Author's copy of:
C. C. Muñoz, J. J. Palacios-Alonso, C. R. Vela and S. Afsar, "Solving a Vehicle Routing Problem with uncertain demands and adaptive credibility thresholds," 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Padua, Italy, 2022.
Doi: 10.1109/FUZZ-IEEE55066.2022.9882650.

Solving a Vehicle Routing Problem with uncertain demands and adaptive credibility thresholds

Carlos Carmona Muñoz
Independent Researcher
Spain
carlosCarmona491@outlook.com

Juan J. Palacios-Alonso
University of Oviedo
Spain
palaciosjuan@uniovi.es

Camino R. Vela
University of Oviedo
Spain
crvela@uniovi.es

Sezin Afsar
University of Oviedo
Spain
afsarsezin@uniovi.es

***Abstract*—Vehicle Routing Problem is an optimization problem of great interest in many real world scenarios such as waste management, delivery routing, etc. When employed in application contexts, there are several constraints that have to be considered such as vehicle fleet capacity and time windows. In real-world cases, it is common to find uncertainty in some parameters such as customer demands or route costs. This work proposes a way to address demand uncertainty based on fuzzy logic and adaptive credibility thresholds joined to a memetic algorithm to find the route assignments with minimum total cost. This approach is tested over several fuzzified benchmark instances and case scenarios to validate its adequacy and performance.**

***Index Terms*—Capacitated VRP, Time Windows, Fuzzy demands, Credibility, Memetic algorithm.**

I. Introduction

The vehicle routing problem (VRP) is a well-known NP-hard optimization problem [1] whose importance has increased over time with the ever-growing demand for package delivery services, efficient waste management and some other transport logistic domains where the operational costs are a main factor. The problem consists of, given a customer set's demands and locations and a fleet of vehicles, finding a set of minimum cost routes that the vehicles have to travel while fulfilling certain conditions, such as starting and ending at the depot and serving each customer only once. This, however, is the classic problem statement. In order to make meaningful contributions to real world contexts, the VRP has been reformulated in various ways, considering the nature of the vehicle fleets (capacitated [2], heterogeneous [3], [4]), time windows [3], [5], multiple depots [4], pick-up and delivery tasks [5], etc.)

Depending on the VRP application domain, different constraints need to be introduced that make the problem even harder to solve efficiently. In this work, we tackle a variant with time windows and uncertain demands. This variant is common in waste management, which presents two main characteristics:

- **Accessibility**: Waste collection cannot be done at any time of the day. In addition to the general (nightly) time interval during which the waste can be collected, there can be other factors that limit when a waste point can be visited, and they must be taken into account to provide good service to the citizens.
- **Uncertainty**: When planning waste collection routes, it is impossible to know the exact waste quantity that will be present at each waste container in advance. This leads us to the necessity of making robust plans that can assure that a route will not be abandoned prematurely leaving some neighborhoods without service.

Thus far, and to the best of our knowledge, there has not been any approach to tackle demand uncertainty in VRP that fully exploits the features of fuzzy numbers to model customer demands and make decisions based on the fuzzy computations. Instead, they are typically based on Monte Carlo simulations [6] or use a β-robust approach to assure that solution costs are no greater than a predefined value [7] in stochastic environments, which can be referred just to customer demand or also to displacement time and the mere customer presence.

In [6], the authors use an ant-colony algorithm combined with simulated annealing to solve a VRP with uncertain customer demands. Uncertainty is modeled by means of fuzzy set theory using the notions of possibility, necessity and the average of both metrics, credibility [8]. This approach uses credibility thresholds, and runs 1000 times per credibility value. For each one of them, customer demands are predetermined within an interval, and optimal routes and their costs are computed. The optimal credibility corresponds to the one whose average cost is minimal.

In this work we tackle VRP with time windows and uncertain demands to minimize the total cost. We propose a novel strategy to solve the problem using adaptive credibility thresholds, thus trying to find good solutions for the problem with relaxed capacity constraints and then evolve to find solutions with higher credibility levels. In the process, solutions for all credibility levels are found, obtaining a collection of solutions that can be adopted depending on the desired credibility levels. We propose a memetic algorithm as solving method which combines a genetic algorithm with local search strategies. A new version of the Split decodification algorithm [9] is designed to work with the credibility thresholds. The work is organized as follows: first, Section II formalizes the Vehicle Routing Problem with time windows and uncertain demands. Section III explicitly explains the proposed algorithm for the presented problem, such as individual encoding, evaluation and operators used in a genetic algorithms and the local search that is integrated into it. Then, Section IV discusses the algorithm's performance across several testing scenarios and Section V

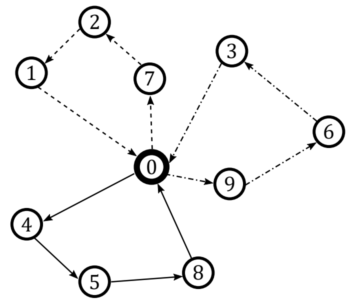

Figure 1. An example of a VRP instance with 9 customers and 3 routes.

yields the conclusions extracted from the whole work and also presents future lines of work.

II. Problem Formulation

The classic VRP consists in finding the best set of routes (typically minimizing costs) that allow the customers' demand coverage. The problem can be modeled as a complete undirected graph $G = (S, A)$ where $S = \{0, 1, 2, \ldots, n\}$ is a set of nodes representing customers (node 0 represents the depot) and $A = \{(i, j) \mid i \neq j, i, j, \in S\}$ is the set of arcs joining all the nodes. Each arc (i, j) is labeled with a non-negative cost that meets the triangle inequality $c_{ij} \leq c_{ik} + c_{kj}, \forall i, j, k \in S$. On the other hand, a vehicle fleet $V = \{1, \ldots m\}$ is stationed at the depot and is tasked to visit the customers.

In some cases, each customer i is also associated with a time window $[e_i, l_i]$ that indicates the time interval where the demanded service can be provided (VRP-TW). This way, e_i is the earliest possible time to do so, and l_i the latest. The depot has a time window as well, which is denoted by $[e_0, l_0]$. Note that every vehicle of the fleet must return to the depot before l_0. In addition to the time windows, each customer has a service time s_i assigned that indicates the amount of time needed to satisfy the customer's demand d_i. All vehicles are assumed to have a speed of 1 unit/s., so the time spent traveling an arc equals its distance ($t_{ij} = c_{ij}$). If a vehicle arrives to a node i before the earliest moment of its time window ($a_i < e_i$) a waiting time is incurred, so the beginning of the service time is $b_i = \max(e_i, a_i)$.

In this work, we consider one of the most common variants of the VRP, the Capacitated VRP (CVRP), where every vehicle has a maximum capacity constraint Q common to them all, and the number of vehicles available in the depot is unlimited. This way, a solution for the CVRP is a partition $\{R_1, R_2, \ldots R_k\}$ where each R_i contains a subset of clients whose demand sum does not exceed Q.

The CVRP-TW problem can be modeled with Equations (1)-(11). First, Equation (1) gives the objective function to minimize, which in this case is the sum of route distances. Equation (2) considers a binary variable x_{ij}^v which indicates whether the arc (i, j) is traversed by the vehicle v. Equations (3) and (4) make sure that a customer is visited and by a single vehicle. Equations (5) and (6) assure that every route starts and ends at the depot. Equation (7) expresses that the demand attended by each vehicle must not exceed the maximum capacity. The remaining four Equations ((8)-(11)) handle time window constraints. The first two are used to keep track of the time taken by each route, including waiting times if necessary, whereas the other two define the depot's time window, and how every vehicle must end their route before that time window expires.

$$\min f(x) = \min \sum_{v \in V} \sum_{i \in S} \sum_{j \in S} c_{ij} x_{ij}^v \tag{1}$$

Subject to:

$$x_{ij}^v \in \{0, 1\}, \forall i, j \in S, v \in V \tag{2}$$

$$\sum_{v \in V} \sum_{j \in S} x_{ij}^v = 1, \forall i \in S - \{0\} \tag{3}$$

$$\sum_{j \in S} x_{ij}^v - \sum_{j \in S} x_{ji}^v = 0, \forall i \in S - \{0\}, \forall v \in V \tag{4}$$

$$\sum_{j \in S - \{0\}} x_{0j}^v = 1, \forall v \in V \tag{5}$$

$$\sum_{j \in S - \{0\}} x_{j0}^v = 1, \forall v \in V \tag{6}$$

$$\sum_{i \in S - \{0\}} \sum_{j \in S} d_i x_{ij}^v \leq Q, \forall v \in V \tag{7}$$

$$x_{ij}^v = 1 \rightarrow a_j = b_i + s_i + t_{ij}, \forall i, j \in S, v \in V \tag{8}$$

$$b_i = \max\{e_i, a_i\}, \forall i \in S - \{0\} \tag{9}$$

$$e_0 \leq b_0^v \leq l_0, \forall v \in V \tag{10}$$

$$e_0 \leq b_{n+1}^v \leq l_0, \forall v \in V \tag{11}$$

A. Uncertain demands and credibility

In the cases where demand is not known in advance, and only an interval of possible values and a most plausible one are available, Triangular Fuzzy Numbers (TFN) can be used to model the uncertainty. A TFN $\hat{a} = (a^1, a^2, a^3)$ is a fuzzy value whose membership function is given in Equation (12).

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x - a^1}{a^2 - a^1} & \text{if } a^1 < x < a^2 \\ \frac{a^3 - x}{a^3 - a^2} & \text{if } a^2 \leq x < a^3 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

In the context of this work, for each demand $\hat{d}_i$, each defining point represents the smallest possible demand value (d_i^1), the most plausible one (d_i^2) and the largest possible value (d_i^3). These values are generally given by experienced workers. We can denote this variant as Capacitated Vehicle Routing Problem with Time Windows and Fuzzy Demands (CVRP-TW-FD).

Uncertain demands make it more difficult to determine if vehicle with capacity Q will necessarily cover all the demand of its route (Equation (7)). In the context of triangular fuzzy numbers and possibility theory, the necessity of meeting the demand is a matter of degree, which is given by Equation (13). Similarly, the degree to which the demand will possibly be met can be computed using Equation (14). The average of those equations (Equation (15)) is called Credibility degree, and it is the criterion adopted for solution evaluations in this approach, based on the work from [6].

$$\text{Nec}(\hat{q}_i \leq Q) = \sup_{x \leq Q} \mu_{\hat{q}_i}(x) = \begin{cases} 0 & Q \leq q_i^1 \\ \frac{Q-q_i^1}{q_i^2-q_i^1} & \text{if } q_i^1 \leq Q \leq q_i^2 \\ 1 & \text{if } q_i^2 < Q \end{cases} \tag{13}$$

$$\text{Pos}(\hat{q}_i \leq Q) = 1 - \sup_{x > Q} \mu_{\hat{q}_i}(x) = \begin{cases} 0 & Q \leq q_i^2 \\ \frac{Q-q_i^2}{q_i^3-q_i^2} & \text{if } q_i^2 < Q \leq q_i^3 \\ 1 & \text{if } Q > q_i^3 \end{cases} \tag{14}$$

$$\text{Cr}(\hat{q}_i \leq Q) = \begin{cases} 0 & Q < q_i^1 \\ \frac{Q-q_i^1}{2(q_i^2-q_i^1)} & \text{if } q_i^1 \leq Q \leq q_i^2 \\ \frac{Q+q_i^3-2q_i^2}{2(q_i^3-q_i^2)} & \text{if } q_i^2 < Q < q_i^3 \\ 1 & Q \geq q_i^3 \end{cases} \tag{15}$$

Consequently, the constraint given by Equation (7) has to be reformulated to only accept routes whose credibility equals or exceeds a threshold ζ. That is, it is replaced with Equation (16).

$$\text{Cr}\left(\sum_{i \in S-\{0\}} \sum_{j \in S} \tilde{d}_i x_{ij}^v \leq Q \right) \geq \zeta, \forall v \in V \tag{16}$$

III. Memetic Algorithm

Genetic algorithms (GA) are classic metaheuristics which are demonstrated to have high performance in many problems. The general schema of a genetic algorithm is as follows. Solutions of the problem are encoded as individuals, which conform a population. At each step, individuals in the population are paired (selection) and combined (crossover) with a certain probability, thus creating an offspring population. Each offspring has a probability of suffering a small mutation. Finally, the offspring population is combined with the previous one to conform the population that will go ahead to the next generation. This process continues until a stopping criterion is met, and the best individual found so far is returned as the best found solution to the problem. Genetic algorithms are popular not just for their performance but also for the ease to complement them with other metaheuristics and optimization algorithms. Memetic algorithms are a good example of that, combining the exploration of an evolutionary strategy with the exploitation capabilities of local search.

A. *Individual encoding and evaluation*

Typically, solutions to the VRP can be encoded using a permutation of all customers (also called “giant tour”). When decoding a permutation to obtain a solution, the Split algorithm implemented in [9] as part of their GA to solve the CVRP, appears as the most successful method. This algorithm decomposes a giant tour taking into account vehicle capacity and other constraints such as time windows. In order to evaluate individuals for the CVRP-TW-FD, the original algorithm must be adapted to consider time windows and uncertain demands with credibility thresholds. The pseudo-code of the Split algorithm adapted to CVRP-TW-FD is given as Algorithm 1.

First, two vectors V and P are defined such that V contains the best accumulated cost of every route considering the first i destinations and P keeps the start of the best route that ends at the element i (lines 1-5). The next phase fills the previous vectors with the needed data for evaluation and route reconstruction (lines 7-30). Lines 8-10 initialize the control variables that model each route that picks the i-th element of the grand tour T as first destination. The following *repeat* block (lines 11-29) adds a new element to be part of the current route (T_i as the first location to visit) and computes the cost, load and inverted time associated with that addition. If the new addition conforms an infeasible route, the *repeat* block is exited, and the algorithm continues with the next route's first destination point $i+1$. Until that happens, vector values V_j and P_j are updated with the previous accumulated costs and the index of the best predecessor (end of previous route), respectively.

Reconstruction of the routes is not hard to perform once the predecessor list is ready, since the only thing to do is to trace what node is the best predecessor to the current one, starting from the end, and add each subroute or $trip$ to the set of routes S in the corresponding order [9].

An example of the impact that the credibility threshold ζ may have on the solution evaluation for a certain problem instance is illustrated in Figure 2. The example instance is detailed in Table I, where each row is a customer (0 is the depot) and (x_i, y_i) represents its coordinates. Note that the larger the threshold value is, the more vehicles are involved in the solution with the consequent increase in reliability and distance cost.

B. *Genetic operators*

In this proposal, we test different crossover and mutation operators ([10]), as well as their respective probability values to find the best setup for our algorithm. Regarding the crossover operators, the following ones are considered:

- Partially Mapped Crossover (**PMX**)
- Ordered Crossover (**OX**)
- Edge Recombination Crossover (**ERX**) [11]

The first two are commonly-used crossover operators for permutation, whereas the last one is specific to Traveling Salesman Problem (TSP) and VRP problems. We also consider three well-known mutation operators:

Algorithm 1: Pseudo code of the Split algorithm for CVRP-TW-FD

Data: Grand tour T, Cost matrix c, Time windows $[e, l]$, Credibility threshold ζ
Result: Best predecessors and costs

$V_0 \leftarrow 0$; `// Accumulated routes cost`
$P_0 \leftarrow 0$; `// Predecessor vector`
for *i from 1 to n* **do**
  $V_i \leftarrow \infty$;
  $P_i \leftarrow -1$;
end
for *i from 1 to n* **do**
  $j \leftarrow i$; `// End of route cursor`
  $\tilde{load} \leftarrow 0$; `// Route load (fuzzy)`
  $z \leftarrow 0$; `// Route time`
  repeat
    $\tilde{load} \leftarrow \tilde{load} + \tilde{d_{T_j}}$;
    if $i = j$ **then**
      $cost \leftarrow c_{0T_i} + s_{T_i} + c_{T_i0}$;
      $z \leftarrow \max(t_{0T_i}, e_{T_i}) + s_{T_i} + t_{T_i0}$;
    else
      $cost \leftarrow cost - c_{T_{j-1}0} + c_{T_{j-1}T_j} + s_{T_j} + c_{T_j0}$;
      $aux \leftarrow \max(z - c_{T_{j-1}0} + c_{T_{j-1}T_j}, e_{T_j})$;
      if $aux \leq l_{T_j}$ **then**
        $z \leftarrow aux + s_{T_j} + t_{T_j0}$;
      else
        $z \leftarrow \infty$;
      end
    end
    if $Cr(\tilde{load} \leq Q) \geq \zeta$ *and* $(V_{i-1} + cost) < V_j$ *and* $z \leq l_0$ **then**
      $V_j \leftarrow V_{i-1} + cost$;
      $P_j \leftarrow i - 1$;
    $j \leftarrow j + 1$;
  until $j > n$ *or* $z > l_0$ *or not* $Cr(load \leq Q) \geq \zeta$;
end

Table I
EXAMPLE INSTANCE DESCRIPTION.
VEHICLE CAPACITY $Q = 50$

i	(x_i, y_i)	$\tilde{d_i}$	e_i	l_i	s_i
0	$(94.53, 110.24)$	-	0	350	-
1	$(33.68, 62.54)$	$(8, 19, 26)$	30	80	10
2	$(65.65, 36.84)$	$(11, 21, 31)$	100	240	10
3	$(135.20, 51.58)$	$(3, 6, 11)$	150	290	10
4	$(31.26, 139.27)$	$(18, 19, 20)$	10	160	10
5	$(65.65, 168.37)$	$(3, 7, 11)$	30	220	10
6	$(184.34, 92.40)$	$(8, 12, 18)$	80	220	10
7	$(93.62, 65.56)$	$(9, 16, 23)$	30	90	10
8	$(118.19, 163.84)$	$(3, 6, 12)$	10	110	10
9	$(134.07, 118.86)$	$(8, 16, 24)$	20	180	10

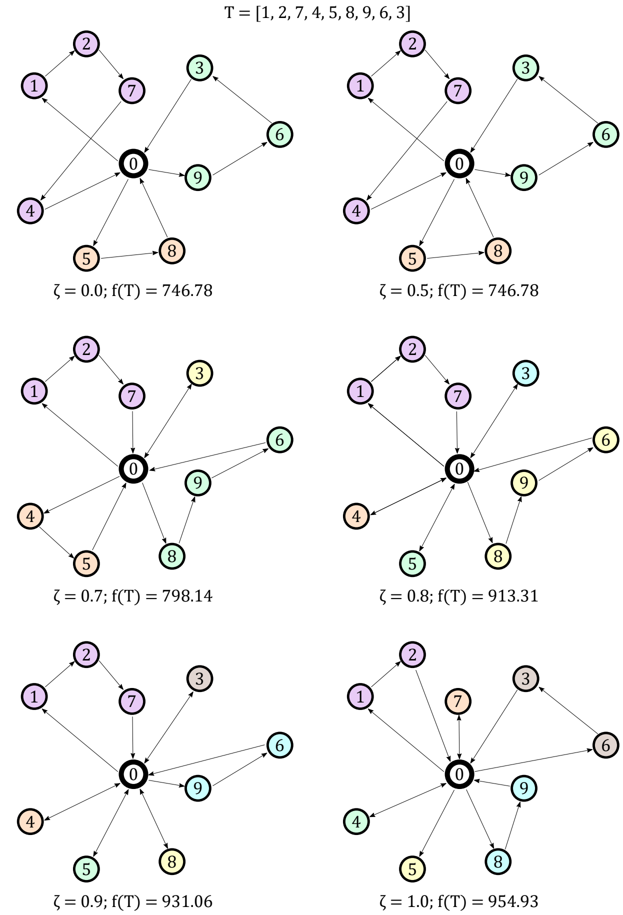

Figure 2. Evaluation of a same grand tour with different credibility thresholds.

- Index Shuffle (**ISM**): It swaps two positions at random.
- Scramble (**SM**): A sub-array in the permutation is rearranged at random.
- Inversion (**IM**): A sub-array in the permutation is inverted.

In this proposal, both selection and replacement operators are predefined to be 3-tournament (3 population members are picked at random and the best one is picked) and elitist (the best member of the current generation survives to the next by replacing the worst one of the new generation), respectively.

C. Local Search

Local search algorithms are a type of trajectory-based metaheuristics. The most common one, hill climbing, has a quite simple structure: given an initial solution or state, generate a set of neighbor solutions and examine them. If a better position in the neighborhood is found, a move is performed to that state and the process is repeated until no neighbor is better than the current state. Several neighborhood structures can be implemented to work with any solution representation and problem. In this case, two of the most common operators for VRP are the ones considered:

- **Swap**: Two vector elements are swapped. Similar to Index Shuffle mutation.

- **2-opt**: Two cut points define a section in the original vector and the solution with that segment inverted define a neighbor.

Notice that for an instance with n customers, the neighborhood size for both neighborhoods would be $\frac{n \times (n-1)}{2}$.

Another factor that should be considered while implementing local search algorithm is the neighborhood exploration or movement policy and, in large scale problems, pruning. The first one refers to the process to select the neighbor solutions and to perform a move. There are two neighborhood exploration strategies:

- **First improvement** (FI): Once a better solution than the starting one is found, the movement is performed. In this case the order that the neighborhood set is explored can have a notable impact. In this work, this strategy randomly explores the generated neighborhood.
- **Best improvement** (BI): Every neighbor solution is evaluated and the one that improves most determines the movement. The order in which the neighborhood is explored is irrelevant here, since all of them are visited.

As to the pruning, it is a process that discards neighbors in order to improve the efficiency of the search. A naive approach is to limit the evaluations per neighborhood by means of a percentage of the neighborhood size or a fixed number.

D. Proposed algorithm

The aim of this work is to provide a method that can produce good quality and robust routes for a given CVRP-TW-FD instance. To achieve this goal, the process listed in Algorithm 2 is proposed.

Algorithm 2: Memetic algorithm with dynamic credibility for the CVRP-TW-FD

Data: Population size p, Credibility threshold step $\Delta\zeta$, Crossover probability p_c, Mutation probability p_m, LS probability p_{LS}, LS neighborhood $neigh\_op$, LS exploration $expl\_str$
Result: Best routes sets found per credibility threshold

$pop \leftarrow$ **initialization**(p);
for ζ *from* 0 *to* 1 *step* $\Delta\zeta$ **do**
  evaluate(pop, ζ);
  repeat
    $parents \leftarrow$ **3-tournament_selection**(pop);
    $offspring \leftarrow$ **crossover**($parents$, p_c);
    $offspring \leftarrow$ **mutation**($offspring$, p_m);
    $pop \leftarrow$ **elitist_replacement**(pop, $offspring$);
    for ind *in* pop **do**
      if $uniform(0,1) \leq p_{LS}$ **then**
        $ind \leftarrow$ **local_search**(ind, $neigh\_op$, $expl\_str$);
    end
  until *15 generations without improvement*;
  record_best(pop, ζ);
end

Essentially, the algorithm follows the classic scheme of a Genetic Algorithm. However, after replacement, there is a small chance p_{LS} for each individual to be improved by Local Search. This aspect is the one that brings exploitation to the algorithm. The process continues until no better solution is found for 15 consecutive generations. At that time, the best solution found is stored and the credibility threshold is increased by $\Delta\zeta$. The population is not reset at this stage, but a complete evaluation is needed, since it cannot be assured that the previous routes fulfill the new credibility requirement. In the end, the algorithm yields the set with the best solutions per credibility threshold. In this work, we propose to set $\Delta\zeta = 0.1$.

IV. Results and analysis

This section focuses on the previously defined algorithm's performance across various well-known CVRP instances conveniently adapted for uncertain demands. In particular, six instances from [12]: *R110*, *R203*, *C107*, *C205*, *RC101* and *RC208* are transformed and tested. Problems of type R have generated customer locations from a squared grid, while type C are clustered and RC implies a combination of both methods. The first digit in each instance name indicates how constraining time windows and vehicle capacity are, being 1 very constraining and 2 hardly constraining. The instances are adapted by transforming every customer's demand d_i into a TFN $\hat{d}_i = (d_i - \Delta d_i, d_i, d_i + \Delta d_i)$, where Δ is a random value between 0 and 0.8 and each obtained point is rounded to the nearest integer. The algorithm has been implemented with the help of the DEAP Python framework [13] and run on an Intel Core i5-9400F processor with 16 GB RAM and Windows 10.

A. Parameter setting

A first set of experiments is carried out to determine the best combination of Local Search (LS) and GA parameters. For LS, it is needed to find the combination of neighborhood operator and exploration strategy that offers the best improvements. For any medium to large sized instance, the neighborhood size can be deemed as too large in the sense that the computational time needed to evaluate all solutions is too high to make the memetic algorithm converge in a reasonable amount of time. Therefore, we propose to limit the number of evaluations per run, applying LS to a small percentage of the population and exploring a limited fraction of the neighborhood which is randomly selected. In this testing, LS is applied to 2% of the population each iteration, including the best solution in the population, and only 10% of all neighbors are explored. A set of 50 random solutions are created, and the four LS variants that result from combining the two exploration strategies and neighborhood structures are run on every one of them.

Table II shows fitness-related values: minimum, maximum, average and standard deviation obtained through the 50 runs. The last two columns count how many times the corresponding variant gives the best and worst solution among the four. On average, 2-opt operator behaves worse than Swap independently on the exploration strategy. Not only the means are greater than their counterpart operator, but the rankings

Table II
LOCAL SEARCH VARIANTS PERFORMANCE SUMMARY ON FUZZY *RC208* (FITNESS)

Variant	Min.	Max	Mean	Std. Dev.	#1	#4
FI - Swap	1342.3	1797.8	1588.4	110.93	11	0
BI - Swap	1326.4	2004.4	1586.7	124.33	16	0
FI - 2-opt	1419.3	2214.4	1681.8	159.37	9	27
BI - 2-opt	1330.2	1961.4	1640.0	149.73	14	23

indicate that, though there are times when they have the best performance, the optimization gets stuck much more often. This, naturally, can be due to the limitations on the neighborhood exploration, but it can be stated that the best operator for our method is Swap. To better assess differences between all four variants, a Friedman statistical test is run, showing no significant differences between FI-Swap and BI-Swap. However, the FI strategy can potentially save execution time, since it does not necessarily evaluate all neighbors. Therefore, this is the strategy that is used.

Concerning the genetic operators, a similar test as with LS has been carried out: 50 different populations with 60 individuals are optimized with all proposed algorithm variants. Population size is selected on the basis of the necessary balance with respect to execution time when LS is present. We consider the 3 crossover and mutation operators introduced in the previous section as well as values 0.4, 0.6 and 0.8 for crossover probability and 0.1 and 0.2 for mutation probability, leading to 54 different variants. In this case, there are two factors to consider: fitness and diversity. Naturally, the combination of operators that is capable of obtaining the best possible solutions should be selected. However, when combined with LS, diversity should be also encouraged in the population to avoid premature convergence which is much more likely when there is potential to run LS in every generation. We consider that two solutions are similar if they have the same route set. Each population is associated a diversity rate, computed as the quotient between the number of distinct individuals in it and the population size.

Regarding fitness, the best 12 variants make use of the ERX crossover, and the best results are obtained with crossover probability of 0.8 and ISM mutation with probability 0.2. Regarding diversity, as expected, the best variants are those that have a high crossover and mutation probabilities, since these operators are in charge of creating new solutions. In fact, the three setups yielding more diverse solutions are those using ISM mutation, the highest combination probabilities and the three crossover operators. There, it seems straight forwards to choose ERX crossover with probability of 0.8 and ISM mutation with probability 0.2, since it obtains both the best results in terms of fitness and it is in the top three in terms of diversity.

B. Algorithm's performance

This subsection presents the results of the CVRP-TW-FD with dynamic credibility when applied to set of instances with uncertain demands. Figure 3 shows population's evolution through the process for instances C107 and R203, being these two representative of the others. Fitness (in red) and number of routes obtained (in blue) of the best solution at each credibility threshold are depicted. Table III shows the average route cost and number of vehicles obtained for each credibility threshold and instance.

Table III
BEST SOLUTIONS FOUND FOR ORIGINAL FUZZY INSTANCES PER CREDIBILITY THRESHOLD
(FITNESS, NUMBER OF ROUTES)

	R110	R203	C107	C205	RC101	RC208
$\zeta = 0.0$	1455.4, 19	1163.1, 10	1672.5, 20	1306.8, 14	2073.9, 25	1121.6, 7
$\zeta = 0.1$	1301.7, 15	1163.1, 10	1311.7, 16	1055.8, 10	2068.9, 25	1118.4, 7
$\zeta = 0.2$	1293.2, 15	1163.1, 10	1311.7, 16	1055.2, 10	2068.9, 25	1118.4, 7
$\zeta = 0.3$	1293.2, 15	1163.1, 10	1311.7, 16	1055.2, 10	2068.9, 25	1118.4, 7
$\zeta = 0.4$	1293.2, 15	1163.1, 10	1311.7, 16	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 0.5$	1293.2, 15	1163.1, 10	1311.7, 16	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 0.6$	1293.2, 15	1163.1, 10	1245.5, 15	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 0.7$	1293.2, 15	1163.1, 10	1245.5, 15	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 0.8$	1293.2, 15	1163.1, 10	1330.4, 16	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 0.9$	1293.2, 15	1163.1, 10	1308.1, 16	1053.7, 10	2068.9, 25	1118.4, 7
$\zeta = 1.0$	1299.3, 15	1163.1, 10	1347.3, 16	1053.7, 10	2068.9, 25	1118.4, 7

At a first glance, it is remarkable that route cost and the number of routes show a strong correlation, having very similar patterns during evolution. The other aspect that stands out is that credibility thresholds have almost no impact in the routes. It is not until $\zeta \geq 0.8$ in some of the most constraining instances (e.g. C107) that route costs increase, usually due to the necessity of adding an extra vehicle to cover the demand. This does not happen in instances of type 2, though that should be expected since the vehicle capacity is high. In fact, in many cases best individual optimization cannot go further than what is achieved when $\zeta \leq 0.1$.

Our hypothesis is that this behavior is due to customers' demands not posing a very tight constraint. It is our belief that time windows are in fact the most restrictive constraint. To test this hypothesis, the algorithm is applied to a set of instances with relaxed time windows, and another set with a more restricted vehicle capacity.

C. Relaxed time windows

The next scenario tested in the present work softens the time window constraint in every instance to study whether the impact of uncertainty is more notable in the final solutions and search process. Each customer's time window is set to be equal to the depot's ($e_i = e_0, l_i = l_0 \ \forall i \in S - \{0\}$). Figure 4 and Table IV represent the information concerning population behavior and best solutions per credibility. For the graphics, instances fuzzC107 and fuzzR203 are selected as the most representative.

As expected, fitness values and the number of vehicles involved in the solutions are lower with respect to the original instances, so the main focus in this scenario is the behavior of the population throughout evolution. Strong correlation in line patterns between fitness and vehicles still apply here, and there are still some instances of type 2 behaving similarly to the original instances (*fuzzR203*, *fuzzRC208*). It must be noted that in these cases the number of routes needed to manage

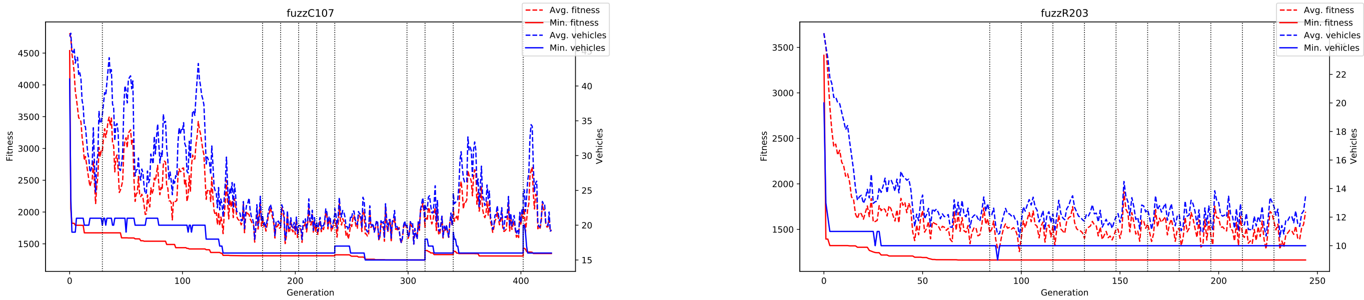

Figure 3. Algorithm evolution for fuzzy instances C107 and R203.

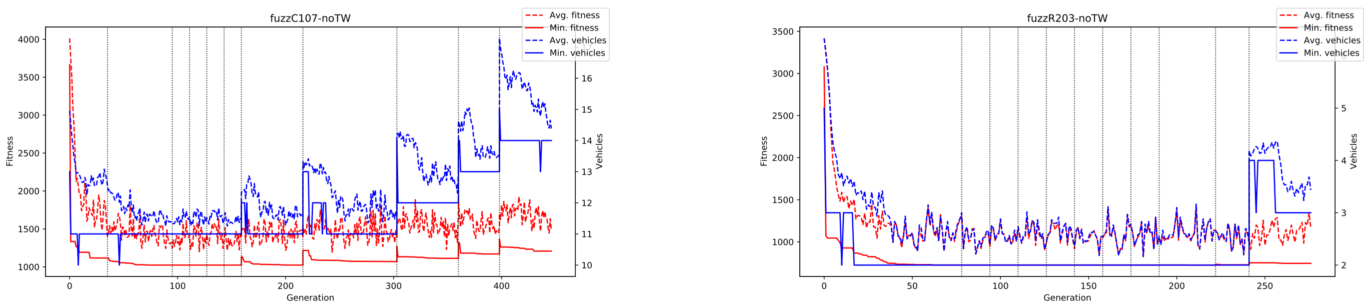

Figure 4. Algorithm evolution for fuzzy relaxed time windows instances.

Table IV
BEST SOLUTIONS FOUND FOR FUZZY INSTANCES WITHOUT TIME WINDOWS PER CREDIBILITY THRESHOLD (FITNESS, NUMBER OF ROUTES)

	R110	R203	C107	C205	RC101	RC208
$\zeta = 0.0$	1057.8, 10	724.91, 2	1116.8, 11	746.72, 3	1125.4, 10	846.17, 3
$\zeta = 0.1$	1017.9, 10	724.91, 2	1023.3, 11	746.72, 3	1124.9, 10	846.17, 3
$\zeta = 0.2$	1014.1, 10	724.91, 2	1023.3, 11	746.72, 3	1124.9, 10	846.17, 3
$\zeta = 0.3$	1014.1, 10	724.91, 2	1023.3, 11	746.72, 3	1124.9, 10	846.17, 3
$\zeta = 0.4$	1014.1, 10	724.91, 2	1023.3, 11	746.72, 3	1124.9, 10	846.17, 3
$\zeta = 0.5$	1014.1, 10	724.91, 2	1023.3, 11	746.72, 3	1124.9, 10	846.17, 3
$\zeta = 0.6$	1014.8, 10	724.91, 2	1023.7, 11	747.57, 4	1119.7, 10	846.17, 3
$\zeta = 0.7$	1030.4, 10	724.91, 2	1069.9, 11	759.92, 4	1185.4, 11	846.17, 3
$\zeta = 0.8$	1026.6, 11	724.29, 2	1111.8, 12	772.38, 4	1216.1, 13	846.17, 3
$\zeta = 0.9$	1039.4, 11	727.42, 2	1169.8, 13	760.02, 4	1313.7, 14	846.17, 3
$\zeta = 1.0$	1049.7, 11	744.57, 3	1208.0, 14	765.60, 4	1387.3, 14	846.17, 3

Table V
BEST SOLUTIONS FOUND FOR FUZZY INSTANCES WITH REDUCED VEHICLE CAPACITY PER CREDIBILITY THRESHOLD (FITNESS, NUMBER OF ROUTES)

	R110	R203	C107	C205	RC101	RC208
$\zeta = 0.0$	1638.4, 24	1100.3, 10	1892.9, 25	1421.2, 15	2583.5, 31	1079.0, 7
$\zeta = 0.1$	1609.1, 24	1100.3, 10	1892.9, 25	1289.7, 14	2483.4, 30	948.44, 7
$\zeta = 0.2$	1550.2, 22	1100.3, 10	1886.4, 25	1174.1, 12	2444.8, 29	942.23, 7
$\zeta = 0.3$	1550.2, 22	1100.3, 10	1886.4, 25	1174.1, 12	2444.8, 29	942.23, 7
$\zeta = 0.4$	1548.9, 22	1100.3, 10	1886.4, 25	1174.1, 12	2444.8, 29	942.23, 7
$\zeta = 0.5$	1548.9, 22	1100.3, 10	1886.4, 25	1174.1, 12	2444.8, 29	942.23, 7
$\zeta = 0.6$	1566.7, 23	1100.3, 10	1886.4, 25	1191.7, 12	2468.7, 30	942.23, 7
$\zeta = 0.7$	1645.1, 23	1100.3, 10	1886.4, 25	1179.5, 12	2529.5, 30	944.44, 7
$\zeta = 0.8$	1688.2, 24	1100.3, 10	2043.9, 28	1183.0, 12	2603.0, 31	951.77, 7
$\zeta = 0.9$	1744.1, 26	1100.3, 10	2266.9, 32	1194.4, 12	2640.2, 33	991.84, 8
$\zeta = 1.0$	1858.5, 28	1102.4, 10	2325.1, 33	1196.5, 12	2776.8, 35	986.51, 8

demand has significantly decreased, hinting to a really high difference between demand quantity and vehicle capacity.

The rest of instances show a more interesting demeanor. The number of routes found by the best solutions are also slightly reduced, though not as much as type 2 instances. Credibility threshold starts to affect the best solution at $\zeta \geq 0.6$, triggering "steps" in both fitness and number of vehicles. What stands out in this stages is that fitness manages to be optimized after the bump by around a 10%, usually at the expense of adding more vehicles.

D. Highly-constrained capacity

To better assess the impact of the credibility thresholds, the last scenario that is tested keeps the original time windows but reduces vehicle capacity to a 40% of its original values. Results of the runs in this situation are shown in Figure 5 and Table V. Instances R110 and R203 are the most illustrative ones in this case.

The first notable effect of the new scenario is the rise of the number of routes, both in average and best individual in instances of type 1, while type 2 ones keep a similar number of vehicles, suggesting that the reduction to a 40% is insufficient to impact the search process. This can be easily seen in Figure 5, where fitness "steps" happen at high values of ζ (if is the case) and are not of great magnitude.

On the other hand, instances of type 1 have a similar behavior to the ones analyzed in the previous subsection: "steps" in fitness happen with the increase of ζ, that come coupled with a significant increase of number of vehicles. Of course, this vehicle growth is much higher than the relaxed

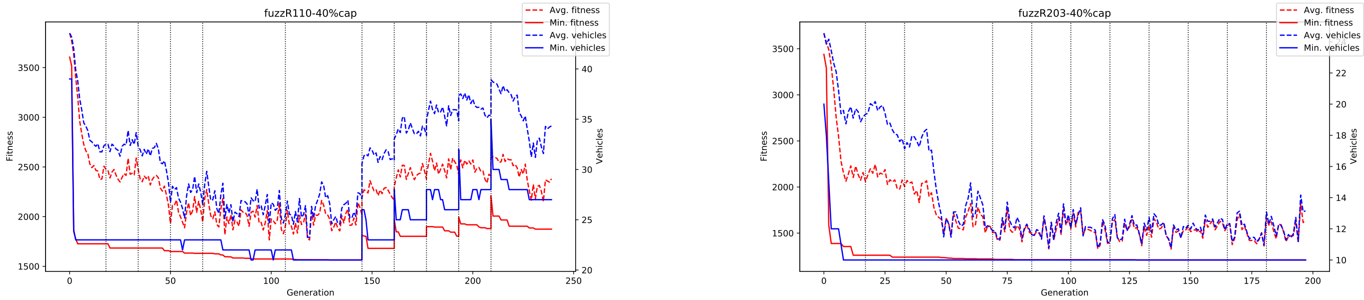

Figure 5. Algorithm evolution for fuzzy highly-constrained vehicle capacity instances.

time windows scenario because of the lower transport capacity. Nevertheless, solutions are well optimized after the bump, minimizing the cost rise both in fitness and number of vehicles.

Based on all these results, one may wonder if it is really necessary start the search from $\zeta = 0.0$ and increase the threshold until reaching 1 or just execute the algorithm at the desired ζ value instead of building solutions step by step. Although good results can be achieved with this strategy it cannot be overlooked that the probability of that happening is really low when the population is initialized with no heuristic and almost completely dependent of the Local Search. One advantage of this approach is that, when credibility requirements strengthen (ζ increases), in the best case population fitness is the same as at the previous threshold (as happens most of the times with the original fuzzified instances), giving a chance of further improve solutions and consider them for previous thresholds. If that happens, a solution found at a higher credibility threshold with lower fitness at least keeps that fitness in the previous ζ, if not improves. In the other case, when solutions change to assure demand coverage, the degree of fitness augmentation may not be big enough to equate to starting from scratch at a new credibility threshold.

V. Conclusions and Future work

The present work has tackled the Capacitated Vehicle Routing Problem with Time Windows and uncertain demands. In this problem, customer demand is not known beforehand, so it becomes necessary to design routes that can optimize costs while making sure that all demand is satisfied and constraints are not violated. Demand has been modeled as triangular fuzzy numbers, and the uncertainty aspect of the problem has been tackled with the notion of credibility to achieve robust solutions across possible scenarios. To solve the problem, a Genetic Algorithm together with a hill-climbing Local Search and dynamic credibility handling has been developed, where solution representation is carried out as a permutation and its evaluation is done by means of the Split algorithm, adapted to uncertainty and time windows. The proposed algorithm is tested against six adapted instances from a well-known literature benchmark. After a parameter study for both LS and GA, its behavior is analyzed across several scenarios according to time window severity and vehicle capacity, pointing to the good performance of the method when it comes to minimize route costs after a demand requirement augmentation. It also shows that when demand becomes a very restrictive constraint, the algorithm is capable of quickly adapting to new thresholds and finding more robust solutions while decreasing the costs.

Acknowledgements

This research has been supported by the Spanish Government under research grant MCI-20-PID2019-106263RB-I00.

References

[1] J. Lenstra and A. Kan, "Complexity of vehicle routing and scheduling problems", *Networks*, vol. 11, pp. 221 – 227, 2006.
[2] A. Ibrahim, R. Abdulaziz, and J. Ishaya, "Capacitated vehicle routing problem", *International Journal of Research - GRANTHAALAYAH*, vol. 7, pp. 310 – 327, 2019.
[3] J. C. Molina, J. L. Salmeron, I. Eguia, and J. Racero, "The heterogeneous vehicle routing problem with time windows and a limited number of resources", *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103745, 2020.
[4] S. Salhi, A. Imran, and N. A. Wassan, "The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation", *Computers & Operations Research*, vol. 52, pp. 315–325, 2014.
[5] C. Lin, "A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources", *Computers & Operations Research*, vol. 38, no. 11, pp. 1596–1609, 2011.
[6] E. B. Tirkolaee, M. Alinaghian, A. A. R. Hosseinabadi, M. B. Sasi, and A. K. Sangaiah, "An improved ant colony optimization for the multi-trip capacitated arc routing problem", *Computers and Electrical Engineering*, vol. 77, pp. 457–470, 2019.
[7] O. Bahri, N. B. Amor, and E. G. Talbi, "Robust routes for the fuzzy multi-objective vehicle routing problem", *IFAC-PapersOnLine*, vol. 49, pp. 769–774, 2016.
[8] A. Nadizadeh and H. Hosseini Nasab, "Solving the dynamic capacitated location-routing problem with fuzzy demands by hybrid heuristic algorithm", *European Journal of Operational Research*, vol. 238, no. 2, pp. 458–470, 2014.
[9] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem. computer & operations research 31(12), 1985-2002", *Computers & Operations Research*, vol. 31, pp. 1985–2002, 2004.
[10] E.-G. Talbi, *Metaheuristics. From Design to Implementation*. Wiley, 2009.
[11] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "A study of permutation crossover operators on the traveling salesman problem", in *Proc. 2nd International Conference on Genetic Algorithms and Their Applications*. Hillsdale, USA: L. Erlbaum Associates Inc., 1987, pp. 224–230.
[12] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints", *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.
[13] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy", *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012.