sustainability

MDPI

Article

An Adapted Version of the Water Wave Optimization Algorithm for the Capacitated Vehicle Routing Problem with Time Windows with Application to a Real Case Using Probe Data

Letizia Tebaldi [1], Teresa Murino [2] and Eleonora Bottani [1,*]

[1] Department of Engineering and Architecture, University of Parma, viale delle Scienze 181/A, 43124 Parma, Italy; letizia.tebaldi@unipr.it

[2] Department of Chemical, Materials and Industrial Production Engineering, University of Naples "Federico II", Piazzale Tecchio 80, 80125 Napoli, Italy; murino@unina.it

* Correspondence: eleonora.bottani@unipr.it; Tel.: +39-0521-905872; Fax: +39-0521-905705

Received: 1 March 2020; Accepted: 22 April 2020; Published: 1 May 2020

Abstract: Customers' habits, as far as shipping requests are concerned, have changed in the last decade, due to the fast spread of e-commerce and business to consumer (B2C) systems, thus generating more and more vehicles on the road, traffic congestion and, consequently, more pollution. Trying to partially solve this problem, the operational research field dedicates part of its research to possible ways to optimize transports in terms of costs, travel times, full loads etc., with the aim of reducing inefficiencies and impacts on profit, planet and people, i.e., the well-known triple bottom line approach to sustainability, also thanks to new technologies able to instantly provide probe data, which can detail information as far as the vehicle behavior. In line with this, an adapted version of the metaheuristic water wave optimization algorithm is here presented and applied to the context of the capacitated vehicle routing problem with time windows. This latter one is a particular case of the vehicle routing problem, whose aim is to define the best route in terms of travel time for visiting a set of customers, given the vehicles capacity and time constraints in which some customers need to be visited. The algorithm is then tested on a real case study of an express courier operating in the South of Italy. A nearest neighbor heuristic is applied, as well, to the same set of data, to test the effectiveness and accuracy of the algorithm. Results show a better performance of the proposed metaheuristic, which could improve the journeys by reducing the travel time by up to 23.64%.

Keywords: metaheuristic algorithm; logistics; water wave optimization; routing; vehicle routing problem; sustainability; transport

1. Introduction

There is no doubt that together with industry 4.0, big data, artificial intelligence, climate change and many others, sustainability is one of the keywords of the 21st century we are living right now. Indeed, concerns about the planet and living beings must be put first, as the future strongly depends on our behavior and actions; according to that, authorities as well as companies and simple consumers are encouraged to act by taking care of this aspect. Researchers and practitioners also play a crucial role, as their fundamental contribution is to develop innovations, metrics, models and tools which can support and stimulate sustainable practices in all the fields in which sustainability can be declined, as well as on the other side understand how current activities are unsustainable [1]. One of the areas which deserves particular attention, since facilities and activities involved have significant impacts, is the transportation activity [2]. In terms of the well-known triple bottom line (TBL) approach, from an

economic perspective, the main effects of transport include, among others, traffic congestion, accident damages, facility and consumer costs, mobility barriers and the depletion of non-renewable resources. This latter can also be seen as an environmental consequence, together with more air and water pollution (e.g., greenhouse gas emissions), habitat loss or hydrologic and noise impact, in general. Finally, as far as the social dimension, it is worth mentioning the effects on human health, traffic safety, mobility disadvantage or community interaction and livability [2,3]. In the light of this list, it is clear that many issues can be addressed when dealing with this topic.

Following this line of reasoning, the aim of this paper is to propose the application of a metaheuristic algorithm (i.e., the water wave optimization algorithm) to a particular transport problem, namely the capacitated vehicle routing problem with time windows (CVRPTW).

The CVRPTW is a particular case of the general vehicle routing problem (VRP), a key problem within the operational research field, which concerns the design of routes for a fleet of vehicles to service a set of customers with known demand subject to side constraints [4]. The pioneers were Dantzig and Ramser (1959) who first proposed a linear programming formulation intended to find the optimum route of a fleet of trucks delivering gasoline to a large number of service stations [5]. Taking into account the possible real applications and the different features and limitations that this problem may assume, many VRP variants were introduced and studied in the literature [6] for more than half a century, and research on this topic is incessantly ongoing [7]. Specifically, the CVRPTW [8], which is under examination in this paper, is one of these variants and represents a generalization of the VPR in which constraints exist on the vehicle capacity and on the specific time-frame (namely the time window), in which each customer needs to be visited [4]. For a complete and exhaustive formulation of the CVRPTW, see [9,10].

If we think to the rapid spread of business to consumer (B2C) e-commerce and to the changes of customers' shipping habits, which generate a significant demand for dedicated deliveries services [11], it is easy to evaluate the current relevance of this problem. Indeed, the CVRPTW can be easily seen as the realistic case of an express courier which has to deliver goods to a set of customers in a day and some customers, normally with a price increase, need to be visited in a specific time span. This kind of problem has of course several implications in terms of the impacts generated during the transport activity, in all the three sustainability dimensions that are above-mentioned.

As far as the solution algorithm is concerned, in literature, there is evidence of various strategies developed to solve the class of VRPs, including the CVRPTW itself; nonetheless, particular attention has been paid to the metaheuristic algorithms. These latter represent a set of stochastic approaches that set off with a randomly generated population, which is subsequently updated by using a succession of different mathematical operations, primarily inspired by some activities of the natural law [12]. Indeed, most of these algorithms are based on analogies with nature, in order to solve difficult and complicated real-world phenomena, and they are able to provide good solutions with small computational effort. Examples of metaheuristic nature-inspired algorithms are tabu search [13], simulated annealing [14] or ant system [15], besides others which are available in the literature. Moreover, compared to heuristic algorithms, by using a certain tradeoff of randomization and local search [16], metaheuristics generally allow one to avoid the risk of falling into a local optimal solution. This is why nowadays there is an increasing attention in applying metaheuristic algorithms to optimization problems [12], also regarded as high-level heuristics [17]. Overall, the reason for the interest in metaheuristic (or heuristic) algorithms is that the VRP and its variants are all NP-hard problems, meaning that the global optimum for the problem cannot be determined under certain conditions as the size of the problem increases [18], as in many real applications. Even if they do not guarantee that the optimum solution can be found, they were demonstrated to be effective and, in most cases, nearly exact.

The water wave optimization (WWO) algorithm used in this paper is a nature-inspired algorithm proposed by Zheng [19]; its original structure has been slightly modified to make it suitable for implementation to solve the CVRPTW problem. The WWO is relatively new, but because of its capability of providing effective solutions to real problems, it has already been applied to different

logistics issues; examples of these contexts include the travelling salesman problem (TSP) [20], the problem of routing a picker in manual warehouses [21], or the scheduling of high-speed trains [19]. To the best of the authors' knowledge, however, there is no evidence in the literature of its application to the CVRPTW and this is the gap intended to fill in this paper.

Clearly, there is no doubt that these innovative solutions can be implemented only thanks to the availability of data which are directly generated from vehicles, such as their speed or their locations, which can generate useful information in terms of travel time.

To test the efficiency and effectiveness of this tool, the adapted WWO was then applied to the case of an express courier operating in Caserta, a city in the South of Italy. Probe data were provided by the company itself, which instantly collect information as far as the localization of its vehicles (origin-destination), through a Global Positioning System. Results from this case are then compared with those obtained by applying a different algorithm to the same set of data, namely the heuristic nearest neighbor [22], considered one of the best known heuristics for solving the CVRP [23]; indeed, previous studies used its results as a benchmark for their assessment [24,25].

In the remainder of the paper, Section 2 provides a background on the WWO, followed by Section 3, where the adapted version of the algorithm is presented; Section 4 shows the case study and its results, which are also compared with those obtained applying the nearest neighbor algorithm. Finally, implications and conclusions are provided (Section 5).

2. Background: The Water Wave Optimization Algorithm

According to Zheng [19], the WWO algorithm is a nature-inspired algorithm that mimics the shallow water wave models. It consists of an initial population of waves randomly generated, each of which defined by two parameters: the wavelength $\lambda \in \mathbb{R}^+$ and the wave height (or amplitude) $h \in \mathbb{Z}^+$. The aim is to explore the seabed area, corresponding to the solution space X, and to identify the "best wave" $x^* \in X$, namely the wave with the highest energy level; this involves determining a fitness value $f(x)$, acting as discriminatory in preferring a certain wave rather than another one. The final aim is to define the best $f(x)$, i.e., the higher fitness value for a maximization problem, the minimum for a minimization one. The lower the distance between the wave and the seabed and the wavelength is, the higher the wave height and the fitness value are. In the general case, a wave can have n dimensions, corresponding to n aspects of the problem under examination. At each iteration of the algorithm, three operations are carried out on the waves to modify them and try to improve the current best solution of the problem; the operations are named propagation, refraction and breaking [26] and recall the natural behavior of waves. The development of the algorithm can be resumed in the following five steps [21]:

Step 1, Initialization. A population of N solutions (i.e., waves) for the problem under examination is randomly generated; each wave has an initial h set at h_{MAX}.
Step 2. The fitness value $f(x)$ is determined and the wave having the best value x^* is identified.
Step 3, Propagation. Propagation is the operation by which the solutions of the problem change at each iteration of the algorithm. It is therefore carried out on all waves of the population; anytime a wave is propagated, it gets a new position, which is obtained by applying the following formula:

$$x'(d) = x(d) + rand(-1,1)\cdot\lambda\cdot L(d) \quad (1)$$

where $x(d)$ represents the original wave, $rand(-1,1)$ is a uniformly distributed random number in the range [−1;1], d ($1 \le d \le n$) is a generic dimension of the problem, $L(d)$ is the length of the d dimension of the search space. The fitness value $f(x')$ of the new solution is calculated and then compared to the corresponding fitness value before the propagation; to choose whether to maintain x or replace it with x', the following set of inequalities is applied (for a maximization problem):

$$f'(x)\begin{cases} > f(x) \Rightarrow & x = x' \text{ and } \quad h' = h_{max} \\ \le f(x) \Rightarrow & x = x \text{ and } \quad h = h - 1 \end{cases} \quad (2)$$

The wavelength is thus updated according to the following formula:

$$\lambda = \lambda \cdot \alpha^{-(f(x)-f_{min}+\varepsilon)/(f_{max}-f_{min}+\varepsilon)} \tag{3}$$

where f_{max} and f_{min} are respectively the maximum and the minimum fitness values among the current population, α is the wavelength reduction coefficient and $\varepsilon > 0$ is a very small positive number, to avoid division-by-zero. The propagation is repeated once for each iteration of the algorithm.

Step 4, Refraction. After some iterations, it is possible that some waves reach $h = 0$. This means that, despite their propagation for h_{max} times, these waves were not able to identify an effective solution for the problem under examination; the logical consequence would be their removal from the set of problem solutions. Refraction is performed exactly on those waves that are expected to disappear, and consists in identifying a new wave, determined as a Gaussian random number with mean μ and standard deviation σ:

$$x'(d) = N\left(\frac{x^*(d) + x(d)}{2}, \frac{\left|x^*(d) - x(d)\right|}{2}\right) \tag{4}$$

where x^* is the best solution found so far; hence, the aim of refraction is to replace the ineffective solution of the problem with a good new one. The height of the new wave is also restored to h_{max}, as previously done in the propagation phase when better solutions were obtained, and the wavelength is updated as follows:

$$\lambda' = \frac{f(x)}{f(x')} \tag{5}$$

Step 5, Breaking. Breaking is a procedure that mimics the real behavior of waves when moving to a position where the water depth is low, i.e., their breaking into a train of solitary waves. This procedure is only applied to those waves that returned the best solutions to the problem in question, i.e., on waves x that became x^* after propagation. By breaking, a local search is performed around this best solution by taking k random dimensions (with $1 \leq k \leq k_{max} < n$, being k_{max} a predefined number) and generating a solitary wave x' for each dimension d, as follows:

$$x'(d) = x(d) + N(0,1) \cdot \beta L(d) \tag{6}$$

where β is the breaking coefficient. If there are no solitary waves better than x^*, x^* is maintained; otherwise it is replaced by the fittest one among the solitary waves.

Steps 2–5 are repeated for each iteration of the algorithm on the whole set of waves in the population. Generally, a maximum number I of iterations is set in advance, and once it has been completely carried out, the algorithm stops.

3. The Adapted Water Wave Optimization to the CVRPTW

The original formulation of the WWO cannot be directly applied to the CVRPTW; hence, in this section, we illustrate its adaptation to make it suitable for solving the problem under examination.

The application of the WWO to the CVRPTW is intended to generate an effective route for a set of M trucks, by minimizing their travel distance and time (namely a minimization problem), according to capacity and availability constraints of the vehicles and time constraints. Vehicles are assumed to start from a depot (indicated as vertex 0) and return to the depot at the end of the trip (indicated as vertex $N + 1$, where N corresponds to the number of customers). Consequently, the objective function corresponds to the sum of the single contributions (i.e., the (i, j) arches in the graph where $i \neq j$) forming the route. Note that each route originates at vertex 0 and ends at vertex $N+1$; on the contrary, no arc terminates in vertex 0, or originates from vertex $N + 1$.

When applying the WWO to this problem, each wave x (denoting the solution of the problem) will correspond to a generic route, consisting of a vector containing at least two zero elements,

which indicate the location of the depot and represent respectively the origin and destination nodes. The remaining $n \in \mathbb{N}^+$ non-zero nodes represent the customers to be visited $(1, 2, \ldots, N)$. The fitness function $f(x)$ will instead denote the total travel time to cover route x. For instance, considering 5 customers to be visited by 2 vehicles (i.e., $N = 5$ and $M = 2$), a potential solution to the problem, reflecting a route configuration, could be the following:

0 2 4 0 5 1 3 0

The phases of the algorithm are below detailed, recalling the previous subdivision in five steps, preceded by Table 1, which illustrates the nomenclature adopted.

Table 1. Nomenclature adopted for the development of the algorithm.

Symbol	Description	Symbol	Description
N	Number of customers (i.e., number of nodes)	$J^{*'}$	Node belonging to Q which minimizes $f(x)$
M	Number of vehicles	d_h	Demand of each customer, non-negative, integer
t_{ij}	Travel time from node i to node j, positive, integer	G	Demand vector
$t_{ij_{C_k}}$	Travel time from node i to node j of the same *k-th* route traveled by one specific vehicle	c	Node set constituting a sub-circuit
I	Number permutations	c_m	Load capacity of each vehicle, non-negative, integer
V(i,j)	Initial matrix (each line represents a vector, i.e., a solution)	J	Nodes belonging to c for which the capacity constraint is not observed
ite	Number of iterations	j	Random number ∈ [1; J] corresponding to the node subjected to shift
K	Number of zeros in each string	d_j	Demand associated to node j
k	Random number $k \in [1; K]$	Q	Set of elements not belonging to c, for which $d_h < d_j$
j^*	Position of the *k-th* zero within the *i-th* string	j'	Nodes belonging to Q
		$[a, b]$	Time window.

Before proceeding, two variables are introduced: a binary variable x_{ij} for each arc (i, j), where $i \neq j$, $i \neq N+1$, $j \neq 0$, which scores 1 in the case that the arc belongs to the optimal solution, 0 otherwise; s_{iC_k} instead is defined for each node i of each *k-th* route and indicates the time in which the vehicle services customer i.

The algorithm aims to determine the following value:

$$\min f(x) = \sum\nolimits_i t_{ij_{C_k}} \quad \forall C_k \tag{7}$$

Subject to:

$$\sum\nolimits_{C_k} \sum\nolimits_j x_{ij} = 1 \qquad \forall\, i \in N \tag{8}$$

$$\sum\nolimits_{h=1}^{N} d_h \leq c_m \quad \forall C_k,\ \forall m \in M \tag{9}$$

$$\sum\nolimits_j x_{0j} = 1 \quad \forall C_k \tag{10}$$

$$\sum\nolimits_i x_{ig} - \sum\nolimits_j x_{gj} = 0 \quad \forall C_k,\ \forall g \in N \tag{11}$$

$$\sum\nolimits_i x_{i,n+1} = 1 \quad \forall C_k \tag{12}$$

$$a_i \leq s_{iC_k} \leq b_i \quad \forall \in N,\ \forall C_k \tag{13}$$

$$x_{ij} = \begin{cases} 1 & if\ arc\ (i,j)\ belongs\ to\ the\ optimal\ solution \\ 0 & otherwise \end{cases} \qquad \forall i, j \in N,\ i \neq j,\ i \neq N+1,\ j \neq 0 \tag{14}$$

Clearly, $f(x)$ corresponds to the objective function, namely the fitness function, and expresses the total time for servicing the customers, computed on the set of optimal routes C_k covered by the trucks. According to what has been said in Section 2, being a minimization problem, the aim in this case is to determine the lowest fitness value.

Constraint (8) states that each customer is visited exactly once, while the next inequality (9) corresponds to the vehicle capacity constraint. The three following equations (10, 11, and 12) impose that each vehicle leaves at vertex 0 (the depot), and after arriving at a customer, the vehicle leaves again, and finally reaches vertex $N+1$ (the depot). Inequality (13) ensures that the time windows are observed, and finally, (14) corresponds to the integrality constraint.

Step 1, Initialization. A population of possible routes is randomly generated. To this end, an initial round trip tour is set, whose corresponding string includes $N-1+2 = N+1$ zeros: two respectively opening and closing the route, and the remaining ones placed between each pair of nodes, since at this stage it is assumed that each customer is directly connected to the depot. Again, for instance, considering the previous example with $N = 5$, a possible configuration (with $N+1 = 6$ zeros) can be as follows:

0 1 0 2 0 3 0 4 0 5 0

Clearly, it would be possible to start with any other random configuration.

In literature, the idea of generating a population starting from a unique configuration is known as seed initialization, and it was noted that both heuristics and metaheuristics implementing this kind of initialization returned better results [27]. After I permutations, whose number is set at the start of the algorithm and defines the problem size, the result is an initial population with the following characteristics:

- the number of solutions corresponds to the number of permutations;
- all solutions have the same size (2*N*+1) and the same value of the objective function, i.e., the same fitness value (the journey time for each route is steady, as each node is always included between two zeros);
- all solutions are possible and realistic.

These solutions form the matrix $V(i, j)$, whose dimensions are (I; 2*N*+1).

Note that after many iterations of (ite) launched, it was observed that the best number of iterations is equal to *N-M*.

Step 2, Propagation. In the description of the original algorithm, at the second step, the fitness value *f(x)* was determined, and the wave having the best value x^* was identified. In the adapted version, however, after initialization, all of the possible solutions own the same fitness value (cf. step 1), meaning that it is not possible to identify the best one. As a result, propagation immediately comes, performed on all the solutions determined in the initialization phase.

From each x solution, a new x' solution is identified, and this last will replace x (if consistent with the constraints, otherwise it will not be considered), even if the objective function value gets worse, namely, it gets a higher value, in this specific case.

For each solution, a $k \in [1; K]$ integer number is randomly generated, where $K = V\ (i,\ j+1:\ j-1)$ is the number of the zeros in each string; *j**, the zero occupying the *k-th* position, is removed. This corresponds to the union of two arches; in other words, two nodes initially belonging to two different sub-circuits are connected, meaning that two customers will be served without any stop at the depot. Of course, this union could generate either a saving or an increase in time depending on the route in question.

The benefit of the seed initialization is that, after the permutations, it is no longer possible to have two identical solutions, even if the k number generated is the same. Indeed, for instance, assuming

$k = 3$ for two different solutions, still having $N = 5$ the two updated solutions could be as follows:

0	1	0	2	4	0	3	0	5	0
0	1	0	2	3	0	4	0	5	0

The value of the objective function is then computed, for each c containing j^*-1 and j^*+1.

Finally, the capacity constraint is checked: given G, demand vector $G = [d_h]$, where $h \in [1,n]$, the following summation is determined: $\sum_{h=j_k,j_{k+1}} d_h$; $\forall c$, $\forall i$ solution and compared to c_m.

After each iteration, K is updated and decreased by one until its value reaches $m - 1$.

Step 3, Refraction. Aiming to improve solutions and making the search be as thorough as possible, refraction operator is performed on those solutions that generated sub-circuits breaking the capacity constraint, i.e., when $\sum_{h=j_k,j_{k+1}} d_h > c_m$, as it is imposed that the vehicle capacity loading is not exceeded. These sub-circuits c are identified, and given that J is the number of their elements, a random value $j \in [1; J]$ is generated, to which d_j demand corresponds. This number reflects the position occupied by the node (i.e., customer) candidate for switching its position with a node belonging to another sub-circuit. The criteria for selecting this second node is that of having a request lower than the one owned by the node that occupies the *j-th* position, in the attempt of respecting the capacity constraint actually violated. Those nodes to which a lower capacity corresponds constitute the Q set ($Q=\left\{j' \notin c \cup d_{j'} < d_j\right\}$), and among them, the one contributing to minimizing the total time involved, i.e., the objective function, is chosen j'^*. These two nodes are then switched. In case the capacity constraint is still not respected, this operation is iterated until Q becomes an empty set; anytime there is the need to repeat the procedure, Q is updated, excluding j'^*.

The value of the objective function and the capacity constraints are evaluated again.

Step 4, Breaking. Net of refraction; if the capacity constraint is still violated, breaking is activated. In the adapted approach, this procedure simply consists in permuting two random nodes: one of the zeros of the string in question (whose position belongs to the range [1; K], where K is appropriately updated according to the current iteration) and one of the non-zero elements of the solution. In mathematical terms, the swap is operated on y_1 and y_2, where $y_1 \in V(i,j)=0$, while $y_2 \in V(i,j)\neq$. The check on the constraint is repeated, and if it is still not respected (i.e., $\sum_{h=j_k,j_{k+1}} d_h > c_m$), the refraction process is repeated. Theoretically, breaking and refraction could alternate n times; however, to avoid excessive computational efforts, the algorithm is set so that this can occur just once. If after one breaking and refraction process, the solution is still not acceptable, there is probably little potential to exploit this solution effectively. Therefore, propagation will be repeated on this solution to get a more effective one. On the contrary, if the solution becomes acceptable, it will be checked for correspondence to the time windows constraint.

Step 5. At this step, finally, the check on the time constraint is activated. For each of the nodes requested to be visited within a time window [a; b], the expected time of the visit is computed; for node j this accounts for $T_j = \sum_i t_{ij}$. In the case $T_j \notin [a_j, b_j]$ (i.e., the visit does not fall into the time span), the maximum value $\max t_{ij}$ within the sub-circuit is determined; then, a new value for the arrival time a'_j is computed as $a'_j = b_j - \max t_{ij}$. A new node j'' is then selected, so that $b_{j''} \leq a'_j$, and inserted instead of j in the sub-circuit.

This procedure is iterated until the time windows are respected, i.e.,$\forall i$, $\forall c \in I$, $\forall j \in c \rightarrow \sum_j t_{ij} \in [a_j, b_j]$, and finally, the best solution, in terms of minimal time, is proclaimed.

Steps 2–5 are repeated for each iteration of the algorithm on the whole set of routes in the population. As in the original formulation of the algorithm, a number *ite* of iterations is allowed and once it has been completely carried out the algorithm stops.

The flowchart depicted in Figure 1 resumes the main steps of the algorithm.

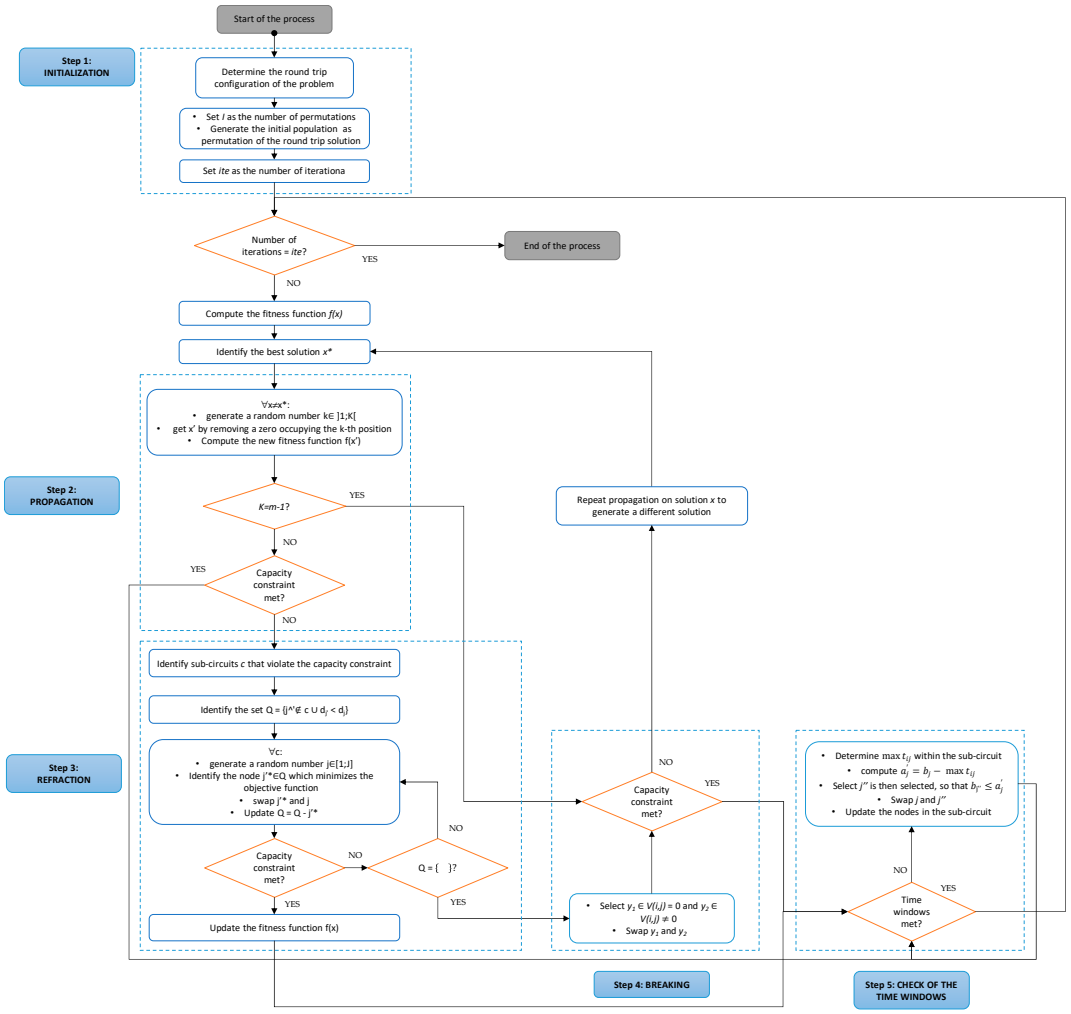

Figure 1. Flowchart of the adapted algorithm.

4. Case Study

In order to test its effectiveness, the adapted version of the WWO to the CVRPTW was applied to a real case study using the software MATLAB®, with real data obtained from an express courier operating in Caserta, South of Italy, and serving customers who ordered products through an e-commerce platform. Related results are discussed in this section.

According to the taxonomy proposed by Braekers et al. [28], the scenario and the physical characteristics of the problem in question are below resumed, respectively in Tables 2 and 3.

Table 2. Scenario characteristics according to [28] for the problem in question.

Scenario Characteristics	
Number of stops on the route	Known (deterministic)
Customer service demand quantity	Deterministic
Onsite services/waiting times	Deterministic
Time windows structure	Strict time windows
Node/Arc covering constraints	Precedence + Subset covering constraints

Table 3. Problem physical characteristics according to [28].

Physical Characteristics	
Transportation network design	Direct network
Location of addresses (customers)	Customers on nodes
Number of points of origin	Single origin
Number of points of loading/unloading facilities (depot)	Multiple depots
Time window type	Restriction on customers + vehicle
Number of vehicles	Limited number of vehicles
Capacity consideration	Capacitated vehicles
Vehicle homogeneity (Capacity)	Similar vehicles
Travel Time	Deterministic
Objective	Travel time + distance + vehicle + lateness dependent

Data about the case study were obtained thanks to interviews with a manager from the company itself, and they reproduce a real working day, specifically Thursday 19th of September 2019. Note that the sequencing followed on that specific day was not provided, as well as the tools used by the company to determine the routes their vehicles have to travel.

That day, exactly 67 customers needed to be visited (this means $N = 67$, reflecting the number of nodes of the graph in question) and two vehicles were available ($M = 2$), both with a load capacity of 300 kg. The depot is unique. As far as the time windows are concerned, 55 customers shall be reached between 10 a.m. and 6 p.m., while the remaining 12 benefit from the "sprinter service" offered by the company and need to be visited within the timeframe 10 a.m. – 1 p.m., i.e., an interval of three hours. This characteristic makes this CVRP a CVRPTW, since the time spans impose constraints on how the routes can be built. The necessary data for the application of the propose approach are the distances between all the possible pairs of nodes (constituting the distance matrix), appropriately converted into travel time, and the weights associated to each delivery. This latter one reflects the capacity constraint, which is expressed in terms of weight of the freight shipped; the volume of the shipment, instead, is not taken into account in the evaluation. The full set of data is too big to be fully reported in the paper, but it could be provided to the interested readers upon request.

Note that no information was provided as far as the goods transported; anyway, considering that in an e-commerce context the quantities ordered can be very low (e.g., a single item such as a book, a garment rather than a small domestic accessory) as well as their weight (for instance, a book may weigh approximately 1-2 kg), the available capacity of the vehicles appears to be reasonable.

The number of permutations I made on the initial round trip configuration was set at 200, which corresponds to the number of solutions in the population; the number of iterations *ite*, instead, was set as $N - M = 65$.

At the end of the last iteration, among the whole set of solutions generated (200), 50 of them resulted in being respectful of all the constraints, and the route returning the best value in terms of objective function, i.e., the minimum travel time, was found to be as follows (Tables 4 and 5, respectively, for the two vehicles):

Table 4. Route traveled by the first vehicle resulting from the adapted water wave optimization (WWO). The values in brackets represent the time (minutes) required to reach the next customer.

Vehicle 1															
0	49	32	22	51	53	10	6	60	37	39	40	65	20	29	24
(9)	(8)	(7)	(9)	(2)	(1)	(9)	(9)	(3)	(1)	(5)	(5)	(3)	(2)	(6)	(4)
21	63	1	25	64	62	3	27	17	16	13	46	26	56	0	
(8)	(4)	(8)	(4)	(4)	(3)	(9)	(1)	(9)	(1)	(3)	(7)	(2)	(6)		

Table 5. Route traveled by the second vehicle resulting from the adapted WWO. The values in brackets represent the time (minutes) required to reach the next customer.

Vehicle 2															
0	30	43	66	61	34	55	45	58	18	2	4	31	15	57	33
(8)	(3)	(4)	(7)	(5)	(5)	(5)	(3)	(3)	(5)	(5)	(7)	(3)	(7)	(10)	(6)
54	35	42	28	48	67	7	36	23	5	8	52	38	50	47	11
(11)	(3)	(9)	(11)	(3)	(5)	(7)	(1)	(4)	(7)	(5)	(3)	(10)	(5)	(5)	(5)
59	12	14	9	19	41	44	0								
(6)	(4)	(7)	(3)	(8)	(9)	(9)									

Note that each customer was assigned to a node, in sequential order; zeros corresponds to the single depot.

According to what has been said and to what is shown in the previous tables, vehicle 1 will leave the depot at around 9:51 a.m. so that, after nine minutes at 10 a.m., the delivery guy rings the doorbell of customer 49; once the delivery is over, he will move towards customer 32, and so on. The last customer to be visited before finishing the route and returning is customer 56. The same reasoning can be repeated for vehicle 2, which will leave the depot one minute later (since, in this case, 8 min are required to reach the first client), bound for customer 30.

The first route, including the 29 customers to be visited, returns a travel time of 152 min, i.e., 2.53 h, while the second route, including 38 customers, is more onerous, and requires 226 min, i.e., 3.76 h. According to that, the critical route is the second, but despite this, it is extinguished in a time which is definitely below the normal business hours. Moreover, the time windows turn out to be all respected; for instance, node 59 represents a client who benefits from the sprinter service, and he is visited at the 180th minute, within the required timespan.

The overall computational time recorded was 22 s.

To evaluate the accuracy of the algorithm proposed here, the heuristic nearest neighbor was applied to the same set of data as well.

The two routes resulting which are, of course, different from those returned by the adapted WWO, are detailed below (Tables 6 and 7 again divided according to the two vehicles).

Table 6. Route traveled by the first vehicle resulting from the nearest neighbor. The values in brackets represent the time (minutes) required to reach the next customer.

Vehicle 1															
0	12	15	19	23	24	36	2	9	44	32	65	57	61	58	59
(3)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(9)	(10)	(9)	(9)	(9)	(5)	(5)	(6)
64	60	55	56	62	67	66	50	63	42	34	46	39	47	48	51
(7)	(6)	(10)	(8)	(5)	(7)	(6)	(6)	(3)	(4)	(5)	(6)	(3)	(4)	(3)	(1)
52	16	20	0												
(1)	(1)	(7)													

Table 7. Route traveled by the second vehicle resulting from the nearest neighbor. The values in brackets represent the time (minutes) required to reach the next customer.

Vehicle 2															
0	11	4	22	28	21	25	17	10	5	7	6	8	27	29	40
(10)	(12)	(8)	(5)	(4)	(10)	(11)	(12)	(12)	(9)	(11)	(12)	(13)	(12)	(10)	(9)
13	43	31	30	38	26	53	14	37	1	41	49	33	54	35	18
(9)	(5)	(10)	(10)	(11)	(12)	(11)	(8)	(10)	(11)	(9)	(10)	(6)	(5)	(13)	(13)
3	45	0													
(6)	(10)														

The two objective values for the first and the second route, respectively, are 166 (2.76 h) and 329 min (5.48 h), serving 34 and 33 different customers. The critical path is again the second, which however, is quite different from that returned by the adapted WWO, and in particular, it is 103 min

longer. Specifically, as far as vehicle 1, the nearest neighbor made the solution 9.2% worse in terms of extra time; same reasoning goes for vehicle 2, which should have to travel 45.6% longer in time.

Moreover, if we compute the total time from the two routes as returned by the algorithms tested, we obtain 378 min (6.3 h) in the case of WWO and 495 min (8.25 h) with the nearest neighbor heuristic, meaning that the WWO could improve the solution by reducing the overall travel time by up to 23.64%. This latter value in minutes, however, would be outside the normal work shift of one day (8 h) and therefore, the resulting solution would not be suitable for application in practice. Overall, the proposed algorithm is able to identify a better solution in terms of efficiency, which is also suitable for direct implementation.

Results were clearly provided to the express courier as well, and the company's managers positively assessed the tool according to their practical experience.

5. Conclusions

In the present manuscript, an adapted version of a metaheuristic algorithm, namely the water wave optimization is presented, applied to the vehicle routing problem with time windows and capacity constraints. The aim of the application is to optimize the route travelled by a fleet of transport means for reducing unnecessary road travelled and thus reducing impactful emissions and inefficiencies. The algorithm was implemented through the software MATLAB®, and it was tested on a working day of an express courier operating in the South of Italy, in the province of Caserta. Results were satisfactory, both in terms of computational time (22 s) and of the solution identified; indeed, by comparing the routes found by the adapted WWO to those obtained by applying the heuristic nearest neighbor, the first algorithm has demonstrated better performance and outcomes, being able to improve the solution found by the heuristic algorithm by 23.64%, in terms of overall time spent by both the vehicles.

Moreover, the proposed approach could be easily adapted to problems different from those modelled in this paper. For instance, it could be used:

- for minimizing cost instead of travelling time;
- for modelling and solving a simple CVRP without time windows, by removing step 5 (i.e., the check on time constraints) from the algorithm structure.

Note that the time values used as input data do not take into account any inconvenience, traffic condition or real situations which can compromise the possibility of being respected; this could be particularly the case for big cities, where traffic or congestion can affect the real travel time of vehicles. Although neglecting these inconveniences is what is typically done in similar studies, this is for sure also a limitation of the current formulation of the problem. Moreover, the capacity constraint was modelled and assessed according to the weights of the items transported, without taking into account their volumes. In the case under examination, due to the limited size of the goods, it is reasonable to think that using the item's weight instead of the volume could be acceptable; however, in other contexts, the lack of one of these aspects could represent a stringent limitation. This point will be taken into account in future applications.

Theoretically, anyway, the outcomes are brilliant.

With regard to the future, our ongoing research activity is intended to improve the programming language, to avoid redundancies and streamline the procedure, for reducing the average number of iterations and further improve the computational time. Higher programming languages and/or software applications can also be involved as well.

For sure, we mean to carry out a set of experiments on known benchmark sets (e.g., see [8] or [26]) and compare their results; a further possible research activity could also address the comparison of the results and the effectiveness of the algorithm with other metaheuristics (e.g., genetic algorithms), whose original formulation would probably need to be adapted for the resolution of the VRP problem with time windows, in case appropriate adaptations do not exist.

Finally, other case studies are highly recommended to further validate the adapted version of the WWO to the CVRPTW.

Author Contributions: Conceptualization, L.T., E.B. and T.M.; methodology, E.B. and T.M.; validation, T.M.; data curation, T.M.; writing-original draft preparation, L.T. and T.M.; writing-review and editing, E.B.; visualization, E.B.; supervision, E.B. All authors have read and agree to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Singh, R.K.; Murty, H.R.; Gupta, S.K.; Dikshit, A.K. An overview of sustainability assessment methodologies. *Ecol. Indic.* **2012**, *15*, 281–299. [CrossRef]
2. Litman, T.; Burwell, D. Issues in sustainable transportation. *Int. J. Glob. Environ. Issues* **2006**, *6*, 331–347. [CrossRef]
3. Banister, D.; Anderton, K.; Bonilla, D.; Givoni, M.; Schwanen, T. Transportation and the. Environment. *Annu. Rev. Environ. Resour.* **2011**, *36*, 247–270. [CrossRef]
4. Baldacci, R.; Mingozzi, A.; Roberti, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *Eur. J. Oper. Res.* **2012**, *218*, 1–6. [CrossRef]
5. Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
6. Abbatecola, L.; Fanti, M.P.; Ukovich, W. A Review of New Approaches for Dynamic Vehicle Routing Problem. In Proceedings of the 2016 IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, USA, 21–25 August 2016.
7. Drexl, M. Rich Vehicle Routing in Theory and Practice. In *Fraunhofer Centre for Applied Research on Supply Chain Services*; Technical Report LM-2011-04; Mainz, Johannes Gutenberg University and SCS: Nuremberg, Germany, 2011.
8. Solomon, M.M. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Oper. Res.* **1987**, *35*, 234–265. [CrossRef]
9. Kallehauge, B. Formulations and exact algorithms for the vehicle routing problem with time windows. *Comput. Oper. Res.* **2008**, *35*, 2307–2330. [CrossRef]
10. El-Sherbeny, N.A. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *J. King Saud Univ.* **2010**, *22*, 123–131. [CrossRef]
11. Morganti, E.; Seidel, S.; Blanquart, C.; Dablanc, L.; Lenz, B. The impact of e-commerce on final deliveries: Alternative parcel delivery services in France and Germany. *Transp. Res. Procedia* **2014**, *4*, 178–190. [CrossRef]
12. dos Santos Coelho, L.; Maidl, G.; Pierezan, J.; Mariani, V.C.; da Luz MV, F.; Leite, J.V. Ant Lion Approach Based on Lozi Map for Multiobjective Transformer Design Optimization. In Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Amalfi, Italy, 20–22 June 2018.
13. Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [CrossRef]
14. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
15. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41. [CrossRef] [PubMed]
16. Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*; University of Cambridge, UNIVER PRESS: Cambridge, UK, 2010.
17. Arockia Panimalar, S. Nature Inspired Metaheuristic Algorithms. *Int. Res. J. Eng. Technol.* **2017**, *4*, 306–309.
18. Kozan, E. and Preston, P. Genetic algorithms to schedule container transfers at multimodal terminals. *Int. Trans. Oper. Res.* **1999**, *6*, 311–329. [CrossRef]
19. Zheng, Y.-J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [CrossRef]
20. Wu, X.-B.; Liao, J.; Wang, Z.-C. Water Wave Optimization for the Traveling Salesman Problem. In *Intelligent Computing Theories and Methodologies*; Huang, D.S., Bevilacqua, V., Premaratne, P., Eds.; ICIC: Mumbai, India, 2015; Volume 9225.

21. Bottani, E.; Rinaldi, M.; Montanari, R.; Murino, T.; Centobelli, P. An adapted water wave optimization for routing order pickers in manual warehouses. In Proceedings of the XXI Summer School "Francesco Turco"—Industrial Systems Engineering, Palermo, Italy, 13–15 September 2016.
22. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
23. Layeb, A.; Ammi, M. A GRASP algorithm based on new randomized heuristic for vehicle routing problem. *J. Comput. Inf. Technol.* **2013**, *1*, 35–46. [CrossRef]
24. Li, Y.H.; Mao, H.J.; Qin, Y.M. Vehicle Routing Problem with Multiple Time Windows and Batch Splitting Based on Inferior First Bidirectional Search Algorithm. *J. Phys.:Conf. Ser.* **2019**, *1314*, 012116. [CrossRef]
25. Liu, J.; Liu, D.; Liu, M.; He, Y. An improved multiple ant colony system for the collection vehicle routing problems with intermediate facilities. In Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010.
26. Soltanian, A.; Derakhshan, F.; Soleimanpour-Moghadam, M. MWWO: Modified Water Wave Optimization. In Proceedings of the 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC2018), Bam, Iran, 6–8 March 2018.
27. Sathyanarayanan, S.; Suresh, J.; Jayakumar, S.K.V. A hybrid population seeding technique based genetic algorithm for stochastic multiple depot vehicle routing problem. In Proceedings of the International Conference of Computing and Communications Technologies, Liverpool, UK, 26–28 October 2015; pp. 119–127.
28. Braekers, K.; Ramaekers, K.; Van Nieuwenhuyse, I. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [CrossRef]