost(newSol) − cost(baseSol) % Acceptance criterion
    if (Δ < 0) then % Improvement case
        simulation(sol, q_short)
        Δ_exp ← expCost(newSol) − expCost(baseSol)
        if (Δ_exp < 0) then
            credit ← (−Δ_exp)
            baseSol ← newSol
            if (expCost(newSol) < expCost(bestSol)) then
                bestSol ← newSol
                insert(eliteSols, newSol)
            end if
        end if
    end if
    if (0 < Δ ≤ credit) then % Deterioration case
        credit ← 0
        baseSol ← newSol
    end if
    t_elapsed ← update(t_elapsed)
end while
for (sol ∈ eliteSols) do
    simulation(sol, q_long)
end for
eliteSols ← sort(eliteSols)
topSols ← selectTop(eliteSols)
return topSols
```

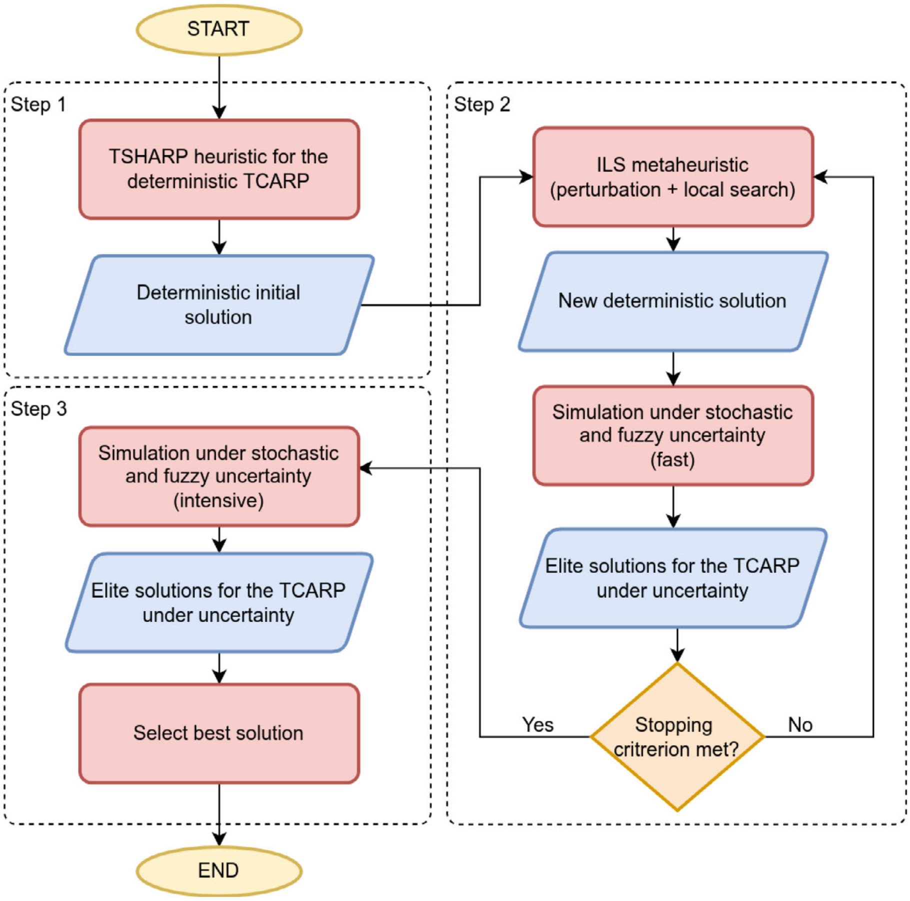

FIGURE 4 Flowchart of the fuzzy-simheuristic methodology.

sequentially applies a diversification method (perturbation) and a local search around the perturbed solution until a stopping criterion is met. Hence, the process starts by perturbing the base solution (*baseSol*), generating a new solution (*newSol*). In our case, the perturbation is based on a destruction and reconstruction procedure, which depends on the degree of destruction to be applied, k. To reconstruct the partially destroyed solution, we use a biased-randomized version of the TSHARP savings heuristic. At each step of the solution-reconstruction process, the next element in the savings list is selected following a decreasing geometric distribution, where elements with higher cost savings have a higher probability of being chosen. As described in Grasas et al. [22], this type of distribution provides a natural bias for the candidate elements in the list. In addition, it offers two advantages over using empirical distributions: (i) they can be sampled using well-known analytical expressions and (ii) they contain at most one simple parameter. The geometric distribution is controlled by a parameter $\beta \in (0, 1)$, which facilitates its setting in most practical applications. As values of β get closer to 0, the more uniform-randomized the selection process will be. On the contrary, as these values of β get closer to 1, the more greedy the selection will be. Thereby, the deterministic TSHARP heuristic is extended into a probabilistic one by assigning different probabilities to the savings list elements. In this way, we prevent the same solution from being obtained at every iteration, as different routes will be reconstructed at each destruction and reconstruction step. After generating a new solution, the algorithm starts a local search procedure around *newSol*. This procedure consists of applying a well-known 2-opt local search to every possible combination of two edges of a route. This is applied to each route of the solution until no further improvement can be made this way. Thus, the routes become 2-optimal after the complete 2-opt local search. Subsequently, we use a hash map data structure, which acts as a fast-access memory, to store the best-known way (shortest time route) of traversing the set of edges in each route of the solution. The information in this hash map is updated each time a set of edges is traversed so that newly generated routes are quickly improved as more information is stored in the data structure. So far, the new solution responds to the deterministic version of the TCARP. In order to deal with the uncertainty of the problem and estimate its cost under uncertainty, an exploratory fast simulation phase is carried out next. Specifically, a relatively short number of simulation runs are carried out to assess the effect of stochastic and fuzzy uncertainty on the objective value of the solution. In addition, constraints are also evaluated under uncertain conditions. Notice that, in each simulation run, a different value is assigned to random variables or fuzzy elements according to a probability distribution or fuzzy rules. Besides assessing the solutions under uncertainty scenarios, the examination of solutions also guides the metaheuristic in generating new solutions. For example, a "good" solution could be the starting point for generating new solutions in the subsequent iterations of the metaheuristic. Once the uncertainty cost of the solution has been computed, some evaluations are performed to update the *baseSol* and *bestSol* solutions if needed. Specifically, if the cost of the new solution under uncertainty is able to improve the cost of the *baseSol*, the latter is updated. In the same way, if the cost of the *newSol* improves the cost of the *bestSol* found so far, the latter is updated and added to a pool of "elite" solutions. With the goal of further diversifying the search, the algorithm might occasionally accept nonimproving solutions following an acceptance criterion based on a credit value [26]. Once the stopping condition is met, an intensive simulation phase is launched to better assess the elite solutions before reporting the final results. In this phase, a large number of runs are used to examine the solutions' quality in comparison with the first examination. Identifying the best solution might take into account measures of interest other than the expected value, for example, the variance or the reliability of each solution. Finally, descriptive statistics are obtained for each solution, providing detailed information.

5 | DETAILS ON THE SOLVING APPROACH AND COMPUTATIONAL EXPERIMENTS

The proposed fuzzy-simheuristic algorithm has been implemented using Python 3.10, and tested on a workstation with a multi-core processor Intel Xeon E5-2630 v4 and 32 GB of RAM. For the computational experiments, we have set the parameters of the algorithm using the one-factor-at-a-time (OFAT) method [6], which studies each parameter of the algorithm individually. The OFAT method has been chosen over more sophisticated design of experiments (DOE) methods since the number of parameters to tune is reasonably small, and there are no interactions between the parameters of the algorithm. Thus, we test different values for each of the parameters using uniformly spaced samples and choose the values that obtain the best results over a small set of instances. After the tuning process of the parameters, we set a computational time of 60 seconds for the small and medium instances and 600 s for the large ones. The value of the β parameter for the geometric distribution was set to be randomly selected in the interval $(0.10, 0.25)$. The exploratory and refinement stages of the examination phase were set to 100 and 1000 simulation runs, respectively. Finally, each instance has been executed 10 times using our fuzzy-simheuristic algorithm with different seeds for the random number generator, and the best and average results are reported. To the best of our knowledge, there are no TCARP instances using random and fuzzy times that we can use as a benchmark. Accordingly, a set of previously published deterministic TCARP and CARPDD benchmarks have been extended to assess the performance and the quality of the proposed algorithm. In particular, we have employed the *tcarp*, *tegl*, *val*, and *rural* data sets described in Keenan et al. [32].

Table 1 summarizes the main characteristics of the used data sets: the number of instances, the minimum number of nodes, the maximum number of nodes, the minimum number of edges, the maximum number of edges, and time-based capacities. Notice that not all datasets use the same value for the time capacity Q. Hence, the table shows the different values of Q employed in each dataset.

To extend the instances mentioned above, we have assumed that the service time, Q_e, is uncertain and, therefore, it is either stochastic or fuzzy depending on the considered scenario. The service time Q_e is comprised of two components: the deterministic service time q_e and the service delay D_e, that is, $Q_e = q_e + D_e$. The deterministic service time represents the travel time required under perfect conditions, while a random or fuzzy time is considered for the service delay. The remainder of this section describes the process used to transform the deterministic instances of each data set into stochastic or fuzzy ones. In the deterministic scenario, no uncertainty is considered. Thus, the service time Q_e is known in advance. These service times are provided by the dataset instances themselves as there are no service delays.

Considering the stochastic scenario, the Log-Normal distribution is a natural choice for describing nonnegative random variables, such as service times [33]. In our experiments, the Log-Normal probability distribution is used to model the service delay D_e of service time Q_e. The Log-Normal distribution has two parameters: the location parameter, μ_e, and the scale parameter, σ_e, which can be derived from the expected value $E[D_e]$ and the variance $\text{Var}[D_e]$. We have set the expected value $E[D_e] = 0.1 \cdot d_e$, where d_e is the service delay, and the variance $\text{Var}[D_e] = c \cdot E[D_e]$. The parameter c is a design parameter that allows us to set

TABLE 1 Summary of the datasets.

Dataset	Instances	Minimum nodes	Maximum nodes	Minimum edges	Maximum edges	Time capacities (Q)
tcarp	10	17	44	24	50	40, 50, 60
tegl	8	77	140	98	190	240, 360
val	5	30	50	50	97	144–918
rural	5	221	599	245	872	360, 400, 420

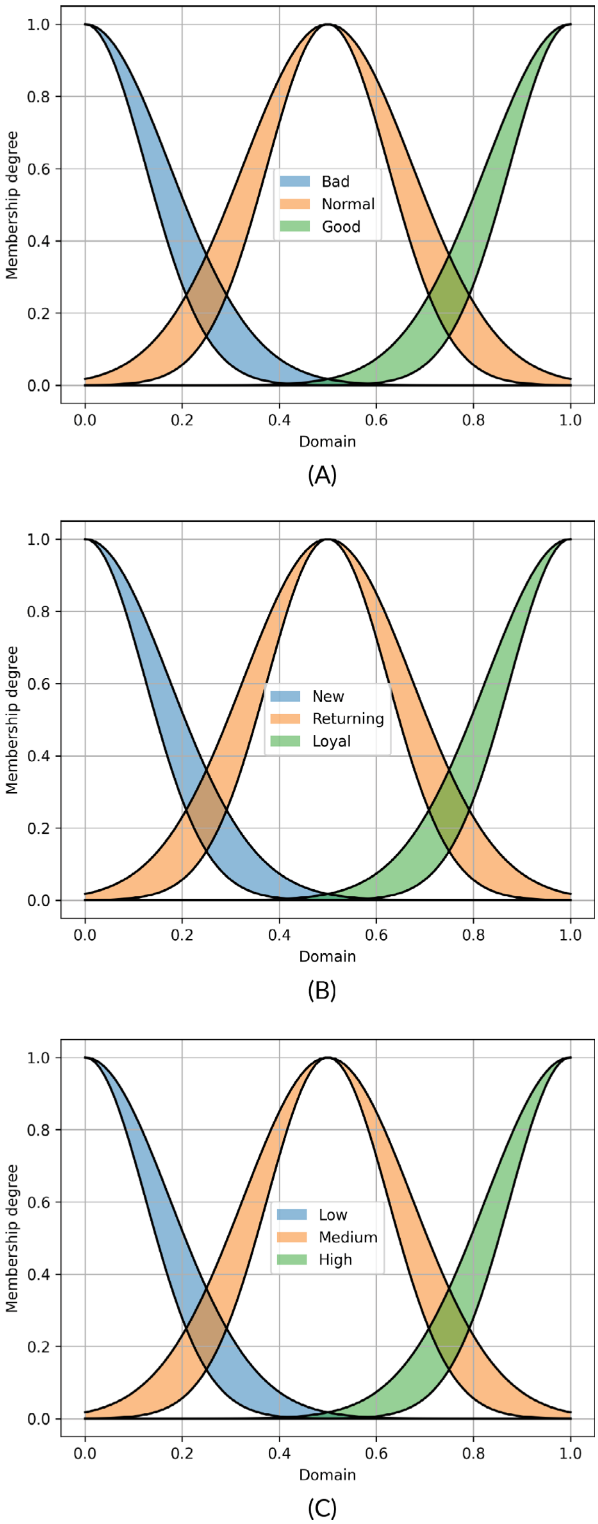

FIGURE 5 Fuzzy sets of the fuzzy inference system. (A) Fuzzy sets of the input variable edge accessibility; (B) Fuzzy sets of the input variable client type; (C) Fuzzy sets of the output variable service delay.

TABLE 2 Fuzzy rules of the fuzzy inference system.

	Client-type		
Edge accessibility	New	Returning	Loyal
Bad	High	High	Medium
Normal	High	Medium	Low
Good	Medium	Low	Low

up the level of uncertainty. It is expected that, as c converges to zero, the results of the stochastic scenario will converge to those obtained in the deterministic scenario. Also, we have utilized the value $c = 1$, which introduces a medium level of uncertainty.

In the stochastic-fuzzy scenario, two different types of uncertainty are considered. Hence, the service time Q_e can be either stochastic or fuzzy. For experimentation purposes, half of the service delays are modeled with the described Log-Normal probability distribution, while the remaining service delays are modeled with the fuzzy inference system presented next. Specifically, we have considered that the service time is stochastic when the end node of an arc has an even *id*, and fuzzy otherwise. Finally, regarding the pure fuzzy scenario, both the *type-1* and the more sophisticated *type-2* fuzzy sets are generally used to model fuzzy variables. This scenario represents the highest degree of uncertainty, where the fuzzy variables cannot be modeled using a probability distribution due to the lack of historical observations. In our experiments, *interval type-2* fuzzy sets are employed to model the service delay D_e of edge e. The details of the designed fuzzy inference system are described below:

- The edge accessibility and client type are considered input variables, expressed as values ranging from 0 to 1. Considering that deterministic instances do not specify these variables, we generate synthetic values for them. This is achieved by: (i) dividing the time required to travel from edge e to the depot by the time required to travel from the farthest edge to the depot; and (ii) dividing the pure service time of edge e by the maximum pure service time, respectively. Similarly, the service delay is considered as the output variable, expressed as a value ranging between 0 and 1. This value is then multiplied by the edge pure service time.
- The fuzzy sets describe the fuzzy representation of input and output variables. The edge accessibility and client type are both defined as three Gaussian IT2FS systems with uncertain standard deviation values. These can take the values *Bad*, *Normal*, or *Good* and *New*, *Returning*, or *Loyal* depending on the input crisp values, respectively. Similarly, the service delay is defined as three Gaussian IT2FS systems with uncertain SD values, which can take the values *Low*, *Medium*, or *High*. Figure 5 shows the membership functions of the described fuzzy sets for each of the three fuzzy variables.
- The fuzzy rules determine the service delay for a given edge accessibility and client type pair. Thus, a total of nine rules are defined for the different combinations of edge accessibility and client type. Table 2 shows the designed fuzzy rules for each service delay, where each cell in the table defines a rule. The table can be interpreted as follows: that is, if the edge accessibility is *Bad* and the client type is *New* then the service delay will be *High*; if the edge accessibility is *Normal* and the client type is *Returning* then the service delay will be *Medium*; finally, if the edge accessibility is *Good* and the client type is *Loyal*, then the service delay will be *Low*.
- The input variables with these membership functions are combined using the minimum and maximum norms to produce the fuzzy output following the fuzzy rules. Afterward, the enhanced IASC-type reduction algorithm is used along with the well-known centroid method to obtain the output crisp value from the *interval type-2* fuzzy sets.

6 | RESULTS AND DISCUSSION

Tables 3–6 present the results for the instances of the different datasets. The first column identifies the instances, while the second column depicts the time-based capacity (Q). The remaining columns show the obtained best and average results under the four considered scenarios. First, our best and average deterministic solutions (*OBD*/*OAD*) are reported and the best-found solutions are compared with the best-known solutions (*BKS*) provided by De Armas et al. [13]. In the second section of the table, we report the obtained results for the stochastic scenario. The *OBD-S*/*OAD-S* columns show the expected cost associated with the best and average deterministic solutions when they are evaluated under a stochastic scenario, with the corresponding level of uncertainty. To compute the expected cost, we have applied an intensive simulation process to the deterministic solutions. The main idea behind this process is to assess the deterministic solutions under different levels of uncertainty. The next columns (*OBS*/*OAS*) display the best and average solutions expected cost obtained using our fuzzy simheuristic approach for the stochastic version of the problem. This approach considers the stochastic elements during the search for a solution. The next section of the table offers the obtained results for the stochastic-fuzzy scenario. The *OBD-H*/*OAD-H* columns report the expected cost associated with

TABLE 3 Computational results for the *tcarp* dataset.

		Deterministic scenario				Stochastic scenario				Stochastic-fuzzy scenario				Fuzzy scenario			
		BKS	OBD	OAD	GAP (%)	OBD-S	OAD-S	OBS	OAS	OBD-H	OAD-H	OBH	OAH	OBD-F	OAD-F	OBF	OAF
Instance	Q	[1]	[2]	[3]	[1-2]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
tcarp-s1	60	104	104.0	105.8	0.00	114.0	113.3	110.4	113.2	124.2	126.9	121.6	126.9	138.5	140.1	134.5	140.1
tcarp-s1	50	108	108.0	108.0	0.00	117.2	117.0	115.8	116.9	128.7	131.5	128.6	132.1	147.5	147.1	144.1	146.4
tcarp-s1	40	112	112.0	112.4	0.00	119.9	121.1	119.6	120.8	136.4	136.7	132.1	135.6	152.8	153.4	150.3	153.8
tcarp-s2	60	156	156.0	156.2	0.00	165.8	165.8	164.9	165.9	188.0	184.5	180.8	185.2	214.9	213.1	205.7	214.6
tcarp-s2	50	158	159.0	159.3	0.63	170.5	169.8	168.3	170.1	195.8	194.5	186.5	193.1	230.6	228.5	213.4	224.7
tcarp-s2	40	164	164.0	165.7	0.00	174.6	176.3	173.6	177.2	202.1	200.5	195.1	199.9	230.1	232.5	229.6	234.4
tcarp-s3	60	215	215.0	215.4	0.00	234.2	236.8	231.8	234.9	270.2	275.9	260.0	270.0	302.4	308.9	297.5	307.3
tcarp-s3	50	229	229.0	230.0	0.00	246.6	248.2	246.1	248.0	289.5	289.9	277.7	288.2	326.1	331.3	320.9	329.1
tcarp-s3	40	245	245.0	249.6	0.00	263.2	266.2	262.3	266.2	309.3	307.7	289.2	303.7	349.4	355.3	345.5	352.8
tcarp-s4	60	146	146.0	146.0	0.00	154.4	154.3	154.0	154.4	172.1	173.2	169.3	173.2	203.1	200.6	195.3	200.6
tcarp-s4	50	162	162.0	162.0	0.00	169.1	169.3	168.9	169.4	180.1	183.2	179.1	182.5	209.4	208.7	203.9	208.2
tcarp-s4	40	174	174.0	174.0	0.00	181.2	181.3	181.1	181.3	194.4	194.5	191.3	194.4	222.5	224.5	219.5	223.5
tcarp-s5	60	140	140.0	140.0	0.00	149.6	150.1	149.5	150.1	173.6	171.6	167.5	170.9	181.8	188.6	181.8	187.9
tcarp-s5	50	149	149.0	149.0	0.00	162.5	161.4	157.8	159.8	183.3	183.2	175.3	179.1	210.3	205.2	193.4	199.5
tcarp-s5	40	165	165.0	165.0	0.00	174.6	174.9	174.1	174.9	198.9	198.3	192.2	197.7	222.4	219.2	213.3	219.5
tcarp-s6	60	104	105.0	105.7	0.96	110.2	111.2	110.1	111.2	128.4	129.6	126.3	129.7	138.9	141.0	138.8	141.0
tcarp-s6	50	107	107.0	109.4	0.00	114.4	116.3	114.2	116.7	132.7	133.4	127.3	132.8	149.1	146.0	137.2	144.0
tcarp-s6	40	113	113.0	113.0	0.00	118.5	118.6	118.1	118.6	137.2	137.6	132.8	137.1	161.6	153.4	146.9	153.4
tcarp-s7	60	68	68.0	68.1	0.00	71.2	71.3	71.1	71.4	77.5	78.8	76.3	78.7	87.5	89.5	87.4	89.5
tcarp-s7	50	68	68.0	68.5	0.00	71.2	71.8	71.1	71.8	77.5	80.1	77.5	80.1	87.5	92.7	87.5	92.7
tcarp-s7	40	68	68.0	69.5	0.00	71.2	72.8	71.1	72.9	78.1	81.1	77.6	80.5	89.5	93.6	89.5	93.6
tcarp-s8	60	83	83.0	83.0	0.00	86.4	86.5	86.3	86.4	95.7	94.8	91.7	94.8	115.9	107.2	103.4	107.2
tcarp-s8	50	83	83.0	83.3	0.00	86.6	87.2	86.4	87.2	97.2	97.2	94.6	97.2	108.1	110.2	103.5	110.2
tcarp-s8	40	87	87.0	87.2	0.00	90.6	91.2	90.6	91.2	105.0	104.9	101.8	104.2	117.4	119.0	114.8	118.2
tcarp-s9	60	177	177.0	177.8	0.00	187.2	188.7	186.3	189.0	226.1	225.1	210.9	216.5	248.1	252.3	238.9	246.9
tcarp-s9	50	193	193.0	193.6	0.00	203.0	203.6	202.6	204.5	237.2	238.6	231.1	237.1	276.4	271.5	264.1	270.7
tcarp-s9	40	221	221.0	221.2	0.00	229.9	230.3	229.5	230.9	264.3	264.9	257.9	261.2	299.8	298.3	289.9	298.0
tcarp-s10	60	171	171.0	172.3	0.00	183.6	183.9	180.3	184.0	216.2	214.6	204.7	214.8	246.5	243.3	234.2	236.5
tcarp-s10	50	180	181.0	181.4	0.56	192.4	192.6	192.3	192.9	223.1	223.2	212.5	221.5	251.6	251.3	240.3	249.6
tcarp-s10	40	192	192.0	192.0	0.00	203.2	202.8	202.3	203.0	238.3	237.7	223.5	235.4	280.1	275.3	265.2	273.1
Average:		144.7	144.8	145.5	0.1	153.9	154.5	153.0	154.5	176.0	176.5	169.7	175.1	200.0	200.1	193.0	198.9

the best and average deterministic solutions when they are evaluated under a hybrid stochastic-fuzzy scenario. Similarly, the next columns (*OBH*/*OAH*) show the best and average solutions expected cost obtained using our fuzzy simheuristic algorithm for the stochastic-fuzzy scenario of the problem. Finally, the last section of the table exhibits the obtained results for the fuzzy scenario. The *OBD-F*/*OAD-F* columns show the expected cost associated with the best and average deterministic solutions when they are evaluated under a fuzzy scenario, with the corresponding level of uncertainty. The last columns (*OBF*/*OAF*) display the best and average solutions expected cost obtained using our fuzzy simheuristic approach for the fuzzy version of the problem.

Figure 6 depicts an overview of Tables 3–6 about the performance of our algorithm using our best-found solutions for all the considered scenarios. In these box plots, the horizontal and vertical axes represent the four uncertainty scenarios and the percentage gap obtained with respect to the *BKS* reported in the literature, respectively. Notice that our best-found solutions using the fuzzy simheuristic approach nearly reach the *BKS* for the *tcarp*, *val*, and *rural* datasets, obtaining a gap about 0.07%, 0.69%, and 0.25%, respectively. Concerning the *tegl* dataset, we outperform the solutions provided in De Armas et al. [13] by about −5.41%. These results highlight the quality of our algorithm since our approach provides highly competitive solutions for the deterministic TCARP.

Regarding the uncertainty scenarios, which is the main contribution of this paper, the obtained results show that the best solutions provided by our approach for the different uncertainty scenarios clearly outperform the best solutions for the deterministic TCARP when they are simulated under the corresponding level of uncertainty. In other words, our best-found solutions

TABLE 4 Computational results for the *tegl* dataset.

		Deterministic scenario				Stochastic scenario				Stochastic-fuzzy scenario				Fuzzy scenario			
		BKS	OBD	OAD	GAP (%)	OBD-S	OAD-S	OBS	OAS	OBD-H	OAD-H	OBH	OAH	OBD-F	OAD-F	OBF	OAF
Instance	Q	[1]	[2]	[3]	[1-2]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
tegl-e1-A	360	1367	1198.0	1200.0	−12.36	1297.7	1306.7	1273.0	1297.7	1523.8	1538.6	1408.5	1502.8	1740.8	1716.6	1613.3	1689.5
tegl-e1-C	240	1655	1456.0	1456.3	−12.02	1552.1	1551.3	1542.2	1552.8	1784.6	1847.1	1784.3	1835.4	2067.4	2099.6	2060.5	2093.9
tegl-e2-A	360	2023	1935.0	1940.2	−4.35	2151.8	2131.4	2102.6	2127.7	2490.1	2470.2	2419.3	2466.5	2725.0	2708.3	2669.9	2716.7
tegl-e2-C	240	2481	2379.0	2393.4	−4.11	2560.5	2549.6	2532.3	2550.1	2928.6	2926.0	2847.9	2911.5	3338.1	3352.0	3278.7	3337.9
tegl-e3-A	360	2439	2378.0	2381.6	−2.50	2598.6	2576.1	2548.1	2574.7	2960.9	3001.6	2905.3	2971.8	3308.4	3321.8	3260.8	3307.5
tegl-e3-C	240	3019	3021.0	3026.8	0.07	3212.1	3227.9	3207.6	3232.6	3597.0	3649.8	3562.2	3623.8	4066.6	4089.5	3991.8	4061.4
tegl-e4-A	360	2686	2701.0	2713.9	0.56	3000.1	2976.8	2936.3	2969.7	3441.7	3457.1	3362.5	3428.6	3799.6	3831.0	3739.0	3812.0
tegl-e4-C	240	3270	3297.0	3316.0	0.83	3525.9	3534.2	3506.3	3527.4	4079.8	4065.8	3982.6	4046.5	4662.9	4579.0	4492.9	4565.5
tegl-s1-A	360	2225	1740.0	1740.6	−21.80	1909.1	1896.1	1884.6	1900.3	2272.9	2276.6	2185.9	2251.7	2537.9	2544.4	2517.8	2535.4
tegl-s1-C	240	3094	2518.0	2526.8	−18.62	2666.3	2675.1	2651.5	2671.1	3043.4	3076.4	2993.3	3052.9	3364.9	3378.8	3360.0	3372.8
tegl-s2-A	360	4114	3938.0	3951.2	−4.28	4321.9	4353.1	4295.4	4345.1	5041.3	5048.7	4941.3	5029.5	5524.6	5586.0	5494.9	5570.3
tegl-s2-C	240	5605	5386.0	5442.6	−3.91	5706.3	5760.6	5704.5	5751.7	6503.0	6556.5	6416.3	6525.4	7242.1	7264.7	7081.2	7233.0
tegl-s3-A	360	4436	4320.0	4332.5	−2.61	4796.5	4766.2	4676.1	4757.2	5578.4	5512.8	5362.9	5491.0	6032.0	6077.9	5897.2	6050.8
tegl-s3-C	240	6122	5965.0	6018.3	−2.56	6362.7	6392.0	6324.7	6397.0	7286.9	7290.7	7099.9	7229.6	8075.6	8022.7	7906.5	7997.1
tegl-s4-A	360	5202	5226.0	5251.5	0.46	5745.9	5815.3	5742.8	5815.9	6655.0	6711.9	6644.4	6700.6	7354.3	7421.1	7344.9	7426.8
tegl-s4-C	240	7282	7325.0	7367.4	0.59	7809.7	7842.1	7777.1	7835.8	8941.0	8905.2	8812.0	8880.0	9757.8	9748.9	9664.9	9732.6
Average:		3563.8	3423.9	3441.2	−5.4	3701.1	3709.6	3669.1	3706.7	4258.0	4270.9	4170.5	4246.7	4724.9	4733.9	4648.4	4718.9

TABLE 5 Computational results for the *val* dataset.

		Deterministic scenario				Stochastic scenario				Stochastic-fuzzy scenario				Fuzzy scenario			
		BKS	OBD	OAD	GAP (%)	OBD-S	OAD-S	OBS	OAS	OBD-H	OAD-H	OBH	OAH	OBD-F	OAD-F	OBF	OAF
Instance	Q	[1]	[2]	[3]	[1-2]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
val6	918	221	223.0	233.3	0.90	245.0	245.5	245.0	245.7	294.4	293.5	291.2	293.6	346.7	346.7	344.6	346.7
val6	648	223	223.0	233.9	0.00	245.4	246.3	245.1	246.3	293.8	294.4	290.4	294.4	348.6	347.4	344.6	347.4
val6	270	239	241.0	246.6	0.84	265.2	274.1	265.0	271.4	316.6	326.7	314.9	322.5	375.3	388.6	372.7	384.1
val7	465	279	279.0	280.5	0.00	307.0	309.9	306.9	308.6	369.2	373.6	366.5	372.2	441.5	446.6	439.9	443.6
val7	349	283	283.0	283.8	0.00	311.2	313.6	310.9	312.2	376.1	380.0	375.5	378.4	449.3	451.0	447.5	449.3
val7	151	352	352.0	355.9	0.00	387.0	397.7	387.0	391.4	474.7	486.5	470.4	477.8	561.1	580.1	560.7	568.6
val8	579	386	390.0	399.0	1.04	429.1	447.3	428.8	438.9	484.2	501.4	482.5	497.3	552.3	570.5	549.5	564.0
val8	434	386	387.0	403.5	0.26	425.6	443.8	425.2	443.8	481.0	517.0	480.4	505.2	545.4	588.3	542.5	571.7
val8	188	519	523.0	546.5	0.77	575.4	612.4	574.9	601.1	673.6	721.9	671.9	699.9	769.2	834.4	766.7	805.0
val9	511	324	329.0	332.8	1.54	361.9	367.0	360.9	366.2	429.7	437.6	427.8	436.7	510.9	519.4	508.7	516.7
val9	380	326	327.0	342.5	0.31	360.9	378.0	359.7	376.7	430.0	450.4	428.8	449.3	514.8	536.4	511.9	534.5
val9	304	338	342.0	354.6	1.18	377.3	390.1	376.9	390.1	450.8	466.9	447.4	464.3	541.0	560.8	539.9	555.6
val9	152	445	445.0	464.4	0.00	490.4	515.7	489.2	510.9	581.7	610.2	578.9	601.4	718.8	762.8	715.7	744.4
val10	480	438	439.0	443.2	0.23	483.1	491.0	483.0	487.6	565.7	569.2	563.8	566.6	652.1	662.5	648.0	656.5
val10	365	449	455.0	459.3	1.34	500.3	508.7	499.6	505.3	589.0	596.8	588.0	590.7	684.2	694.8	681.6	685.2
val10	288	465	472.0	489.6	1.51	519.3	545.6	518.6	538.6	610.2	644.1	608.8	633.6	708.8	752.1	704.6	737.0
val10	144	844	859.0	867.0	1.78	945.0	957.6	944.8	953.6	1131.5	1164.3	1129.8	1147.9	1373.9	1412.0	1370.9	1387.6
Average:		383.4	386.4	396.3	0.7	425.2	437.9	424.8	434.6	503.1	519.7	501.0	513.6	593.8	615.0	591.2	605.7

for the deterministic version of the problem might be sub-optimal for the different uncertainty versions. Moreover, the average solutions of the 10 execution runs are always relatively close to the best solutions obtained in those 10 executions runs for each instance-uncertainty level combination, with an average gap of 0.5% with respect to the gap obtained by the best-found solutions. This implies that our fuzzy simheuristic algorithm is expected to perform quite well every time it is run. Hence, the importance of integrating fuzzy-simulation methods when dealing with optimization problems with fuzzy and stochastic uncertainty. Notice also that the *OBD* can be seen as a reference lower bound in an ideal scenario with perfect information—that is, without uncertainty—for the expected cost under uncertainty conditions. Similarly, the *OBD-S*, *OBD-H*, and *OBD-F* can be seen as an upper bound for the expected cost in the different uncertainty scenarios. All things considered, the results presented confirm the importance of accounting for the uncertainty during the solution search process, since this might have a significant impact on the quality of our best-found solutions. For instance, the simulation of our best deterministic solution in a fuzzy scenario provides a much higher expected cost, while our best fuzzy solution gives a lower expected cost. Thus, considering uncertainty elements in the search process produces better results than just solving a deterministic version of the problem and then applying this solution in a real-life scenario with stochastic and fuzzy uncertainty.

Once the computational results using the fuzzy simheuristic approach have been discussed, we could compare our approach to other classical frameworks, such as stochastic programming and robust optimization. Pages-Bernaus et al. [52] compared the simheuristic methodology with stochastic programming and concluded that simheuristics perform better than stochastic programming for larger instances. A robust optimization approach assumes a uniform distribution, while our method models times using different distributions. Thus, we compare our fuzzy simheuristic method with a robust optimization approach after generating the service delays with a uniform distribution. To illustrate this comparison, the following analysis has been conducted for the *rural2* instance. We generate 100 samples following the Log-Normal distribution presented in Section 5, and compute the minimum and maximum values of the samples. These values will be the lower and upper bound of a uniform distribution, which will be used to generate the new service delays. As illustrated in Table 7, the best and average stochastic results of our fuzzy simheuristic approach have been compared with the best and average results obtained by the robust optimization method. The percentage gaps show that the simheuristic approach computes more optimal solutions compared to the robust optimization. This is to be expected, as robust optimization equally considers the best- and worst-case service delays as it searches for new solutions, which could be useful for obtaining reliable solutions in risk aversion decision scenarios. Hence, we optimize for minimizing the maximum cost of the solution and consequently generate solutions with a certain measure of robustness, while our fuzzy simheuristic approach minimizes the solution expected cost.

In summary, the fuzzy simheuristic methodology presented here offers unique advantages over other existing approaches for solving the TCARP under uncertainty. Some of the potential benefits of our approach are presented next: (i) an efficient metaheuristic is employed as the basis for solving the TCARP under uncertainty, which ensures that good quality solutions are generated even for instances of large sizes; (ii) the simulation component guides the search of new promising solutions, which considers the TCARP uncertainty conditions during the search for near-optimal solutions; and (iii) the simulation component is able to incorporate any probability distribution with a known mean, either theoretical or experimental, or fuzzy inference system fuzzy sets and fuzzy inference rules to model the uncertainty conditions. However, as with any methodology, there are also limitations when using the extended simheuristic. Some of these limitations are presented next: (i) the metaheuristic basis does not guarantee an optimal solution to an optimization problem, but rather a good enough solution in a reasonable amount of time; (ii) the simulation component requires additional effort when designing the probability distributions or fuzzy inference systems to model the uncertainty conditions; and (iii) the integration of a simulation component within a metaheuristic requires high computational efforts, especially simulation components including type-2 fuzzy inference systems.

7 | CONCLUSIONS

In this paper, we have considered a rich version of the Arc Routing Problem in which: (i) the capacity constraints are given by a time threshold instead of a loading factor; and *(ii)* travel and service times might show a deterministic, stochastic, or even fuzzy behavior. In this context, we have modeled fuzzy elements by employing a type-2 fuzzy set system, which represents one of the novelties of the paper. Then, in order to solve the TCARP under general uncertainty conditions, a fuzzy-simheuristic algorithm has been proposed. This algorithm combines an iterated local search metaheuristic framework with Monte Carlo simulation and a T2FS system.

In order to test our approach, four data sets from the TCARP literature have been adapted and extended, so they consider both fuzzy and stochastic uncertainty. A series of computational experiments contribute to validate our approach by showing a common pattern in all instances and considered uncertainty conditions: as we increase the level of uncertainty (from stochastic to hybrid stochastic-fuzzy and pure fuzzy), the quality of the solution is increasingly reduced when compared with a deterministic scenario, which can be considered as an ideal (yet unrealistic) scenario. As a general conclusion, one should never employ near-optimal solutions to deterministic versions of a TCARP in a real-life scenario with fuzzy and stochastic elements. Another interesting contribution of the paper is that we have been able to improve the best-known solutions for the stochastic version of the problem in the case of the *tegl* dataset.

One open research line that we want to highlight is the integration of the fuzzy-simheuristic algorithm with the machine and statistical learning methods, with the goal of improving the quality of the feedback provided by the simulation component. This, in turn, could be employed to select solutions of special interest. Machine learning methods could also be employed to build surrogate models that make computations faster by saving many simulation runs.

TABLE 6 Computational results for the *rural* dataset.

		Deterministic scenario				Stochastic scenario				Stochastic-fuzzy scenario				Fuzzy scenario			
		BKS	OBD	OAD	GAP (%)	OBD-S	OAD-S	OBS	OAS	OBD-H	OAD-H	OBH	OAH	OBD-F	OAD-F	OBF	OAF
Instance	Q	[1]	[2]	[3]	[1-2]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
rural1	420	1195	1196.0	1198.8	0.08	1284.0	1291.4	1283.1	1292.3	1477.4	1481.9	1459.3	1479.7	1680.9	1679.4	1663.2	1673.0
rural1	400	1195	1195.0	1197.7	0.00	1285.4	1293.6	1281.1	1290.8	1461.6	1479.6	1458.2	1478.0	1656.5	1671.3	1653.8	1673.2
rural1	360	1193	1193.0	1195.7	0.00	1292.0	1289.4	1279.9	1287.6	1502.2	1480.7	1450.7	1476.0	1676.7	1677.0	1665.3	1671.9
rural2	420	1844	1847.0	1854.3	0.16	2008.4	2027.3	2008.0	2027.8	2388.3	2407.8	2368.7	2402.8	2770.0	2791.5	2748.5	2782.2
rural2	400	1840	1841.0	1847.1	0.05	1986.2	2006.8	1985.3	2003.7	2359.4	2393.8	2355.1	2386.3	2739.2	2764.5	2731.7	2761.5
rural2	360	1836	1845.0	1847.7	0.49	2010.2	2006.8	2008.2	2003.6	2381.9	2372.2	2368.9	2368.9	2788.7	2737.9	2748.5	2735.3
rural3	420	2218	2222.0	2233.3	0.18	2409.5	2441.0	2408.9	2440.0	2782.5	2799.9	2747.1	2797.7	3086.9	3100.2	3048.9	3098.4
rural3	400	2193	2195.0	2206.1	0.09	2436.1	2416.9	2382.1	2410.4	2746.5	2749.4	2717.3	2746.9	3063.7	3063.8	3001.7	3060.9
rural3	360	2182	2182.0	2196.1	0.00	2413.2	2395.6	2409.6	2395.4	2806.5	2728.3	2772.4	2728.3	3119.0	3040.5	3060.2	3040.2
rural4	420	2518	2519.0	2536.3	0.04	2683.0	2711.1	2681.8	2711.5	3226.6	3251.4	3198.3	3251.2	3604.7	3621.7	3580.0	3621.7
rural4	400	2489	2480.0	2494.8	-0.36	2646.9	2651.9	2617.4	2648.1	3161.2	3132.9	3082.6	3131.8	3550.0	3513.2	3478.7	3513.2
rural4	360	2450	2498.0	2502.1	1.96	2662.6	2669.1	2656.2	2667.8	3206.1	3211.9	3168.1	3186.9	3584.7	3593.9	3540.0	3564.4
rural5	420	1674	1676.0	1687.4	0.12	1774.7	1771.9	1772.5	1771.9	2101.7	2095.1	2048.5	2095.0	2422.9	2381.3	2375.1	2377.8
rural5	400	1655	1670.0	1678.6	0.91	1744.1	1765.1	1743.1	1764.7	2051.8	2074.1	2028.0	2074.1	2352.8	2358.1	2315.9	2358.1
rural5	360	1666	1666.0	1676.8	0.00	1755.5	1758.7	1755.0	1758.8	2016.8	2043.4	2004.9	2043.2	2315.9	2329.1	2297.1	2327.7
Average:		1876.5	1881.7	1890.2	0.2	2026.1	2033.1	2018.1	2031.6	2378.0	2380.1	2348.5	2376.5	2694.2	2688.2	2660.6	2684.0

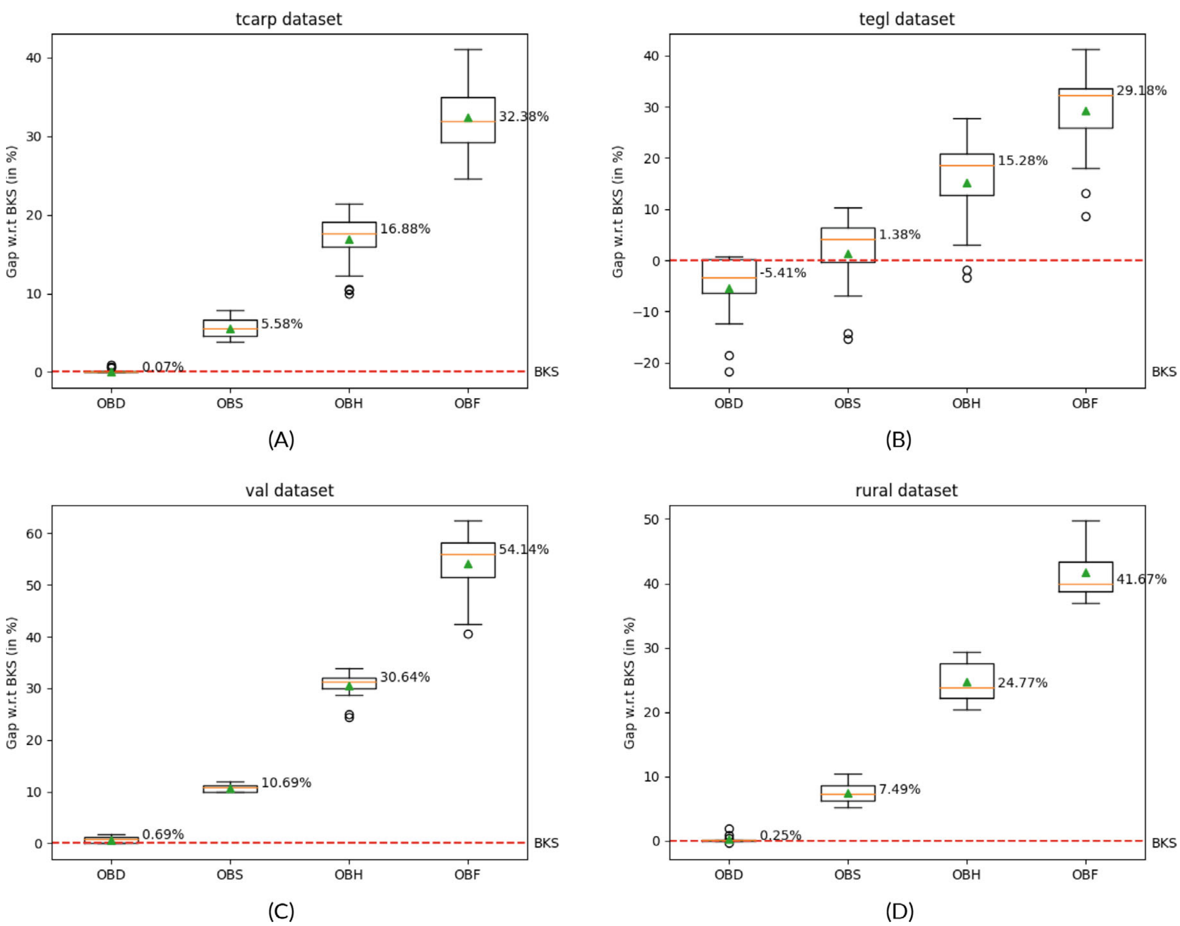

FIGURE 6 Percentage gaps of our fuzzy simheuristic approach w.r.t. the best-known solutions (BKS). (A) Gaps of *tcarp* instances w.r.t the BKS; (B) Gaps of *tegl* instances w.r.t the BKS; (C) Gaps of *val* instances w.r.t the BKS; (d) Gaps of *rural* instances w.r.t the BKS.

TABLE 7 Computational results for the robust optimization comparison.

		Simheuristic		Robust optimization			
		OBS	OAS	OBS-R	OAS-R	GAP (%)	GAP (%)
Instance	Q	[1]	[2]	[3]	[4]	[1-3]	[2-4]
rural2	420	2008.0	2027.8	2687.3	2704.2	33.8	33.4
rural2	400	1985.3	2003.7	2642.2	2683.7	33.1	33.9
rural2	360	2008.2	2003.6	2642.1	2665.4	31.6	33.0

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science (PID2019-111100RB-C21/AEI/10.13039/501100011033), as well as by the Barcelona Council and the "la Caixa" Foundation under the framework of the Barcelona Science Plan 2020-2023 (grant 21S09355-001) and Generalitat Valenciana (PROMETEO/2021/065).

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Xabier A. Martin https://orcid.org/0000-0003-4182-0120
Javier Panadero https://orcid.org/0000-0002-3793-3328
David Peidro https://orcid.org/0000-0001-8678-6881
Elena Perez-Bernabeu https://orcid.org/0000-0002-9221-7623
Angel A. Juan https://orcid.org/0000-0003-1392-1776

REFERENCES

[1] V. Abu-Marrul, R. Martinelli, S. Hamacher, and I. Gribkovskaia, *Simheuristic algorithm for a stochastic parallel machine scheduling problem with periodic re-planning assessment*, Ann. Oper. Res. **320** (2022), 547–572.
[2] S. Alvarez Fernandez, D. Ferone, A. A. Juan, and D. Tarchi, *A simheuristic algorithm for video streaming flows optimisation with qos threshold modelled as a stochastic single-allocation p-hub median problem*, J. Simul. **16** (2022), 480–493.
[3] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, *Simulation optimization: A review of algorithms and applications*, 4OR **12** (2014), 301–333.
[4] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, *Simulation optimization: A review of algorithms and applications*, Ann. Oper. Res. **240** (2016), 351–380.
[5] E. Babaee Tirkolaee, I. Mahdavi, M. M. Seyyed Esfahani, and G. W. Weber, *A hybrid augmented ant colony optimization for the multi-trip capacitated arc routing problem under fuzzy demands for urban solid waste management*, Waste Manag. Res. **38** (2020), 156–172.
[6] A. Banu, M. Y. Ali, M. A. Rahman, and M. Konneh, *Stability of micro dry wire edm: Ofat and doe method*, Int. J. Adv. Manuf. Technol. **106** (2020), 4247–4261.
[7] E. Bartolini, J. F. Cordeau, and G. Laporte, *An exact algorithm for the capacitated arc routing problem with deadheading demand*, Oper. Res. **61** (2013), 315–327.
[8] O. Castillo and P. Melin, *Uncertain rule-based fuzzy systems: Introduction and new directions*, Springer, Berlin, Heidelberg, 2008.
[9] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz. *Type-2 fuzzy logic: Theory and applications*, 2007 IEEE Int. Conf. Granul. Comput. (GRC 2007), San Jose, CA, USA, 2007, p. 145.
[10] M. Cengiz Toklu, *A fuzzy multi-criteria approach based on clarke and wright savings algorithm for vehicle routing problem in humanitarian aid distribution*, J. Intell. Manuf. **34** (2022), 2241–2261.
[11] M. Chica, A. A. Juan, C. Bayliss, O. Cordon, and W. D. Kelton, *Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation*, Stat. Oper. Res. Trans. **44** (2020), 1–24.
[12] Á. Corberán and G. Laporte, *Arc routing: Problems, methods, and applications*, Society for Industrial and Applied Mathematics, Philadelphia, 2014.
[13] J. De Armas, P. Keenan, A. A. Juan, and S. McGarraghy, *Solving large-scale time capacitated arc routing problems: From real-time heuristics to metaheuristics*, Ann. Oper. Res. **273** (2019), 135–162.
[14] A. De León, E. Lalla-Ruiz, B. Melián-Batista, and J. Moreno-Vega, *A simulation–optimization framework for enhancing robustness in bulk berth scheduling*, Eng. Appl. Artif. Intel. **103** (2021), 104276.
[15] A. De Maio, D. Laganà, R. Musmanno, and F. Vocaturo, *Arc routing under uncertainty: Introduction and literature review*, Comput. Oper. Res. **135** (2021), 105442.
[16] J. Faulin, A. A. Juan, C. Serrat, and V. Bargueno, *Predicting availability functions in time-dependent complex systems with SAEDES simulation algorithms*, Reliab. Eng. System Saf. **93** (2008), 1761–1771.

[17] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas. *A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games*, The 14th IEEE Int. Conf. Fuzzy Syst., 2005. FUZZ'05, Reno, NV, USA, 2005, pp. 359–364.

[18] F. Glover, J. P. Kelly, and M. Laguna. *New advances and applications of combining simulation and optimization*, Proc. 1996 Winter Simul. Conf., Piscataway, N.J., 1996, pp. 144–152.

[19] R. Gojković, G. Durić, D. Tadić, S. Nestić, and A. Aleksić, *Evaluation and selection of the quality methods for manufacturing process reliability improvement–intuitionistic fuzzy sets and genetic algorithm approach*, Mathematics **9** (2021), 1531.

[20] B. L. Golden and R. T. Wong, *Capacitated arc routing problems*, Networks **11** (1981), 305–315.

[21] S. Gonzalez-Martin, A. A. Juan, D. Riera, M. Elizondo, and J. Ramos, *A simheuristic algorithm for solving the arc routing problem with stochastic demands*, J. Simul. **12** (2018), 53–66.

[22] A. Grasas, A. A. Juan, J. Faulin, J. de Armas, and H. Ramalhinho, *Biased randomization of heuristics using skewed probability distributions: A survey and some applications*, Comput. Indus. Eng. **110** (2017), 216–228.

[23] H. A. Hagras, *A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots*, IEEE Trans. Fuzzy Syst. **12** (2004), 524–539.

[24] I. Jackson, J. Tolujevs, and T. Reggelin, *The combination of discrete-event simulation and genetic algorithm for solving the stochastic multi-product inventory optimization problem*, Trans. Telecommun. **19** (2018), 233–243.

[25] M. Jamshidi, *Large-scale systems: Modeling, control, and fuzzy logic*, Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1996.

[26] A. A. Juan, H. R. Lourenço, M. Mateo, R. Luo, and Q. Castella, *Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues*, Int. Trans. Oper. Res. **21** (2014), 103–126.

[27] N. Kabadayi and M. Dehghanimohammadabadi, *Multi-objective supplier selection process: A simulation–optimization framework integrated with mcdm*, Ann. Oper. Res. **319** (2022), 1607–1629.

[28] N. N. Karnik, J. M. Mendel, and Q. Liang, *Type-2 fuzzy logic systems*, IEEE Trans. Fuzzy Syst. **7** (1999), 643–658.

[29] P. Keenan. *Tcarp large rural datasets*. 2017.

[30] P. Keenan. *Lower bounds for the time capacitated arc routing problem*, Tech. Rep., UCD Business School, University College Dublin, 2005.

[31] P. Keenan and M. Naughton, *Arc routing for rural Irish networks*, Springer US, Boston, 1996, pp. 599–606.

[32] P. Keenan, J. Panadero, A. A. Juan, R. Mart, and S. McGarraghy, *A strategic oscillation simheuristic for the time capacitated arc routing problem with stochastic demands*, Comput. Oper. Res. **133** (2021), 105377.

[33] J. S. Kim and B. J. Yum, *Selection between weibull and lognormal distributions: A comparative simulation study*, Comput. Stat. Data Anal. **53** (2008), 477–485.

[34] G. Kirlik and A. Sipahioglu, *Capacitated arc routing problem with deadheading demands*, Comput. Oper. Res. **39** (2012), 2380–2394.

[35] P. Kundu, S. Majumder, S. Kar, and M. Maiti, *A method to solve linear programming problem with interval type-2 fuzzy parameters*, Fuzzy Optim. Decision Making **18** (2019), 103–130.

[36] C. P. Lam, M. Masek, L. Kelly, M. Papasimeon, and L. Benke, *A simheuristic approach for evolving agent behaviour in the exploration for novel combat tactics*, Oper. Res. Perspect. **6** (2019), 100123.

[37] A. M. Law, *Simulation modeling and analysis*, 5th ed., McGraw-Hill, New York, 2015.

[38] O. Linda and M. Manic. *Uncertainty modeling for interval type-2 fuzzy logic systems based on sensor characteristics*, 2011 IEEE Symp. Adv. Type-2 Fuzzy Logic Syst. (T2FUZZ), Paris, France, 2011, pp. 31–37.

[39] R. Liu, X. Xie, K. Yu, and Q. Hu, *A survey on simulation optimization for the manufacturing system operation*, Int. J. Model. Simul. **38** (2018), 116–127.

[40] W. Long-Fei and S. Le-Yuan, *Simulation optimization: A review on theory and applications*, Acta Automat. Sin. **39** (2013), 1957–1968.

[41] C. Lynch, H. Hagras, and V. Callaghan, *Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines*, IEEE Int. Conf. Fuzzy Syst. **2006** (2006), 1446–1453.

[42] S. Majumder, S. Kar, and T. Pal, *Uncertain multi-objective chinese postman problem*, Soft. Comput. **23** (2019), 11557–11572.

[43] J. Men, P. Jiang, and H. Xu, *A chance constrained programming approach for hazmat capacitated vehicle routing problem in type-2 fuzzy environment*, J. Clean. Prod. **237** (2019), 117754.

[44] J. M. Mendel, *Uncertain rule-based fuzzy systems: Introduction and new directions*, Springer, New York City, USA, 2017.

[45] J. M. Mendel and R. B. John, *Type-2 fuzzy sets made simple*, IEEE Trans. Fuzzy Syst. **10** (2002), 117–127.

[46] J. Mendel and Q. Liang. *Pictorial comparisons of type-1 and type-2 fuzzy logic systems*, Proc. IASTED Int. Conf. Intell. Syst. Control, Santa Barbara, CA, 1999.

[47] J. M. Mendel, R. I. John, and F. Liu, *Interval type-2 fuzzy logic systems made simple*, IEEE Trans. Fuzzy Syst. **14** (2006), 808–821.

[48] E. Negri, V. Pandhare, L. Cattaneo, J. Singh, M. Macchi, and J. Lee, *Field-synchronized digital twin framework for production scheduling with uncertainty*, J. Intell. Manuf. **32** (2021), 1207–1228.

[49] A. Nieto, M. Serra, A. A. Juan, and C. Bayliss, *A GA-simheuristic for the stochastic and multi-period portfolio optimisation problem with liabilities*, J. Simul. **0** (2022), 1–14.

[50] D. Oliva, P. Copado, S. Hinojosa, J. Panadero, D. Riera, and A. A. Juan, *Fuzzy simheuristics: Solving optimization problems under stochastic and uncertainty scenarios*, Math. **8** (2020), 1–19.

[51] C. S. Orloff, *A fundamental problem in vehicle routing*, Networks **4** (1974), 35–64.

[52] A. Pagès-Bernaus, H. Ramalhinho, A. A. Juan, and L. Calvet, *Designing e-commerce supply chains: A stochastic facility–location approach*, Int. Trans. Oper. Res. **26** (2019), 507–528.

[53] J. Panadero, J. Doering, R. Kizys, A. A. Juan, and A. Fito, *A variable neighborhood search simheuristic for project portfolio selection under uncertainty*, J. Heur. **26** (2020a), 353–375.

[54] J. Panadero, A. A. Juan, C. Bayliss, and C. Currie, *Maximising reward from a team of surveillance drones: A simheuristic approach to the stochastic team orienteering problem*, Eur. J. Indus. Eng. **14** (2020b), 485–516.

[55] D. Pecin and E. Uchoa, *Comparative analysis of capacitated arc routing formulations for designing a new branch-cut-and-price algorithm*, Transplant. Sci. **53** (2019), 1673–1694.

[56] M. S. Pishvaee, J. Razmi, and S. A. Torabi, *Robust possibilistic programming for socially responsible supply chain network design: A new approach*, Fuzzy Set. Syst. **206** (2012), 1–20.

[57] M. Rabe, M. Deininger, and A. A. Juan, *Speeding up computational times in simheuristics combining genetic algorithms with discrete-event simulation*, Simul. Model. Pract. Theory **103** (2020), 102089.

[58] M. A. Sanchez, O. Castillo, and J. R. Castro, *Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems*, Expert Syst. Appl. **42** (2015), 5904–5914.
[59] D. Tadić, A. Đorđević, A. Aleksić, and S. Nestić, *Selection of recycling centre locations by using the interval type-2 fuzzy sets and two-objective genetic algorithm*, Waste Manag. Res. **37** (2019), 26–37.
[60] E. Tekin and I. Sabuncuoglu, *Simulation optimization: A comprehensive review on theory and applications*, IIE Trans. **36** (2004), 1067–1081.
[61] R. Tordecilla, L. Martins, J. Panadero, P. Copado, E. Perez-Bernabeu, and A. A. Juan, *Fuzzy simheuristics for optimizing transportation systems: Dealing with stochastic and fuzzy uncertainty*, Appl. Sci. **11** (2021), 1–22.
[62] P. Villarinho, J. Panadero, L. Pessoa, A. A. Juan, and F. Oliveira, *A simheuristic algorithm for the stochastic permutation flow-shop problem with delivery dates and cumulative payoffs*, Int. Trans. Oper. Res. **28** (2021), 716–737.
[63] H. F. Wang and Y. P. Wen, *Time-constrained chinese postman problems*, Comput. Math. Appl. **44** (2002), 375–387.
[64] M. Wazed, S. Ahmed, Y. Nukman, et al., *Uncertainty factors in real manufacturing environment*, Aus. J. Basic Appl. Sci. **3** (2009), 342–351.
[65] S. Wøhlk, *A decade of capacitated arc routing*, Springer US, Boston, 2008, 29–48.
[66] D. Wu and J. M. Mendel, *Uncertainty measures for interval type-2 fuzzy sets*, Inform. Sci. **177** (2007), 5378–5393.
[67] L. A. Zadeh, *Fuzzy sets*, Inf. Control. **8** (1965), 338–353.
[68] L. A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning–I*, Inform. Sci. **8** (1975), 199–249.
[69] L. A. Zadeh, *Toward a generalized theory of uncertainty (gtu)–an outline*, Inform. Sci. **172** (2005), 1–40.
[70] F. Zandieh and S. F. Ghannadpour, *A comprehensive risk assessment view on interval type-2 fuzzy controller for a time-dependent hazmat routing problem*, Eur. J. Oper. Res. **305** (2022), 685–707.
[71] H. J. Zimmermann, *Fuzzy programming and linear programming with several objective functions*, Fuzzy Set. Syst. **1** (1978), 45–55.
[72] H. J. Zimmermann, *Fuzzy set theory and its applications*, Springer Science & Business Media, Dordrecht, Netherlands, 2011.

How to cite this article: X. A. Martin, J. Panadero, D. Peidro, E. Perez-Bernabeu, and A. A. Juan, *Solving the time capacitated arc routing problem under fuzzy and stochastic travel and service times*, Networks. **82** (2023), 318–335. https://doi.org/10.1002/net.22159