Journal of Quality Engineering and Production Optimization

Vol. 4, No. 2, Summer & Autumn 2019

http://jqepo.shahed.ac.ir

DOI: 10.22070/JQEPO.2020.4978.1119

An Improved Hybrid Cuckoo Search Algorithm for Vehicle Routing Problem with Time Windows

Amin Rezaeipanah [1*], Gholamreza Ahmadi [2], Mahdi Hajiani [3], Mohammad Reza Darzi [3]

[1] *Department of Computer Engineering, University of Rahjuyan Danesh Borazjan, Bushehr, Iran*

[2] *Department of Computer Engineering, Persian Gulf University, Bushehr, Iran*

[3] *Technical and Vocational University, College of Bushehr, Bushehr, Iran*

**** Corresponding author:*** *Amin Rezaeipanah (Email: amin.rezaeipanah@gmail.com)*

Abstract *– Transportation in economic systems such as services, production and distribution enjoys a special and important position and provides a significant portion of the country's gross domestic product. Improvements in transportation system mean improvements in the traveling routes and the elimination of unnecessary distances in any system. The Vehicle Routing Problem (VRP) is one of the practical concepts in the field of investigation and many attempts have been made by researchers in this area. Due to the importance of transportation issues in the real world and the status of these issues in the types of existing systems. In this paper, we investigate the Vehicle Routing Problem with Time Window (VRPTW) and provide a solution for it. The problem of routing vehicles with a time window is an extension of the problem of routing vehicles with limited capacity (CVRP) in which servicing must be done in a specific time window. The purpose of this problem is to optimize the route for each vehicle so as to minimize the total cost of the route and the number of vehicles used, and ultimately maximize customer satisfaction. In the paper, a hybrid method based on cuckoo search and greedy algorithm is proposed to solve the problem of VRPTW. For the cost function, different criteria have been used that are within the framework of the VRPTW problem within hard and soft constraints. In order to evaluate the proposed method, the dataset is used in different sizes. The proposed method is significantly higher compared to similar methods.*

Keywords *– Vehicle Routing, Time Window, Cuckoo Search, Greedy Algorithm, Solomon Dataset.*

I. INTRODUCTION

Transportation in economic systems such as services, production and distribution enjoys a special and important position and provides a significant portion of the country’s gross domestic product. Improvements in transportation system mean improvements in the trodden routes and the elimination of unnecessary distances in any system (Xia & Fu, 2019). In order to ensure high-quality and on-time delivery in logistic distribution processes, it is necessary to efficiently manage the delivery fleet. Nowadays, due to the new policies and regulations related to greenhouse gas emission in the transport sector, logistic companies are paying higher penalties for each emission gram of CO_2/km (Reed et al., 2014). Over the past 20 years vehicle routing problem with time windows has been an area of research

Manuscript Received:30-Jane-2019 & Revised: 23-Aug-2019 & Accepted: 26-Dec-2019
ISSN: 2423-3781

that has attracted many researchers. In this period, a number of papers have been published on the exact, heuristics and meta-heuristic methods of the routing problem with time windows (Esmaeili & Sahraeian, 2019; Lu & Gzara, 2019).

Vehicle routing problem is a set of problems in which a fleet of multiple vehicles of one or more warehouses offers service to customers located in different geographic locations in a way that the costs of doing are minimized. During these routes, customers are only met once and all their demands are received only by a vehicle, each vehicle has a certain capacity, and on the other hand, all routes are starting from a specified point (source of loading), and when the vehicle meets a hierarchy of customers it returns to the same starting point (Reed et al., 2014). This class of problems is at the core of several problems that involve routing and/or scheduling, and occurs in several application domains including, but not limited to, transportation, logistics and communications. Specific examples include airline scheduling, vehicle routing, service network design, load distribution, production planning, computer scheduling, portfolio selection and apportionment (George & Binu, 2018 and Baranwal et al., 2016). For example, the collection of solid waste, fuel distribution, school bus routing, collection of mail from postal boxes or coins from public telephones, rounds of inspection for the preventive maintenance of machinery, or routing of workers movement in warehouses to collect client orders from different areas of the warehouse.

The vehicle routing problem (VRP) is an NP-hard optimization problem that aims to determine a set of least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints (Goel & Maini, 2018). The problem was first defined by Dantzig and Ramser (1959), as the Truck Dispatching Problem (Alzaqebah et al., 2016). VRP is a generalization of the well-known traveling salesman problem (TSP), which aims to design one least-cost route to visit all the customers. The novelty to achieve economic efficiency is exceptionally high in this competitive industry. A distribution firm’s main objective is to make profit, while from a customer’s view the major factor in selecting a carrier is the cost. Distribution is an important domain in our daily life, as it supports most social and economic activities. Improving operational efficiencies in distribution is receiving greater attention as fuel costs are continually increasing. A small reduction in the traveled distance of a daily logistical operation directly translates to cost reduction and decreased environmental impacts. The problem has applications in several real-life optimization problems, which has led to the definition of many problem variants over the years: limited vehicle load capacity (capacitated VRP, CVRP), customer time windows (VRP with time windows, VRPTW), multiple depots (Multi-Depot VRP, MDVRP), pickup and delivery (VRP with pickup and delivery, VRPPD), time-dependent travel time (time-dependent VRP, TD-VRP), heterogeneous fleet (mixed fleet VRP, MFVRP), etc. (Hiermann et al., 2016; Miranda & Conceição, 2016; Kar & Sanyal, 2020; Alvarenga et al., 2007; Kargari Esfand Abad et al., 2019).

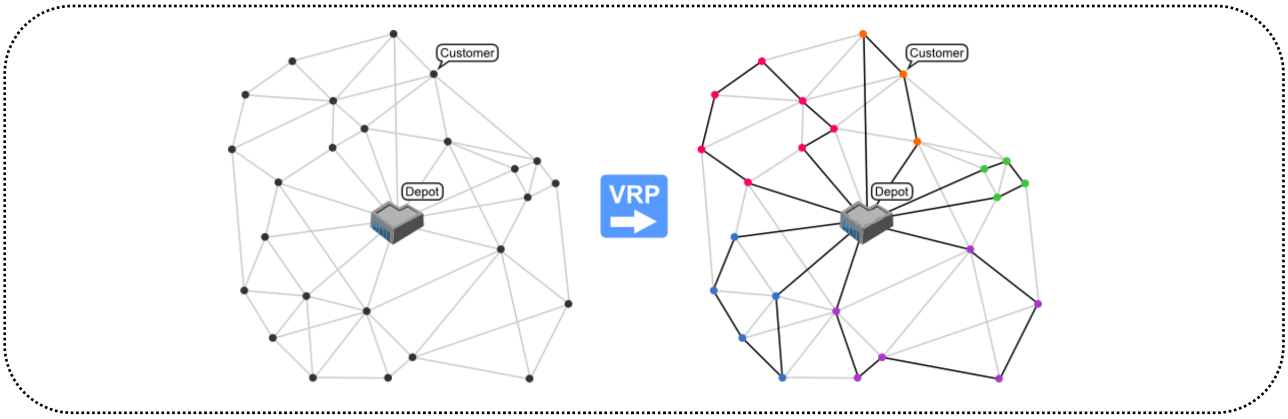


Fig. 1. The Vehicle Routing Problem

The objective of the vehicle routing problem is to deliver a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. In Fig. (1), we can see a picture of a typical input for a VRP problem and one of its possible outputs. For such problems, it is often desirable to obtain approximate solutions, so they can be found fast enough and are sufficiently accurate for the purpose. Usually, this task is accomplished by using various heuristic methods, which rely on some insight into the problem nature.

The cuckoo search algorithm is applied to a variety of combinatorial optimization problems. These include structural optimization problems which are highly nonlinear and involve large numbers of design variables with complex constraints. This algorithm is a novel meta-heuristic based on the reproduction behavior of some cuckoo species in combination with the Levy flight behavior of some birds and fruit flies. It has been applied to a wide range of optimization problems with good performance such as a TSP and VRP. In this paper, we investigate the Vehicle Routing Problem with Time Window (VRPTW) and provide a solution to it. We present a hybrid method based on cuckoo search and greedy algorithm to solve the problem of VRPTW. In general, the main importance of the proposed method is to improve the solutions by a greedy algorithm as well as to provide an objective function with two different types of costs (soft and hard).

The rest of the paper is organized as follows. Related work is reviewed in Section II, and Section III describes the mathematical model and parameters of VRPTW. The basic concept of proposed method is presented in Section IV, and the experimental results are presented in Section V. Finally, Section VI concludes and suggests possible future work.

II. RELATED WORKS

The problem of vehicle routing was first introduced by Dantzig and Ramser (1959). This problem, which is a combination of two round-time vendor problem(s), is trying to optimize a set of routes for the transport fleet to serve a certain number of customers and have different side limits. The Dantzig and Ramser investigated the issue in connection with sending vehicles carrying fuel from the source to a large number of fuel stations (Nazif & Lee, 2010). With increasing the number of stations, the number of possible routes for vehicles movement is significantly increased. As a result, the space is expanded and the optimal solution is found to be very difficult. Dantzig and Ramser, the algorithmic view based on linear formulation of the correct number to find a close answer to the optimal solution is presented. After that, Clarke and Wright (1964) developed the perspective presented by Dantzig and Ramser with their innovative view.

Eshtehadi et al. (2017) provided a suitable solution to the problem of uncertainty pollution routing with demand and travel time. In this study, three innovations are proposed. 1- Uncertain demand that has not been addressed in green vehicle routing issues. 2- Indefinite travel time affected by the speed of movement. 3- Applying three robust approaches to dealing with uncertainty. In addition, a cynical optimistic robust optimization approach has been used. In this method, two main functions of the vehicle routing problem are considered. The first objective function seeks to minimize the overall distance and the second objective function seeks to minimize load weight and distance. Abdallah et al. (2017) proposed a periodic refinement solution for dynamic vehicle routing problems. In this study, an advanced genetic algorithm was used. The proposed algorithm outperforms previously published algorithms in both time-based and weighted fitness evaluation. In this research, the developed genetic algorithm is based on the Hanshar and Omboki-Berman genetic algorithm. However, there are improvements to divert it further and to avoid local optimization to achieve better results.

Hiassat et al. (2017) proposed the problem of locating inventory routing for perishable items using the Genetic Algorithm approach. The distinct structure of the problem requires the development of a new way of displaying chromosomes and an innovative method for local search. The first part of the objective function shows the costs of establishing and operating costs in existing warehouses, while the second part shows the shipping and inventory costs. Vincent et al. (2017) developed the problem of hybrid vehicle routing. This study investigates the problem of hybrid vehicle routing, which is an extension of the problem of green vehicle routing. In this research, simulation of refrigeration with a restart strategy is used to solve this problem and includes two versions. The initial version determines the probability of accepting the worst-case response using the Boltzmann function. The second version determines the probability of accepting the worst-case response using the Cauchy function. The simulated refrigeration algorithm for hybrid vehicle routing problems has a distinct and distinctive representation.

Nasri et al. (2020) proposed a new approach for the robust problem based on the implementation of an adaptive large neighborhood search algorithm and the use of efficient mechanisms to derive the best robust solution that responds to all uncertainties with reduced running times. Goel and Bansal (2020) reviewed papers in the category of RVPRs and Real-life VRPs that are based on hybrid algorithms. In the recent past the interest of applying hybrid optimization algorithms for solving RVRP has grown rapidly. Rich Vehicle Routing Problem (RVRP) is a realistic variant of VRP that incorporates multiple constraints for tackling real-life scenarios. Zulvia et al. (2020) proposed a Green Vehicle Routing Problem (GVRP) for perishable products which optimized the operational cost, deterioration cost, carbon emissions and customer satisfaction. The proposed GVRP model also considers time windows, different travelling time during the peak hour and off-peak hour, and working hours. They use a Multi-Objective Gradient Evolution (MOGE) algorithm to solve the problem. Saxena et al. (2020) suggested an optimized OpenMP-based genetic algorithm as a solution for vehicle routing problem. A contrast has been shown between the serial and parallel implementation of the solution using OpenMP multi-processing architecture which shows a considerable speedup for the execution time of the algorithm to search the best path. For a varied degree of graph structures, this implementation has highly reduced execution time.

Table I provides a summary of some papers covered. The first column references the paper and authors. The second column indicates the main type of VRP problems. The third column describes the main idea of the algorithm. The fourth column lists the main objectives considered in the paper, and the final column indicates the solution approach.

TABLE I. SUMMARY OF SOME PAPERS DISCUSSED

Authors	Type of VRP	Idea	Objective	Solution approach
Eshtehadi et al., (2017)	IPRP	Indefinite travel time affected by the speed of movement	load weight and overall distance	Cynical optimistic robust optimization
Abdallah et al., (2017)	DVRP	Hanshar and Omboki-Berman	Customers service and deviation of routes	Advanced genetic algorithm
Hiassat et al., (2017)	Locating-Inventory-Routing	Two-stage cost function	The shipping and inventory costs	Genetic Algorithm
Vincent et al., (2017)	Hybrid Vehicle Routing	Restart strategy	Schedule disruption trip cancellation costs	Simulated refrigeration algorithm
Afshar-Nadjafi and Afshar-Nadjafi, (2017)	Multi-Depot VRPTW	Time-windows and heterogeneous fleet	Minimize the total heterogeneous fleet cost	Mixed integer programming model
Brandão, (2018)	Open VRPTW	Innovative and effective use of ejections chains and elite solutions	Minimize total travel time	Iterated search algorithm
Marinakis et al., (2019)	VRPTW	Adaptive strategy in PSO	Calculates all the parameters during the optimization process	Multi-Adaptive Particle Swarm Optimization
Zhang et al., (2019)	VRPFlexTW	Pareto optimality for multi-objective optimization	Reduce distribution costs and maximize customer satisfaction	Ant colony optimization and three mutation operators
Nasri et al., (2020)	VRPTW	Two-stage main algorithm	Distance and time windows deviation	adaptive large neighborhood search algorithm
Goel and Bansal, (2020)	RVPRs	Adaptive memory algorithm	Cost of deviation	Hybrid optimization algorithm
Zulvia et al., (2020)	GVRP	Different travelling time during peak hour	Operational cost and customer satisfaction	Many-objective gradient evolution
Saxena et al., (2020)	General VRP	Serial and parallel implementation	Reduce execution time	OpenMP-based genetic algorithm
Proposed Method	VRPTW	Cost function with two types of hard and soft constraints/ Improved cuckoos with greedy algorithm	Reduce travel costs and number of vehicles	Combined cuckoo search and greedy algorithm

III. VRPTW MATHEMATICAL MODEL

Vehicle routing problem, under capacity constraints, is an example of vehicle routing problems on optimal routing of vehicles with the capacity given to servicing a set of customers with a given request, which is known as the Capacitated Vehicle Routing Problem (CVRP) (Yang & Deb, 2009). The difference between CVRP and VRPTW is the addition of time constraints $[e_i, l_i]$ in the case of the problem (Dantzig & Ramser, 1959). Where it is the first time for the client e_i to start operation and l_i the last allowed time to start operation. Starting a vehicle prior to the time limit l_i will result in the addition of the vehicle waiting time on the route. As a result, starting a vehicle after the time limit has resulted in the creation of non-plausible solution and unacceptable solution. In this case, the time limit $[e_i, l_i]$ of a window is a time window for the customer which in the soft mode adds delay time to the cost function and in the hard case, it is justified. The concept of time window is shown in Fig. (2).

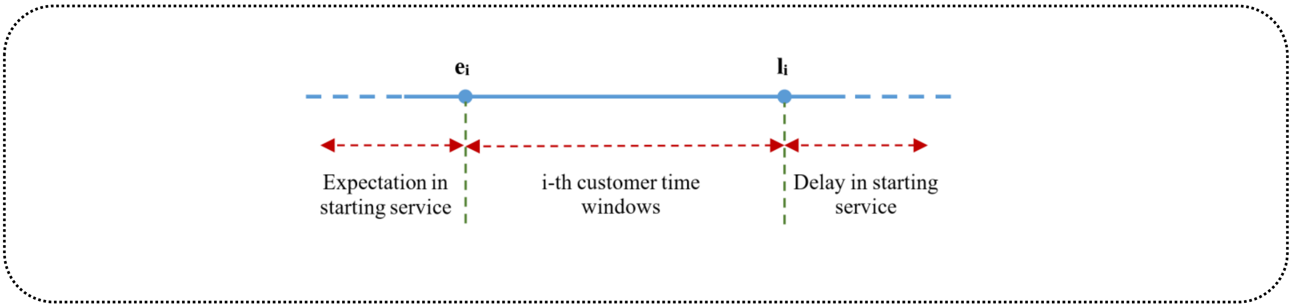


Fig. 2. Concept of time window for customer

The VRPTW is defined on a graph (N, A). The node set N consists of the set of customers and the node 0, which represent the depot. The customers are denoted by $1,2, \dots, n$. The arc set A corresponds to possible connections between the nodes. All routes start at 0 and end at 0 node. A cost (travel time) c_{ij} are associated with each arc $(i, j) \in A$ of the graph. The travel time c_{ij} includes the interval between customers i and j as well as service time at customer i. The set of (identical) vehicles is denoted by K. Each vehicle has a given capacity q and each customer a demand d. The VRPTW model contains two types of decision variables. The decision variable X_{ij}^k (defined $\forall(i, j) \in A, \forall k \in K$) is equal to 1 if vehicle k drives from node i to node j and 0 otherwise. The decision variable S_i^k (defined $\forall i \in N, \forall k \in K$) denotes the time vehicle k, starts service at customer i. If vehicle k does not service customer i, S_i^k has no meaning. We may assume that $S_0^k = 0$ denotes the arrival time of vehicle k at the depot. The objective is to design a set of minimal cost routes, one for each vehicle, such that all customers are serviced exactly once. The routes must be feasible with respect to the capacity of the vehicles and the time windows of the customers serviced. The VRPTW can be stated mathematically as:

$$minimize \quad \sum_{k=1}^{K} \sum_{(i,j)\in A} c_{ij} . X_{ij}^k \tag{1}$$

Subject to:

$$\sum_{k=1}^{K} \sum_{j=1}^{n} X_{ij}^k = 1, \qquad \forall i \in N \tag{2}$$

$$\sum_{i=1}^{n} d_i \sum_{j=1}^{n} X_{ij}^k \le q, \qquad \forall k \in K \tag{3}$$

$$\sum_{j=1}^{n} X_{0j}^k = 1, \qquad \forall k \in K \tag{4}$$

$$\sum_{i=1}^{n} X_{ih}^{k} \sum_{j=1}^{n} X_{hj}^{k} = 0, \qquad \forall h \in N, \forall k \in K \tag{5}$$

$$e_i \leq S_i^k \leq l_i, \qquad \forall i \in N, \forall k \in K \tag{6}$$

$$X_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in K \tag{7}$$

The objective function (1) states that costs should be minimized. Constraint set (2) states that each customer must be assigned to exactly one vehicle, and constraint set (3) states that no vehicle can service more customers than its capacity permits. Constraint set (4), and (5) are the flow constraints requiring that each vehicle k leaves node 0 once, leaves node h, if and only if it enters that node and returns to node 0. Constraint set (6) ensures that all time windows are respected and (7) is the set of integrality constraints.

IV. PROPOSED METHOD

The proposed method for vehicle routing problem is offered with time windows that includes optimized algorithm of the cuckoo search and greedy algorithm. Optimized algorithm of the proposed work deals with ideas for the innovative attempt to improve the resolution of its products. The architecture of our proposed algorithm is depicted in Fig. (3).

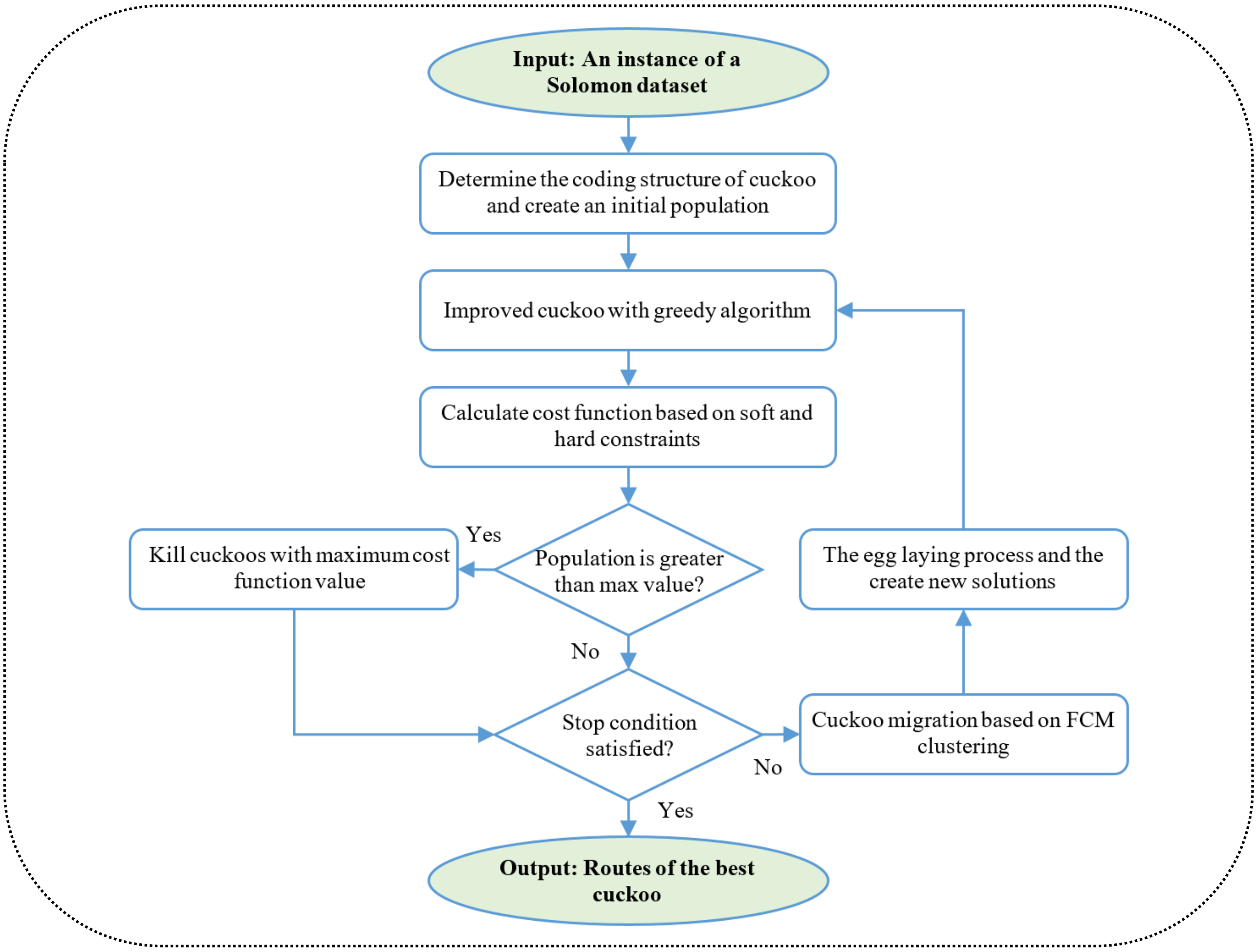


Fig 3. The architecture of the proposed algorithm

In the first step, the coding of solutions (cuckoo structure) and the process of initial population creation are described. The structure of the cuckoos is cell form and the initial population is randomly generated. To increase the quality of cuckoos, each cuckoo is improved by a greedy algorithm in the second step. In the third step, the cost function is presented with different criteria. The criteria considered in accordance with the objectives of the VRPTW problem are both hard and soft constraints. In the fourth step, the population size of the cuckoo is adjusted. At this step, if the number of cuckoos exceeds the maximum population size, the cuckoos with the maximum cost function are eliminated. In the fifth step, the cuckoo migration is done to get a better position. The migration process is based on FCM clustering and community identification. The laying process (creating new solutions) is performed in the sixth step. Here, each cuckoo is by number, and laying the radius, lays eggs within the specified range. The output of each step of the cuckoo algorithm in generating new solutions is improved by the greedy algorithm. These steps are repeated in sequence until a specified termination condition is reached.

The main contribution of the paper is to use greedy algorithms to improve solutions in the cuckoo algorithm. Besides, we reduce the problem constraints by two types of hard and soft. The process is to create feasible solutions and then improve their quality. In the following, the details of each section of the proposed method are described.

A. Cuckoo's Representation

Almost always, the coding structure is represented by an array of stops. Multiple variations of this representation have been used. The literature review revealed that the most used variation is the simplest one, where the array is filled with index numbers of the stops. Generally, the coding structure depends on the problem parameters and route goals. In this paper, we encode the cuckoo in the form of a coding structure. Each cell shows the structure of a vehicle that is encoded as an array. Fig. (4), shows an example of the cuckoo's representation structure.

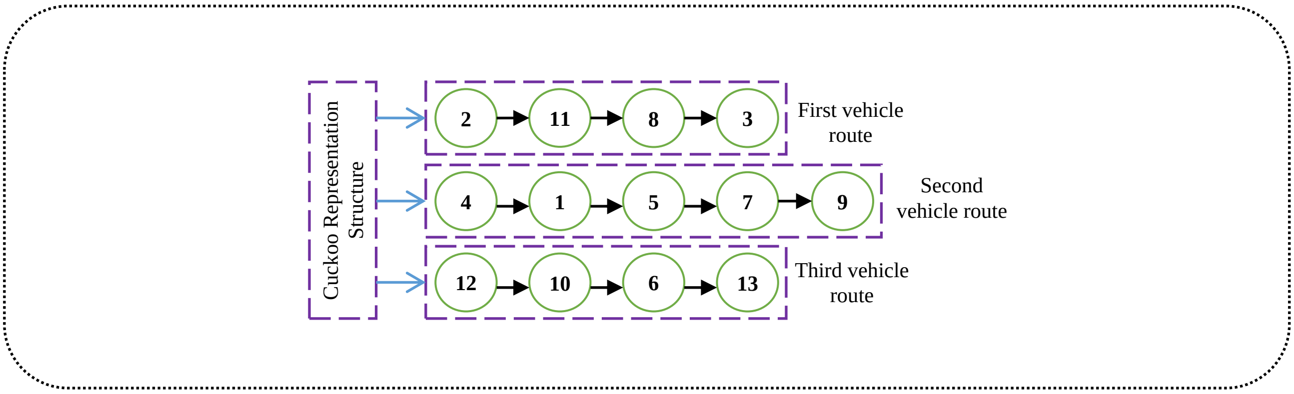


Fig. 4. An example of the cuckoo's representation structure

This structure is designed to simplifying the process of migration. In (Karakatič & Podgorelec, 2015 and Zhang & Hu, 2019) authors suggest multiple arrays to represent one structure where each array is a representation of one route. Given that the first and the end of each direction of the storage route, the apex with number 0 should be considered at the beginning and the end of each vector (route) in the coding structure for calculations. The number of routes is determined. This parameter K^{opt} determines the number of optimal vehicles. For the initial population, the number of cuckoo N_{pop} is randomly generated based on the defined structure.

B. Improved Solutions (Cuckoo) with Greedy Algorithm

Due to the production of the initial population, there may be additional transfers in the route. These transfers reduce the quality of the solution and reduce convergence speed. The concept of additional transitions is shown in Fig. (5).

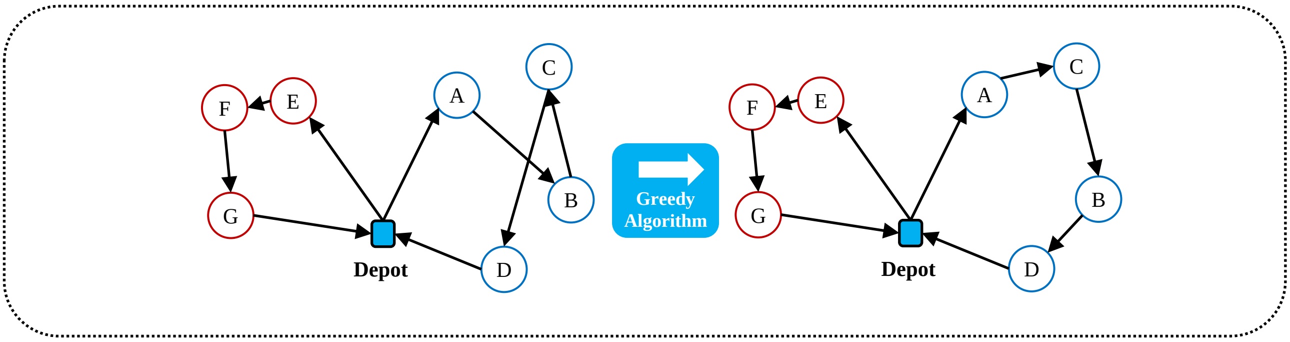


Fig. 5. An example of the proposed greedy algorithm

In this example, for the left graph there are additional transfers under the direction, $A \rightarrow B \rightarrow C \rightarrow D$. In this study, we use a greedy approach to improve solutions. This method is proposed to reduce the cost of the route (the sum of the intervals between nodes in the route). As in the right graph, additional transitions are eliminated by changing the route to $A \rightarrow C \rightarrow B \rightarrow D$. In the greedy proposed algorithm, all four nodes are examined from one route (remember, the cell is also a node). Assuming $A \rightarrow B \rightarrow C \rightarrow D$ subroutine, the route improvement process is performed if C_{AB} (distance between nodes A and B) is greater than C_{AC} (distance between nodes A and C). The recovery process is done by changing the routes of A and B along the way. Algorithm 1, shows the pseudocode of the proposed greedy algorithm.

input: cuckoo F_i and objective function $F_{obj}(F_i)$
for each cuckoo F_i in the population **do**
    for each route R_j of the cuckoo F_i **do**
        for each sub-route such as $A \rightarrow B \rightarrow C \rightarrow D$ of the route R_j **do**
            // All four consecutive nodes are selected from the route R_j (the depot is also a node).
            Changing the position of nodes B and C at new cuckoo F_*.
            if $F_{obj}(F_i) > F_{obj}(F_*)$ **then**
                Accept the new cuckoo; $F_i = F_*$. // The route changed to $A \rightarrow C \rightarrow B \rightarrow D$.
            end
        end
    end
end
output: cuckoo F_i

Algorithm 1: Pseudocode of the proposed greedy algorithm

C. Cost Function

Violating some of the constraints in the VRPTW issue leads to unacceptable solutions and some violating their quality. For this reason, we need to define a function with two different types of cost in this problem. In this study, we define the cost function based on two types of hard and soft constraints. Hard type includes constraints that breach creates unacceptable solutions.

- *First Hard Constraint (H_1):* The sum of customer demands per route exceeds the maximum vehicle capacity.

$$H_1 = \sum_{k=1}^{K^{opt}} \max[0, D_k - q_k] \tag{8}$$

Where, D_k is the sum of customer demands in k-vehicle and q_k is the k-vehicle capacity.

- *Second Hard Constraint (H_2):* The time a customer starts a vehicle service after the last customer service start.

$$H_2 = \sum_{k=1}^{K^{opt}} \sum_{i \in R_k} \max[0, T_i^k - T_i^D] \tag{9}$$

Where, T_i^k is the sum of k-th time of vehicle to customers i and T_i^D is the last time the service is started. Also, R_k is the route assigned to the k-th vehicle.

Third Hard Constraint (H_3): The service end time is longer than the maximum allowed time.

$$H_3 = \sum_{k=1}^{K^{opt}} \max[0, T_{end}^k - T_{max}] \tag{10}$$

Where, T_{end}^k is the routing end time for the k-th vehicle and T_{max} is the maximum routing time allowed.

- *Fourth Hard Constraint (H_4):* The number of vehicles used exceeds the maximum number of vehicles allowed.

$$H_4 = \max[0, K^{opt} - K] \tag{11}$$

In addition, the soft type includes constraints that do not make the solutions unacceptable but will reduce their quality.

- *Soft First Constraint (S_1):* Minimizing the number of vehicles.

$$S_1 = K^{opt} \tag{12}$$

- *Second Soft Constraint (S_2):* Minimizing total routing cost.

$$S_2 = \sum_{k=1}^{K^{opt}} \sum_{(i,j) \in R_k} c_{ij} \tag{13}$$

Where, c_{ij} is the route cost between customers i and j.

- *Third Soft Constraint (S_3):* Minimizing the total waiting time of vehicles.

$$S_3 = \sum_{k=1}^{K^{opt}} \sum_{i \in R_k} \max[0, T_i^R - T_i^k] \tag{14}$$

Where, T_i^R is the first time the service is started. The final cost function is calculated based on two types of hard and soft constraints. The Eq. (15) below shows the final cost function.

$$F_{obj} = \alpha \times [H_1 + H_2 + H_3 + H_4] + \beta \times [S_1 + S_2 + S_3] \tag{15}$$

Where, α and β are the weights of the hard and soft constraints, respectively.

D. Cuckoo's Migration

Once the cuckoo chicks are matured, they will live in their groups for a while, but at the time of egg laying, they will migrate to better habitats with higher chance of survival. After forming the cuckoo groups in various residential regions, the group with best position will be selected as the target point of migration for other cuckoos. When the adult cuckoos live in the whole environment, it is difficult to recognize that each cuckoo belongs to which group; therefore, to solve this problem, the cuckoo grouping is performed using FCM clustering algorithm. Now that the cuckoo groups are formed, the average profit of the group is calculated to obtain the relative optimality of that group's habitat; then, the group with the highest average profit (optimality) is chosen as the target group, and other groups will migrate toward it. In this paper, after identifying the communities (clusters), the average value of the cost function is calculated for each community and then each cuckoo with a probability of 50% to the optimal cuckoo in its community and with a probability of 50% to the best one (the community with the lowest means cost function) migrations.

In the process of cuckoo migration, cuckoo F_a moves to cuckoo F_b. So only the content of F_a changes. Since the search space is discrete, we use the γ coefficient parameter to control the migration. The higher the γ, the greater the similarity of cuckoos F_a and F_b after migration. Here, is the best value for γ is 0.35 experimentally. The migration process is as follows.

1. Select the number of γ% of routes in the cuckoo F_b.
2. Remove all nodes in common between F_a cache and selected routes from F_a cache.
3. The selected routes to the cuckoo F_a.
4. Calculate the cost function for cuckoo F_a with new routes.
5. Select two nodes from two different routes at random.
6. The node is removed from the first route and inserted in the next position of the selected node in the second route.
7. Calculate the cost function and apply changes to the cuckoo F_a, if the cost function is reduced.
8. Repeat steps 5, 6 and 7 in a fixed number.

Table II provides an example of the proposed migration process.

TABLE II. AN EXAMPLE OF THE CUCKOO MIGRATION PROCESS

The first assumed route	$\boldsymbol{A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F}$
The second assumed route	$\boldsymbol{G \rightarrow H \rightarrow I \rightarrow J}$
Select node from the first route (random)	C
Select node from the second route (random)	H
Confirm the first route if the cost improves	$A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$
Confirm the second route if the cost improves	$G \rightarrow H \rightarrow C \rightarrow I \rightarrow J$

E. The Egg Laying Process

The egg-laying process means the creation of new solutions during which each cuckoo lays eggs in the nests of the birds. In this study, we use the E_{max} symbol to indicate the maximum number of eggs per cucumber. By definition i-th cuckoo based on the probable parameter pa lays the number of E_i eggs in the range $[1, E_{max}]$ in the host bird's nest. According to the research Alssager and Othman (2016), $E_{max} = 3$ and $pa = 0.2$ is considered. This number is probabilistically calculated based on the following Eq. (16).

$$E_i = rand(1..E_{max}) * (1 - pa) \tag{16}$$

Specifically, the value of E_i specifies the number of solutions created by cuckoo F_i. In the cuckoo search algorithm, the radius of the egg is determined by the ELR variable. The radius of ELR is defined as the number of routes that can be changed from the mother caterpillar (the caterpillar) to create a new one. In order to reduce the time complexity, $ELR = 2$ is considered in this study. Depending on the number and radius of egg laying, each cuckoo population in the host bird nests will lay eggs. Here each egg represents a new solution that is produced on the basis of mother cuckoo. To generate a solution from cuckoo F_i, the number of ELR routes is first selected from this cuckoo (if the routes were lower than the ELR, all routes are selected). Then to create any new solution, the creation process is as follows.

To create each new solution, first select one route from the selected routes, and then replace all the nodes in that route with all available nodes (X) of the other routes. For a subroutine such as $A \to B \to C$, X is all nodes in the other selected routes. If the cost function is reduced by moving one of the nodes of the X list to B node, a new solution is created. Table III shows an example of this method.

TABLE III. AN EXAMPLE OF A SECOND APPROACH IN CREATING A NEW SOLUTION

Route assumed	$\boldsymbol{A \to B \to C \to D \to E \to F}$
Examine alternative nodes for node B	$A \to B \to C$ $\downarrow$ X
Examine alternative nodes for node C	$B \to C \to D$ $\downarrow$ X
Examine alternative nodes for node D	$C \to D \to E$ $\downarrow$ X
Examine alternative nodes for node E	$D \to E \to F$ $\downarrow$ X

Here, for the better perception of the coding of the algorithm, the main steps of proposed method are summarized as the pseudocode as shown in Algorithm 2.

input: N_{pop}, N_C
Initialize the cuckoos population $F_i(i = 1,2,\dots,N_{pop})$.
Calculate the cost function F_{obj} for $F_i(i = 1,2,\dots,N_{pop})$ through Eq. (15).
for each cuckoo F_i in the population **do**
    Apply greedy algorithm to reduce additional transfers on the cuckoo F_i according to Algorithm 1.
end
repeat
    if population size > N_{pop} **then**
        Kill cuckoos with maximum cost function value.
    end
    Apply FCM clustering to identify N_C communities of cuckoo.
    for each cuckoo F_i in the population **do**
        Selection of F_j cuckoo by the community with the lowest mean cost function.
        The cuckoo F_i migration towards the cuckoo F_j at new cuckoo F_*.
        Apply greedy algorithm on the cuckoo F_*.
        if $F_{obj}(F_i) > F_{obj}(F_*)$ **then**
            Accept the new cuckoo; $F_i = F_*$.
        end
        Calculate the parameter E_i through Eq. (16).
        The E_i number of new solutions (eggs) produced by cuckoo F_i.
        Apply greedy algorithm on the new solutions.
    end
until termination criterion not reached
output: cuckoo F_i with least cost function

Algorithm 2: Pseudocode of the proposed algorithm

V. EXPERIMENTAL RESULTS

In this section, we conduct a computational study to evaluate the performance of the proposed algorithm. All the results are averaged over 10 runs. We run all experiments on an Intel Core i5 CPU at 3.0 GHz and 8GB of memory, and the MATLAB version is R2017a, and Windows 10 64-bit operating system. To evaluate a variety of vehicle routing problem with time window developed in previous works, Solomon dataset instances from the UCI Machine Learning Repository has been widely used as a benchmark dataset.

The Solomon Dataset was created in 1982 by Solomon, one of the most important derivatives of the vehicle routing problem, namely the introduction of a time window. This dataset is available from reference (Xia & Fu, 2019). Instances produced in the Solomon dataset are presented in three types: C, R and RC. The difference in the type of instances is the number of customers considered. These instances have 25, 50 and 100 customers. In this study, C type C101, C102 and C103 instances, R type R101, R102 and R103 instances and RC type RC101, RC102 and RC103 instances were used in the experiments. From each instance the number of different customers is 25, 50 and 100. In this study, two criteria are used for number of optimal vehicles (NV) and total cost of the route (TT) as well as convergence for evaluation and comparison.

In addition, the recently introduced MPFIH (Wassan et al., 2017) and HPSO (Chen & Shi, 2019) algorithms are used for comparison. The MPFIH algorithm was developed by Mungwattana et al. based on a combination of local search and algorithm for vehicle routing problem with time window. This method takes advantage of two Modified Push Forward Insertion Heuristic (MPFIH) and λ-Interchange Local Search Descent (λ-LSD) in a two-objective genetic algorithm. These goals are to minimize the number of vehicles and minimize total travel time. The HPSO algorithm was developed by Chen and Shi based on a hybrid particle swarm optimization (HPSO) with simulated annealing for vehicle routing problem with time window. This method introduces a Multi-Compartment Vehicle Routing Problem with Time Window (MCVRPTW) arising from urban distribution, which essentially reflects the last mile delivery challenge for modern logistics.

A. Parameter Setting

The efficiency and effectiveness of meta-heuristic algorithms highly depend on the appropriate adjustment of the parameters. In most of the research studies, parameters are set based on either reference values of the literature or trial and error. Here, the proposed algorithm parameters are determined using Taguchi design approach to achieve the best solution (Azadeh et al., 2017). This method ensures the identification of effective parameters and levels with fewer experiments by providing balance among the orthogonal index, parameters, and levels (Alssager & Othman, 2016). The aim of Taguchi method is to maximize the S/N ratio (signal-to-noise) which is calculated by Eq. (17) for minimization problems for each parameter i on its related level j. The S/N values are calculated using the "least best" formula, since the shortest total routes cost is desired for the VRPTW.

$$S/N_{ij} = -10\log_{10}\left(\frac{1}{m}\sum_{i=1}^{m} F_{obj}(i,j)^2\right), \qquad \forall j \in level \tag{17}$$

Where, $F_{obj}(i,j)$ is the objective function value using the parameter i on level j and m is the number of times level j of parameter i is repeated over the runs of all trials.

The first step of parameter setting is to determine the parameters as control factors and their levels to implement the adopted experimental design. Table IV presents amount of parameters at each level for proposed method in Taguchi method. Because of similarity in reasoning procedure, the parameters setting process is only shown to customers of size 25. The results obtained through various designs of parameters based on standard table of orthogonal arrays L_{27}, are summarized in Table V. In this Table, rows denote the level of parameters in each experimental scheme and columns indicate a specific level of a parameter which is changeable in each scheme.

TABLE IV. LEVEL VALUES OBTAINED FOR PARAMETERS

Parameters	Level 1	Level 2	Level 3
N_{Pop}	10	15	20
N_C	2	3	4
$Iter_{max}$	50	100	200
α	0.75	0.8	0.85
β	0.15	0.2	0.25

TABLE V. RESULTS OBTAINED FROM DIFFERENT DESIGNS OF TAGUCHI APPROACH

Standard order	Parameters					Cost function
	N_{Pop}	N_C	$Iter_{max}$	α	β	
1	10	2	50	0.75	0.15	483.48
2	10	2	100	0.8	0.2	478.93
3	10	2	200	0.85	0.25	477.59
4	10	3	50	0.8	0.25	483.85
5	10	3	100	0.85	0.15	480.78
6	10	3	200	0.75	0.2	479.59
7	10	4	50	0.85	0.2	481.32
8	10	4	100	0.75	0.25	485.19
9	10	4	200	0.8	0.15	478.00
10	15	2	50	0.8	0.25	489.49
11	15	2	100	0.85	0.15	481.44
12	15	2	200	0.75	0.2	482.48
13	15	3	50	0.85	0.2	486.84
15	15	3	100	0.75	0.25	488.05
15	15	3	200	0.8	0.15	480.64
16	15	4	50	0.75	0.15	486.33
17	15	4	100	0.8	0.2	477.95
18	15	4	200	0.85	0.25	477.27
19	20	2	50	0.85	0.2	486.67
20	20	2	100	0.8	0.25	481.38
21	20	2	200	0.75	0.15	479.90
22	20	3	50	0.75	0.15	483.12
23	20	3	100	0.8	0.2	480.32
24	20	3	200	0.85	0.25	479.38
25	20	4	50	0.8	0.25	484.03
26	20	4	100	0.85	0.15	493.18
27	20	4	200	0.75	0.2	478.48

The S/N ratio obtained from the experiment is shown in Fig. (6). This figure, shows us the order of importance of the variables and displays the level at which variables should be used in order to achieve the best result. According to the S/N ratio, it is inferred that the N_{Pop}, N_C, $Iter_{max}$, α and β, which are 15, 4, 200, 0.75, and 0.25, respectively, is a good solution based on the adopted Taguchi method.

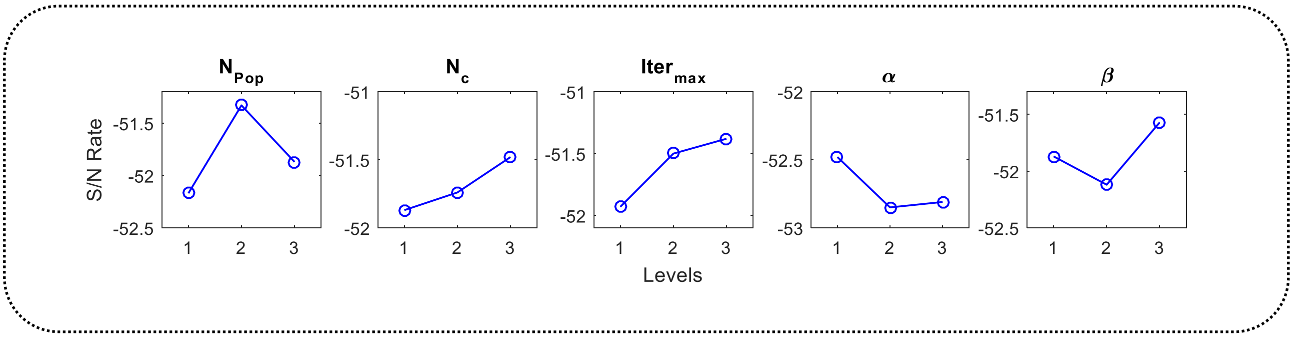


Fig. 6. The S/N ratio obtained in Taguchi method

However, based on Fig. (6), the best parameter set was identified according to the S/N values. The final setting of the parameters of the proposed algorithm is given in Table VI.

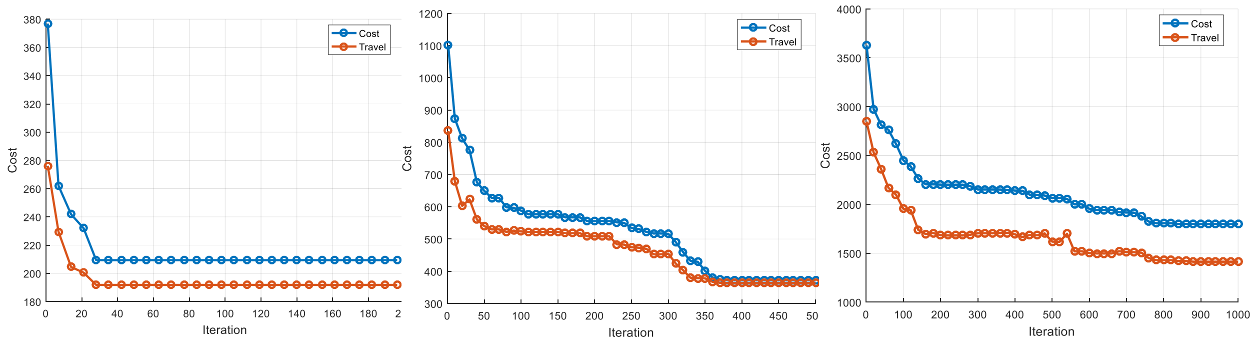
TABLE VI. FINAL PARAMETER VALUES FOR PROPOSED ALGORITHM

Parameters	Description	Values based on the number of customers		
		25	50	100
N_{Pop}	Number of cuckoos	15	25	30
N_C	Number of clusters	4	5	6
$Iter_{max}$	Maximum iteration	200	500	1000
α	Weight hard constraints	0.75	0.75	0.75
β	Weight soft constraints	0.25	0.25	0.25

B. Results and Discussion

In the first experiment, the convergence of the proposed method is investigated. Here the process of improving the problem objective function and travel cost relative to the number of iterations is examined. The performance of the proposed algorithm for C101-25, C101-50 and C101-100 instances is shown by the convergence in the Fig. (7). The convergence option in the statistics allows you to view the progress of a probabilistic analysis, to determine whether or not the analysis results are converging to constant values, and the approximate number of samples at which convergence occurs. Convergence is a phenomenon in evolutionary computation. It causes evolution to halt because precisely every individual in the population is identical. Full convergence might be seen in cuckoo algorithm (a type of evolutionary computation) using only migration (a way of make new solutions).

The results show that in instance C101 with 25 customers the proposed method converges in replication 28. The convergence results indicate a very good and fast performance of the proposed method in solving small-scale VRPTW problems. Also the results of the proposed method in the instance C101 with 50 customers show that convergence is done in 360 iterations with cost 390. Convergence for high-dimensional problems is achieved in C101 instance with 100 customers in replication 765. The results are similar for R and RC instances. These results indicate that the proposed method for solving high-dimensional VRPTW problems is of relatively high temporal complexity. Therefore, it is expected to provide better results by presenting approaches to reduce the time complexity of the proposed method and increase the number of iterations.



C101-25 instance C101-50 instance C101-100 instance

Fig. 7. The convergence of proposed algorithm on the Solomon dataset

The results of the simulation are compared with the MPFIH and HPSO algorithms to further evaluate the performance of the proposed method. The criteria are comparing NV (number of optimal vehicles) and TT (total cost of the route). In fact, NN is the number of vehicles determined by the proposed algorithm, and TT is the total cost of the route, including the sum of the second and third soft constraints $(S_2 + S_3)$. Table VII shows the results of the comparison of the proposed method with the MPFIH algorithm with an average of 15 distinct performances. In addition, Table VIII provides comparative results for the HPSO algorithm.

The results for both comparisons show the superiority of the proposed method in most of the instances and criteria. In other cases, the proposed method has shown acceptable results. For example, the proposed method in the instance C101-25 and TT reached 205.65 with about three vehicles, while MPFIH reports 252.33 and HPSO reports 213.45 with the same number of vehicles. In instance R101-25 the results of the proposed method are better than the MPFIH and HPSO algorithms. The results of the proposed method in RC101-25 instances are also better than the other two methods. This is due to the greater pressure and the chance of discovering better solutions in the greedy algorithm as well as innovative strategies for creating new caches that often result in lower cost solutions.

Generally, according to the instance data of Solomon's vehicle routing problem with time window, instances with 25 customers, 50 customers and 100 customers are adapted to test the performance of MPFIH and HPSO algorithms and proposed method with the identical parameters. The results show that three algorithms can solve the relevant cases within a reasonable time range - even the large-scale instances of 100 customer points can be solved within the 1300s. Although the HPSO algorithm shows the slight advantage in terms of the worst solution and standard deviation, the proposed method consistently delivers better results on the best solution than HPSO for the 25, 50, and 100-customer cases. Also, the efficiency of proposed method becomes higher than that of HPSO algorithm when the case size increases. More importantly, proposed method becomes increasingly more efficient than MPFIH as the customer number increases. In addition, no particular trends can be observed regarding the effect of geographical position distribution type of customers.

TABLE VII. COMPARISON OF PROPOSED METHOD AND MPFIH ALGORITHM

Instance	Evaluation criteria	Algorithms	Number of customers		
			25	50	100
C101	NV	MPFIH algorithm	3.33	5.89	17.33
		Proposed method	3.00	5.81	15.65
	TT	MPFIH algorithm	252.33	481.91	1563.79
		Proposed method	205.65	419.76	1489.17
C102	NV	MPFIH algorithm	1.88	3.13	12.38
		Proposed method	1.88	3.19	11.80
	TT	MPFIH algorithm	279.88	520.31	1252.96
		Proposed method	278.06	476.65	1431.39
C103	NV	MPFIH algorithm	8.10	9.81	13.04
		Proposed method	7.91	9.73	12.87
	TT	MPFIH algorithm	1106.37	1290.73	1477.39
		Proposed method	1048.83	1251.31	1482.40
R101	NV	MPFIH algorithm	3.75	6.17	12.50
		Proposed method	3.71	6.40	13.00
	TT	MPFIH algorithm	433.46	720.84	1326.93
		Proposed method	431.65	696.34	1301.76
R102	NV	MPFIH algorithm	2.00	4.09	7.27
		Proposed method	2.00	4.85	7.05
	TT	MPFIH algorithm	470.54	702.60	1227.61
		Proposed method	470.54	665.73	1174.32
R103	NV	MPFIH algorithm	14.50	14.05	16.18
		Proposed method	13.61	13.00	14.11
	TT	MPFIH algorithm	1289.33	1401.33	1570.47
		Proposed method	1319.80	1353.54	1581.76
RC101	NV	MPFIH algorithm	3.25	6.75	13.50
		Proposed method	3.00	5.25	13.50
	TT	MPFIH algorithm	345.28	711.19	1595.80
		Proposed method	348.21	625.23	1501.32
RC102	NV	MPFIH algorithm	2.61	4.75	8.25
		Proposed method	3.00	4.75	9.80
	TT	MPFIH algorithm	432.14	739.83	1463.30
		Proposed method	387.41	751.46	1398.13
RC103	NV	MPFIH algorithm	10.36	18.10	17.54
		Proposed method	10.00	16.30	16.43
	TT	MPFIH algorithm	1413.30	1489.28	1721.04
		Proposed method	1392.56	1518.50	1681.73

TABLE VIII. COMPARISON OF PROPOSED METHOD AND HPSO ALGORITHM

Instance	Evaluation criteria	Algorithms	Number of customers		
			25	50	100
C101	NV	HPSO algorithm	3.33	6.54	18.00
		Proposed method	3.00	5.81	15.65
	TT	HPSO algorithm	213.45	425.00	1504.78
		Proposed method	205.65	419.76	1489.17
C102	NV	HPSO algorithm	1.88	3.19	12.38
		Proposed method	1.88	3.19	11.80
	TT	HPSO algorithm	279.88	476.65	1300.64
		Proposed method	278.06	476.65	1431.39
C103	NV	HPSO algorithm	8.12	10.31	14.50
		Proposed method	7.91	9.73	12.87
	TT	HPSO algorithm	1087.12	1311.36	1387.23
		Proposed method	1048.83	1251.31	1482.40
R101	NV	HPSO algorithm	3.81	6.40	13.00
		Proposed method	3.71	6.40	13.00
	TT	HPSO algorithm	439.10	699.80	1302.48
		Proposed method	431.65	696.34	1301.76
R102	NV	HPSO algorithm	2.00	4.43	7.05
		Proposed method	2.00	4.85	7.05
	TT	HPSO algorithm	470.54	680.11	1205.39
		Proposed method	470.54	665.73	1174.32
R103	NV	HPSO algorithm	12.89	14.10	16.23
		Proposed method	13.61	13.00	14.11
	TT	HPSO algorithm	1342.32	1380.43	1472.30
		Proposed method	1319.80	1353.54	1581.76
RC101	NV	HPSO algorithm	2.83	7.05	12.45
		Proposed method	3.00	5.25	13.50
	TT	HPSO algorithm	351.58	631.73	1633.64
		Proposed method	348.21	625.23	1501.32
RC102	NV	HPSO algorithm	3.00	4.75	10.13
		Proposed method	3.00	4.75	9.80
	TT	HPSO algorithm	387.41	785.20	1401.43
		Proposed method	387.41	751.46	1398.13
RC103	NV	HPSO algorithm	9.71	17.22	19.23
		Proposed method	10.00	16.30	16.43
	TT	HPSO algorithm	1440.43	1571.49	1765.32
		Proposed method	1392.56	1518.50	1681.73

The approach exploits a population improved by the greedy algorithm, focusing on the minimization of traveled distance and temporal constraint violation. Additionally, the proposed hybrid algorithm uses both improvement and construction type of heuristics during its search and the algorithm got a more powerful search capacity. Results show that our methodology is able to provide fine-quality solutions which can compete with the ones provided by some exact and heuristic approaches. Moreover, because of its simplicity and flexibility, this methodology can easily be adapted to other variants of the vehicle routing problem and even to other combinatorial problems.

VI. CONCLUSION AND FUTURE WORK

In this paper, the vehicle routing problem with time windows (VRPTW) has been studied as an NP-hard problem. The objective of VRPTW is to serve all customers, at different geographic locations, with varying demands and within specific time windows. We proposed a combination of cuckoo search and greedy algorithm to solve this problem. Here, the output of each step of the cuckoo algorithm in generating new solutions is improved by the greedy algorithm, this process accelerating the convergence speed of the algorithm. Since violating some of the constraints in the VRPTW leads to unacceptable solutions and some of them lead to lower quality solutions, we define the cost function based on two types of hard and soft constraints. Here, high numbers of iterations produce significantly better results than the set number of iterations. This gives an idea that the convergence of cuckoo search to the optimal solution may be significantly slower (or proportional to the size of the problem instance). Computational results show that the performance of proposed algorithm is competitive on the Solomon instances in terms of solution quality when compared with the best solutions published so far. Where, the proposed method is 3.5% and 1.5% superior to MPFIH and HPSO algorithms respectively in the total cost of the route.

Future work will focus on implementing the proposed algorithm to different types of vehicle routing problems as well as to various combinatorial optimization problems, for example, the multi-depot vehicle routing problem and split delivery vehicle routing problem. Furthermore, future research can be done on the diversity control of the proposed algorithm so as to further improve its search diversity. Parallel implementation of the proposed technique will be explored as a natural step to gain significant speed-up as well.

REFERENCES

Abdallah, A. M. F., Essam, D. L., & Sarker, R. A. (2017). On solving periodic re-optimization dynamic vehicle routing problems. *Applied Soft Computing*, 55, 1-12.

Afshar-Nadjafi, B., & Afshar-Nadjafi, A. (2017). A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of king saud university-Engineering sciences*, 29(1), 29-34.

Alssager, M., & Othman, Z. A. (2016). Taguchi-based parameter setting of cuckoo search algorithm for capacitated vehicle routing problem. In *Advances in Machine Learning and Signal Processing* (pp. 71-79). Springer, Cham.

Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104, 124-133.

Alvarenga, G. B., Mateus, G. R., & De Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6), 1561-1584.

Alzaqebah, M., Abdullah, S., & Jawarneh, S. (2016). Modified artificial bee colony for the vehicle routing problems with time windows. *SpringerPlus*, 5(1), 1298.

Baranwal, M., Parekh, P. M., Marla, L., Salapaka, S. M., & Beck, C. L. (2016, July). Vehicle routing problem with time windows: A deterministic annealing approach. In *2016 American Control Conference (ACC)* (pp. 790-795). IEEE.

Brandão, J. (2018). Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows.

Computers & Industrial Engineering, 120, 146-159.

Chen, J., & Shi, J. (2019). A multi-compartment vehicle routing problem with time windows for urban distribution–A comparison study on particle swarm optimization algorithms. *Computers & Industrial Engineering*, 133, 95-106.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4), 568-581.

Dantzig, G. B., & Ramser, J. M. (1959) ,The truck dispatching problem. *Management Science*, 6, 81–91.

Eshtehadi, R., Fathian, M., & Demir, E. (2017). Robust solutions to the pollution-routing problem with demand and travel time uncertainty. *Transportation Research Part D: Transport and Environment*, 51, 351-363.

Esmaeili, M., & Sahraeian, R. (2019). Comparing Two-Echelon and Single-Echelon Multi-Objective Capacitated Vehicle Routing Problems. *Journal of Quality Engineering and Production Optimization*, 4(1), 1-16.

George, S., & Binu, S. (2018). Vehicle Route Optimisation Using Artificial Bees Colony Algorithm and Cuckoo Search Algorithm-A Comparative Study. *International Journal of Applied Engineering Research*, 13(2), 953-959.

Goel, R. K., & Bansal, S. R. (2020). Hybrid algorithms for rich vehicle routing problems: a survey. In *Smart Delivery Systems* (pp. 157-184). Elsevier.

Goel, R., & Maini, R. (2018). A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *Journal of Computational Science*, 25, 28-37.

Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems*, 42, 93-103.

Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3), 995-1018.

Kar, U. N., & Sanyal, D. K. (2020). A Critical Review of 3GPP Standardization of Device-to-Device Communication in Cellular Networks. *SN Computer Science*, 1(1), 37.

Karakatič, S., & Podgorelec, V. (2015), A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, 27, 519-532.

Kargari Esfand Abad, H., Vahdani, B., Sharifi, M., & Etebari, F. (2019). A Multi-Objective Optimization Model for Split Pollution Routing Problem with Controlled Indoor Activities in Cross Docking under Uncertainty. *Journal of Quality Engineering and Production Optimization*, 4(1), 99-126.

Lu, D., & Gzara, F. (2019). The robust vehicle routing problem with time windows: Solution by branch and price and cut. *European Journal of Operational Research*, 275(3), 925-938.

Marinakis, Y., Marinaki, M., & Migdalas, A. (2019). A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Information Sciences*, 481, 311-329.

Miranda, D. M., & Conceição, S. V. (2016). The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, 64, 104-116.

Nasri, M., Metrane, A., Hafidi, I., & Jamali, A. (2020). A robust approach for solving a vehicle routing problem with time windows with uncertain service and travel times. *International Journal of Industrial Engineering Computations*, 11(1), 1-16.

Nazif, H., & Lee., L. S. (2010), Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied*

Mathematical Modelling, 36(5), 2110-2117.

Reed, M., Yiannakou, A., & Evering, R. (2014). An ant colony algorithm for the multi-compartment vehicle routing problem. *Applied Soft Computing*, 15, 169-176.

Saxena, R., Jain, M., Malhotra, K., & Vasa, K. D. (2020). An Optimized OpenMP-Based Genetic Algorithm Solution to Vehicle Routing Problem. In *Smart Computing Paradigms: New Progresses and Challenges* (pp. 237-245). Springer, Singapore.

Vincent, F. Y., Redi, A. P., Hidayat, Y. A., & Wibowo, O. J. (2017). A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, 53, 119-132.

Wassan, N., Wassan, N., Nagy, G., & Salhi, S. (2017). The Multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers & Operations Research*, 78, 454-467.

Xia, Y., & Fu, Z. (2019). Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate. *Cluster Computing*, 22(4), 8725-8733.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In Nature & Biologically Inspired Computing, 2009. *NaBIC 2009. World Congress on* (pp. 210-214). IEEE.

Zhang, H., Zhang, Q., Ma, L., Zhang, Z., & Liu, Y. (2019). A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Information Sciences*, 490, 166-190.

Zhang, M., & Hu, Q. (2019). An effective evolutionary algorithm for the biobjective full truckload transportation service procurement problem. *Computers & Industrial Engineering*, 127, 1012-1023.

Zulvia, F. E., Kuo, R. J., & Nugroho, D. Y. (2020). A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products. *Journal of Cleaner Production*, 242, 118428.