On the Application of a Modified Genetic Algorithm for Solving Vehicle Routing Problems with Time Windows and Split Delivery

Olateju Samuel Olaniyi, Adebayo Kayode James, *Member, IAENG,* Abdullahi Adinoyi Ibrahim, *Member, IAENG,* and Aderibigbe Felix Makanjuola

***Abstract*—One of the variants of Vehicle Routing Problems (VRP) is the Vehicle Routing Problems with Time Windows and Split Delivery (VRPTWSD) considered by this paper in which, deliveries of customers' demands are allowed to be split between two or among many vehicles within a time frame. Obtaining an optimal solution for the total distance travelled whenever split is involved has been a major challenge. This paper formulated mathematical model for VRPTWSD aiming at minimizing the distribution cost or travel cost. Since this problem is NP-hard, researchers have developed variants Genetic Algorithms (GA) in solving this class of problem improving on preceding algorithms. This paper discusses modifications carried out on the Reproduction, Crossover and Mutation operators used in Genetic Algorithm which are applied in that order to the current population with a view to solving VRPTWSD. The Modified Genetic Algorithm (MGA) so formed is applied to various sizes of Solomon benchmark problems. The computational results presented show that out of the 56 instances considered, 53 instances were improved upon when compared with best known solutions based on the total distance travelled, average total distance and average number of vehicles. Also, it observes that, with the split deliveries, the total distribution cost tends to decrease. The MGA is seen to be efficient, robust and occupying less computer memory hence, recommended for future use.**

***Index Terms*—Split Deliveries, Time Windows, Mutation, Crossover, Modified Genetic Algorithm, Vehicle Routing Problems.**

I. Introduction

THE concept of Vehicle Routing Problem (VRP) was first put forward officially by [1]. VRP is a combinatorial optimization and integer programming problem which finds the most efficient routes for fleets of vehicles to service sets of customers subjected to some certain constraints [2]. Among the variants of VRP is the Vehicle Routing Problem with Time Windows and Split Deliveries (VRPTWSD) and it was first introduced by Dror and Trudeau [3]. Other VRP variants includes; Traveling Salesman Problem [4] and [5], Capacitated Vehicle Routing Problem (CVRP) [6], VRP with Time Windows [7], Pickup and Delivery VRP [8], Dynamic VRP treated by [9] and [10], VRP with Focus on Multiple Priorities [11] and lots more.

In VRPTWSD, the constraint that every customer is visited by one vehicle is relaxed and a customer can be visited by multiple vehicles [12]. The customer's demand can be split (split delivery) if the demand is greater than the vehicle carrying capacity. The VRPTWSD can be defined as an undirected graph $G(V,E)$ where $V = \{0,1,2,...,n\}$ denotes the set of vertices and $E = \{(i,j):\ i,j \in V, i<j\}$ denotes the set of edges on the graph, G. The vertex, 0, denotes the depot where the fleet of vehicles are placed at initial time and other vertices denote the $m-$customer nodes. Each customer is associated with a certain demand $d_i > 0$. On every edge $\{i,j\} \in E$ of G is a non-negative distribution cost, c_{ij}.

Applications of VRPTWSD to real life problems includes; helicopter screw scheduling [13], cattle feed distribution [14], newspaper distribution process [15] and many more. Metaheuristics have proven to be very efficient in solving variants VRP. The aim of this paper is to modify existing Genetic Algorithm (GA) for solving vehicle routing problem with time windows and split delivery. The objectives are as follow:

(*i*) to formulate a mathematical model for VRPTWSD;
(*ii*) to modify existing Genetic Algorithm (GA);
(*iii*) to implement the modified genetic algorithm on well-known instances to determine the total distance covered and the average number of vehicles;
(*iv*) to compare the results obtained with existing solutions.

The rest of the paper is structured as follow; related works on the problem are presented in section II. Section III and IV present the material, existing algorithm and methodology for this work and the computational results are shown in section V. Finally, VI concludes this paper.

The manuscript was received on 27th September, 2020; revised on 3rd August, 2021.

Olateju Samuel Olaniyi is a student of the Department of Mathematics, African Institute of Mathematical Science (AIMS), Senegal. (email: samuel.olaniyi@aims-senegal.org).

Adebayo Kayode James is a lecturer in the Department of Mathematics, Ekiti State University, Ado Ekiti, Ekiti State, Nigeria. (email: kayode.adeabyo@eksu.edu.ng).

Abdullahi Adinoyi Ibrahim is a lecturer in the Department of Mathematical Sciences, Baze University, Abuja, Nigeria. (email: abdullahi.ibrahim@bazeuniversity.edu.ng).

Aderibigbe Felix Makanjuola is a lecturer in the Department of Mathematics, Ekiti State University, Ado Ekiti, Ekiti State, Nigeria. (Email: felixaderibigbe@eksu.edu.ng.)

II. RELATED WORKS ON VRPTWSD AND GENETIC ALGORITHM (GA)

The authors in [16] developed a hybridized GA to solve VRPTWSD. The method focuses on the two-fitness approaches. Much more than the second fitness gave a better and semi-optimal solution. The hybrid algorithm was recommended for further research in order to obtain much more desired results. In [17], a localized optimization framework was developed for solving VRPTW and stressed on the scheme called Localized Genetic Algorithm (LGA). The LGA was applied to VRPTW for minimizing distance only. The result of the LGA scheme showed a superior performance, producing an improved solution when compared with other algorithms and also attained best solutions on some well-known data sets. According to [18], the authors solved SDVRP with 3*D* developing a hybridized algorithm which fusses the genetic algorithm and local search algorithm. Genetic Algorithm with some heuristics were applied on vehicle routing problem by [19]. In the process of obtaining solution, there were some tuning that were performed during mutation which led to better solutions and made the method relevant. Also, [20] proposed variation to Genetic Algorithm to solve Capacitated Vehicle Routing Problems. The proposed variation adopted optimized crossover operator that was tested on benchmark instances and the experimental results proved that the variation in the algorithm is highly competitive in its solution quality.

Other authors like [21] proposed heuristic into genetic algorithm for solving multi-depot vehicle routing problem with time windows (MDVRPTW). The authors concluded based on the computational result that, the proposed methods can be used to solve big size problems efficiently. In [22], the authors highlighted three contributions. First, reviewed previous literatures on green VRP. Secondly, implemented genetic algorithm to solve energy minimizing vehicle routing problem. Lastly, the authors showed some applications of machine learning for parameter tuning of meta-heuristics. According to [23], a *"three-stage approach"* was designed and implemented on split delivery vehicle routing problem. The results showed that the proposed algorithm yields feasible and effective solutions to solving the problem. [24] proposed a hybrid metaheuristics algorithm to solving multi-depot vehicle routing problem. The computational results showed that the proposed algorithm outperformed other implemented algorithms on the instances used. Marcos *et. al.* [25] proposed an iterated local search algorithm for solving split delivery vehicle routing problem. The computational result showed that the proposed algorithm was highly efficient as exactly 55 best-known solutions were equalled and there were 243 newly improved solutions out of the sampled considered. The algorithm showed an average improvement of 1.15% and the maximum improvement was 2.81%. In [26], the authors proposed and implemented hybrid metaheuristics algorithms on split delivery vehicle routing problems. The experimental results obtained showed that the proposed algorithms are competitive in terms of solution quality and time. [7] developed hybrid genetic algorithm and particle swarm optimization to solve the Solomon's instance for VRPTW. The results obtained were impressive as some of the results outperformed existing well-known solutions. [27] implemented genetic algorithm on vehicle routing problem with simultaneous pick-up and deliveries. The result of the study shows the implemented algorithm have both weak and strong feasibilities.

III. MATERIALS AND ALGORITHM

This section discusses the features necessary for adopting the genetic algorithm in solving VRPTWSD and explicitly explains the modifications on the genetic algorithm.

A. Nomenclatures

Notations to be used throughout this paper

- $G(V, E)$ is a graph G with vertex V and edges E.
- $E = \{(i, j) :\ i, j \in V, i < j\}$ set of edges.
- $V = \{0,\ 1, \ldots, n\}$ is the set of vertices. v_0 is the depot and $v_1,\ \ldots,\ v_n$ is the customer's node.
- $N = \{1,\ 2, \ldots, n\}$ is the number of customers.
- d_{ij}: symmetrical distance travelled.
- t_{ij}: time travelled from node *i* to *j*.
- $q_i,\ \ i = 1,\ 2,\ 3, ..., n$ demand at node *i*.
- $R_i\ =\ \{r_1(\pi_1),\ \ldots,\ r_i(\pi_i)\}$ denotes the route of a vehicle.
- $r_i(j)$: is the index of jth customer in the route (in python time).
- time (T_r): total time required for delivery.
- time (B_s): time taken for the algorithm to achieve best solution (in python time).
- NV: number of vehicles.
- TD: total travelled distance.
- S_i: is the service time at customer i.
- $[t^e,\ \ t^l]$: denote the earliest and latest time windows and $t^e <\ \ t^l$.
- $Q_k,\ k = 1,\ \ldots,\ m$ is the vehicle carrying capacity.

B. The Decision Variables

$X_{ij}^k = 1$, if j is supplied after i by vehicle v_m and 0, if otherwise.
b_i^k is moment at which service begins at customer i by vehicle v_k, $k = 1,\ 2,\ \ldots,\ m$.
Y_i^k is fraction of customer's demand i delivered by vehicle v_m.
The objective of the model is to minimize the total distance travelled with respect to the time windows constraints satisfying the split condition.

C. Priority Based on Time

A well-known priority placed on VRPs is the Time Windows, where every customer is associated with a time horizon, (see [28]), within which, the customer must be visited or serviced. Thus, VRP with Time Priority (VRPTP) is one of the most important extensions of the VRP. In the VRPTP, each customer specifies a time window within which the service must start and probably finish. The VRPTP can be used to model various real-life applications, such as itemized in [10].
In the VRP with Time Windows (VRPTW) credited to [29], the traversal time, t_{ij}, for each arc $(i, j) \in C$ and a time window $(t^e,\ \ t^l)$, (See [30]), which corresponds, respectively, to the earliest time, $t^e,$ and the latest time, $t^l,$ within which the vehicle should service the customer, c_i. A schedule, i.e., the combination of the starting times, T_{ik} for the service at a customer, c_i, when visited by vehicle v_k, is considered feasible if

$$t^e \leq T_{ik} \leq t^l\ \forall\ i \in C,\ k \in V, \tag{1}$$

(if vehicle v_k does not visit customer c_i, the time T_{ik} is irrelevant) and $\xi_{ijk} = 1$ implies that:

$$T_{ik} + t_{ij} \leq T_{jk} \;\; \forall \; (i, j) \in C, \; k \in V, \qquad (2)$$

holds. The latter constraint, (2), merges the routing decisions with the time schedule. They can be linearized by means of MTZ-like [31] constraints of the form:

$$T_{ik} - T_{jk} + M\xi_{ijk} \leq M - t_{ij} \; \forall \; (i, j) \in C, \; k \in V \qquad (3)$$

It is worthy of note that, with the above definitions, time windows are asymmetric in the sense that, arriving at a customer, c_i, before time t^e is allowed. In which case the vehicle has to wait until time t^e, while arriving later than time t^l is prohibited. Some authors also add service times, s_i, at vertices to their models. This is only a minor extension, since it can be included by properly redefining the travel times and time windows.

The time windows can be sub-divided into four frames. Each customer, $c_i \in N$, has a time windows, $t^e < t_{ij} < t^l$, i.e. an interval$(t^e, \;\; t^l)$. According to [11] and [10], the following scenarios arise as sub-divisions:

PT₁$= \left(t^e, \;\; t^l\right)$: This time window indicates that; the vehicle can arrive any time after the earliest, t^e, and must leave before the latest time, t^l. This implies that, the vehicle can arrive at the customer's place at any time of the day and depart at any time as long as the delivery is done before the latest departure time required by the customer. Of all the time windows, $\left(t^e, t^l\right)$ is one that gives room for the vehicle to service the customer at any convenient time within the working period, T_k.

PT₂ $= \left(t^e, \;\; t^l\right]$: Here, the vehicle may arrive at the customer's location any time within the working period but must depart on or before the allotted latest departure time set by the customer.

PT₃$= \left[t^e, \;\; t^l\right)$: Here, the vehicle arrives on or after the earliest time and leaves at any time before the latest departure time. It is closed at the earliest time but open at the latest time interval. In this case, should the vehicle arrive ahead of the arrival time, it cannot be allowed to discharge, hence, has to wait till the earliest arrival time.

PT₄$= \left[t^e, \;\; t^l\right]$: A vehicle that arrives earlier than t^e, has to wait until t^e before it can start serving the customer. Arriving later than t^l is not allowed rather the vehicle must leave at most by t^l. This case places restriction at both the arrival and departure time. It gives no room for the vehicle to come at just any time earlier than the earliest time and must depart on or before the latest departure time. Should the vehicle not have finished discharging, it must leave at the latest departure time to give room for other things as the case may be. This case calls for the unloading time not to be elongated unnecessarily as the customer might have other things to attend to.

The time priority may be represented using the tree diagram in figure 1.

From figure 1, $PT(c_i\;)$ represents the time window priority of a customer, c_i, that can only be linked to a time window priority, $\left(t^e, t^l\right)$, $\left(t^e, t^l\right]$, $\left[t^e, t^l\right)$ or $\left[t^e, t^l\right]$ at a time. Such that:

$$PT(c_i\;) = PT_1 \; or \; PT_2 \; or \; PT_3 \; or \; PT_4 \qquad (4)$$

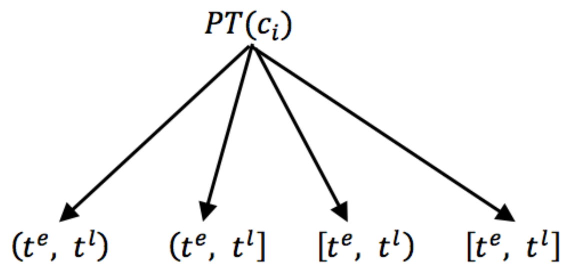

Fig. 1. Time Priority Tree

D. Priority Based on Split Deliveries

In real-life settings where the vehicles used are homogenous, split deliveries occur when the demand of a customer cannot be met by just one vehicle as in [32], [33] and more, which [34] reasoned out as Non-split and Split Services.

Until very recently, we have assumed that all service tasks are performed by a single vehicle in one service operation, i.e., services are non-split. However, there are reasons for splitting some services: On one hand, if demand exceeds the vehicle capacity, more than one visit is unavoidable. On the other hand, splitting service into several smaller services request can produce significant cost savings. The Split Delivery VRP (SDVRP) [35] and [36], allows, in principle, that each demand be split into arbitrarily many smaller demands served by different vehicles.

Obviously, the reasons for split deliveries could be broadly and explicitly classified under the followings:

(i) the vehicle, v_k, has serviced some customer(s), $c_1, \; c_2, \; \ldots, \; c_{N-n}$, along the route, $r_i\,(x_i\;)$, with the quantities, $q(c_1), \; q(c_2), \; \ldots, \; q(c_{N-n})$, where n is the number of customers that have been serviced on the route, thereby causing the quantity, q, to be delivered to customers, c_{N-n+1}, along the route not sufficient. Hence, called for a split delivery by another vehicle, v_{k+1}, rather than v_k to make up for the remaining. If the serviced customers, $c_1, \; c_2, \; \ldots, \; c_{N-n}$, by the vehicle, v_k, get the quantities:

$$q\left[c_1\,(v_k\;)\right] + q\left[c_2\;\;(v_k\;)\right] + \cdots + q[c_{N-n}\,(v_k)] = \sum_{i=1}^{N-n} q[c_i\,(v_k)] \qquad (5)$$

then, from (6) it implies that, the quantity, q, demanded by the customer, $q(c_{N-n+1})$, can be expressed as:

$$\sum_{i=1}^{N-n} q[c_i\;\;(v_k)] + q[c_{N-n+1}\;\;(v_k\;)] > Q(v_k) \qquad (6)$$

If (7) holds then, it leads to

$$q[c_{N-n+1}\,(v_k)] > Q(v_k) - \sum_{i=1}^{N-n} q\left[c_i\;\;(v_k)\right] \qquad (7)$$

From (7), the split quantity required by $q[c_{N-n+1}\,(v_k)]$ is the remaining quantity:

$$q\left[c_{N-n+1}\,(v_k)\right] - \{Q(v_k) - \sum_{i=1}^{N-n} q\left[c_i\;\;(v_k)\right]\} = Q(v_{k+1}) \qquad (8)$$

Since the quantity that will be delivered by $Q(v_{k+1})$ cannot be determined a priori then:

$$q\left[c_{N-n+1}\left(v_k\right)\right] = Q(v_k) + Q(v_{k+1}) - \sum_{i=1}^{N-n} q\left[c_i \ \ (v_k)\right] \quad (9)$$

(ii) the quantity, q, that is demanded by the customer, $q(c_i)$, exceeds the carrying capacity, Q, of the vehicle, $Q(v_k)$. This is given by the relation:

$$q(c_i) = q[c_i \ \ (v_k)] > Q(v_k) \quad (10)$$

leading to

$$q(c_i) = Q(v_k) + Q(v_{k+1}) - q[c_i \ \ (v_k)] \quad (11)$$

where $Q(v_{k+1})$ is the carrying capacity of vehicle v_{k+1} and $q[c_i\,(v_k\ )]$ implies the quantity delivered or to be supplied to customer c_i by v_k. The fractional part or whole of $Q(v_{k+1})$ that is added to $Q(v_k)$ in order to make up for the quantity required by the customer, $q(c_i)$ is the split quantity.
(iii) a rider to (11) is when the quantity, q, that is demanded by the customer, $q(c_i)$, exceeds the carrying capacity, Q, of the vehicle, $Q(v_k)$ and such customer's demand has to be met by more than two vehicles, $v_k, \ v_{k+1}, \ \ v_{k+2}, \ldots$ thus:

$$q(c_i) = Q(v_k) + Q(v_{k+1}) + Q(v_{k+2}) + Q(v_{k+3}) + \cdots - q[c_i \ \ (v_k)] \quad (12)$$

Here, all the vehicles, $v_k, \ v_{k+1}, \ \ v_{k+2}, \ldots$ have to leave the depot for the customer directly. In order not to violate the time window priority, the vehicles might need to time their departure from the depot for the customer in order not to cluster at the customer's warehouse.

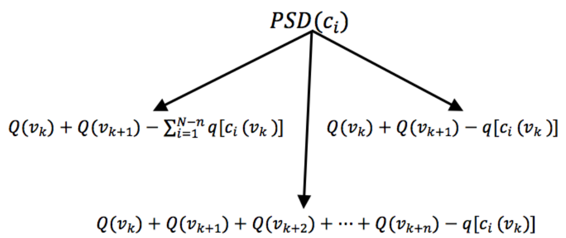

Fig. 2. Split Deliveries Priority Tree

From figure 2, a customer, c_i, can only be linked to a split delivery priority only once such that:

$$PSD(c_i) = Q(v_k) + Q(v_{k+1}) - \sum_{i=1}^{N-n} q\left[c_i \ \ (v_k)\right] \quad (13)$$

or

$$PSD(c_i) = Q(v_k) + Q(v_{k+1}) - q[c_i \ \ (v_k)] \quad (14)$$

or

$$PSD(c_i) = Q(v_k) + Q(v_{k+1}) + Q(v_{k+2}) + \cdots + Q(v_{k+n}) - q[c_i \ \ (v_k)] \quad (15)$$

where $PSD(c_i)$ represents the priority of a customer. It must be put on record that, not every customer is bound to have priorities based on split deliveries.

E. *Mathematical Formulation*

The dynamic of the VRP objective function is as follow:

$$min \sum_{i=0}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} d_{ij}X_{ij}^k \quad (16)$$

Subject to:

$$\sum_{j=1}^{n} X_{0j}^k = 1, \ k = 1, 2, \ \ldots, \ m \quad (17)$$

$$\sum_{i=0}^{n} X_{ip}^k - \sum_{j=0}^{n} X_{pj}^k = 0, \ p = 0, \ 1, \ \ldots, \ n \quad (18)$$

$$\sum_{k=1}^{m} Y_i^k = 1, \ i = 1, \ 2, \ \ldots, \ n \quad (19)$$

$$\sum_{k=1}^{n} q_i Y_i^k \leq a_k, k = 1, \ 2, \ \ldots, m \quad (20)$$

$$Y_i^k \leq \sum_{k=1}^{n} X_{ij}^k, \ i = 1, 2, \ \ldots, \ n; k = 1, \ 2, \ \ldots, \ m \quad (21)$$

$$\sum_{k=1}^{m}\sum_{j=0}^{n} X_{ij}^k \geq l, \quad j = 0, 1, \ldots, n \quad (22)$$

$$b_i^k + s_i + t_{ij} - M_{ij}\left(1 - X_{ij}^k\right) \leq b_j^k \quad (23)$$

$$i = 1, \ldots, n, \quad j = 1, .., n, \quad k = 1, \ldots, n$$

$$e_i \leq b_i^k \leq l_i, \ i = 1, \ \ldots, \ n \quad (24)$$

$$Y_i^k \geq 0, \quad i = 1, ..., n; \ k = 1, ..., m$$

$$b_i^k \geq 0, \ i = 1, ..., n; \ k = 1, ..., m \quad (25)$$

$$X_{ij}^k \in \{0, 1\}, i = 0, 1, \ldots, n; j = 1, \ldots, n \ and \ k = 1, \ldots, m \quad (26)$$

where (17) ensures that each vehicle will leave the depot and arrive at a customer's location. (18) ensures that each vehicle will leave a determined customer and arrive back to the depot. (19) ensures that the total demand of each customer will be satisfied. (20) ensures that the vehicle capacity will not be exceeded. (21) ensures that the demand of the customer will be satisfied if a determined vehicle goes by that place. (22) ensures that each point will be visited at least once by one vehicle. (23) is set of minimum time for beginning the service of customer j in a determined route and also there will be on sub tours. The constant M_{ij} is a large enough number, for instant, $M_{ij} = l^i + t_{ij} - e^j$. (24) ensures that all customers will be served within their time windows. (25) ensures that the decision variables v_i^k and b_i^k are positive. Lastly, (26) ensures the decision variables X_{ij}^k to be binary.

F. Genetic Algorithm

Genetic Algorithm (GA) is inspired by the natural selection mechanism introduced by [37]. The procedure is applied for a pre-selected number of iterations and the output result of the algorithm is the best solution found in the last population or, in some cases, the best solution found during the evolution of the algorithm. The operators used by GA simulates the ways natural selection are carried out. The most widely used operators used in GA are *Reproduction*, *Crossover*, and *Mutation* operators which are applied in that order to the current population.

The fitness assignment evaluates the chromosome based on the ranking the value of the chromosome. This function is defined as: (a) From the objective function (16), the function value is calculated for individual. (b) the values for the chromosomes are evaluated by comparing the values of the chromosomes through sorting as created in objective function. The function is defined as:

$$E(x_i) = \begin{cases} (N - R(x_i))^2, & if \quad R(x_i) > 1 \\ kN^2, & R(x_i) = 1 \end{cases} \tag{27}$$

where N is the number of individuals, x_i is the ith individual, $R(x_i)$ is the sequence number of ith individual and $k, 1 < k < 2$ is used to increase $E(x_i)$.

Summary of Genetic Algorithm

- **Step 0**:
  (a) Generate a random population X of N feasible chromosomes.
  (b) For each chromosome s in the selected population, evaluate its associated fitness. Record s^* as the best solution so far available.
  (c) Encode each chromosome using binary or numeric representation.
- **Step 1**:
  (a) Select two parent chromosomes from population X.
  (b) Crossover the parents' genes to create two children.
  (c) Mutate the children genes randomly.
  (d) If resulting solutions are infeasible, repeat step 1 until feasibility is achieved. Else, replace the weakest two parents with the new children to form a new population X and update s^*. Go to step 2.
- **Step 2**: If a termination condition has been reached, stop; s^* is the best available solution. Else, repeat step 1.

The Existing Genetic Algorithm is improved upon leading to what follows in the next section.

IV. Modified Genetic Algorithm (MGA)

An important area of this paper is the modification that has been done on the existing GA. This is presented herein in Algorithm 1.

V. RESULTS AND DISCUSSIONS

The proposed MGA was implemented in python 3.7 on hp $4GB$ RAM, $2.5Ghz$ processor, core $i7$ laptop.

Algorithm 1: Modified Genetic Algorithm Procedure

Step 1: Initial Population Random generation of chromosome α
Step 2: Initial Population Sorting Sort the initial population based on the total travel time of each chromosome.
Step 3: Counter Setup Set $a, b = 0$
Step 4: Apply Crossover Let $a = a + 1$; Select two parents, $P1$ and $P2$. Apply crossover to $(P1,\ P2)$ and choose one child, C_h at random.
Step 5: Apply Mutation Compare the randomly generated r_g and the Mutation rate M_r. If $r_g > M_r$, then proceed to Step 6. Otherwise, mutation is applied to C_h and c_h^0 is generated using split procedure.
Step 6: Iteration generation $\Delta = 1$. Δ solutions with distinct integer costs. Cost (C_h) $-$ cost $(\delta k) \geq \Delta$, $1 \leq K \leq \alpha$. δ_k is the best solution of the population Then, let $b = 0$ and Ch^0 to resort population. Otherwise, $b = \ b + 1$
Step 7 (Generation limitations) If $a \ = \ a_{max}$ or $b = bmax$, stop otherwise, go to step 4. a_{max} is the maximal number of crossovers that doesn't yield a gene and $bmax$ is the maximal number of iterations without the best solution.

A. Solomon Benchmark Problem Sets

The public available Solomon benchmark's problem set [38] consists of 100 customers with Euclidean distance. According to the problem, the percentage of customers with time windows varies between 25, 50, 75 and 100.

Six sets of problems: Route 1 (R1), Route 2 (R2), Customer 1 (C1), Customer 2 (C2), Route of Customer 1 (RC1) and Route of Customer 2 (RC2). In sets R1 and R2, the customer's positions were created randomly through a uniform distribution. The sets C1 and C2 customers are divided in groups. In the sets RC1 and RC2, customers are in sub-groups, that is, some of the customers is placed randomly and some is placed in groups.

Besides, R1, C1 and RC1 problems have a short-term planning horizon and combined with lighter capacity vehicles. Only some customers are allowed in each route. Sets R2, C2 and RC2 have a long-term planning horizon and since they got a higher capacity vehicle, they are able to supply more customers per route.

In each of the six set of problems, it is assumed that the geographical distributions of the customers, the demand and the service time do not change. Therefore, on set R1, the problems from R101 to R104 are identical, except for the customer with time window percentage, which is 100% in problem R101, 75% in problem R102, 50% in problem R103 and 25% in problem R104. Problems from R101 to R104 are identical to problems from R105 to R108, except for the difference is the time windows. Each Solomon's problem instance specifies the central depot, the maximum vehicle number, carrying capacity of the vehicles, demands of each customer and the maximum travel time of each vehicle.

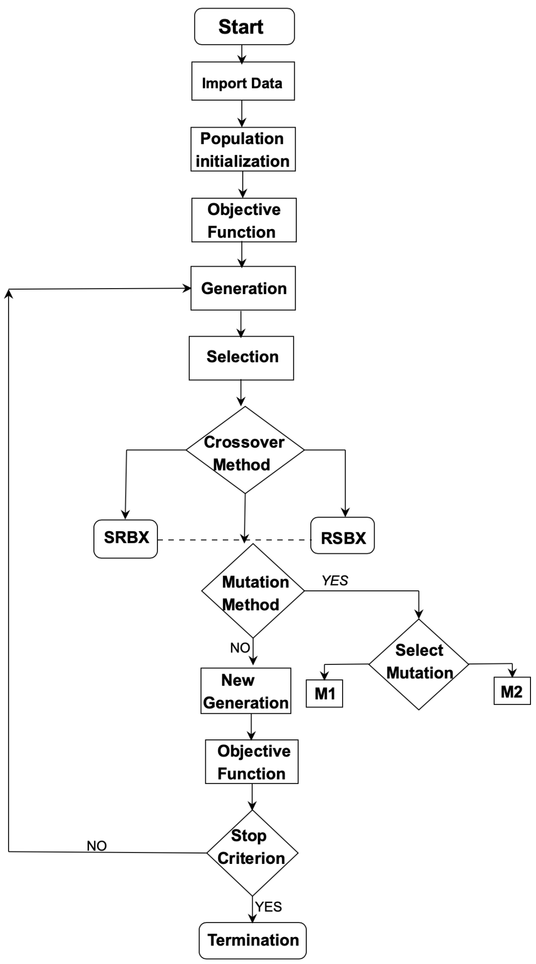

Fig. 3. Flowchat

B. Results Evaluation

In order to determine the efficiency and robustness of the MGA, two different evaluations are carried out: the total distance travel and the average total distance travelled. To compute the Deviation Percentage, (DP), between the Best Solution, (B_s), and the Best-Known Solution, (B_k), we use the relation:

$$Dev.\ \% = \frac{B_s - B_k}{B_k} \times 100\% \tag{28}$$

C. Analysis

The modified algorithm is applied to Solomon's benchmark problem, which consists of 56 instances. The MGA will be compared with other well-known results namely; Best-known solution (B_s), Particle Swam Optimization (PSO) and Ant Colony Optimization (ACO) as in [7].

Based on the formulated model, one of the major concerns will be to minimize the total distance travelled, which are presented in tables below.

In Table I, let the problem instance be (Probs), total travelled distance (TD), number of vehicles required for delivery (NV), total time required for delivery, (T_r) and time taken, in python time, to obtain the best solution (B_s).

The results of the Modified Genetic Algorithm as presented in Table I shows that, out of 56 instances tested on the algorithm, 53 instances achieved the best total distance travelled when compared with best known solution. This implies that, the proposed algorithm shows an outstanding and impressive results for the total distance travelled. While applying the MGA has reduced the number of vehicles for some instances, the split deliveries constraint helped to assign multiple vehicles to a single customer and this feature has contributed to the improvement of our solution.

Also, the number of vehicles utilized for total distance travelled was reduced for some instances. The percentage of deviation of the obtained results from the well-known solution showed that, the proposed algorithm produced optimal solutions. For instances with negative $\%Dev$, it means the obtained total travelled distance outperformed the previously known and existing optimal solutions.

For the implemented instances, we used heuristics in some places in order to place missing customer's information back into the different routes, after they have vanished during crossover, mutation or both processes. Though, different probabilities are considered on how often the heuristic should be used and different penalties found.

In this paper, we set our maximum generation limit to 50 and test run with a population size of 100 as well as the lower heuristic probability. As the probability gets higher, every test ran took relatively the same time. Details of this is shown in figure 1.

Table II depicts the best average distance, the number of vehicles used to achieve this and the corresponding authors for each instance. From the results, it was observed that, the MGA compared favourably with the existing results. It presents the average number of vehicles as well as the average total distance travelled with respect to the problem set. The results in Table II shows that, the values of total distance travelled on average is very close to the best solution obtained. This gives us an insight that, the modified GA yields a robust result on its application to real-life situations. Though, many noticeable researchers described different algorithms for solving VRPTWSD with different crossover and mutation operators, this paper presents a comparison of the result of the MGA with other well-known solution as shown in Table II.

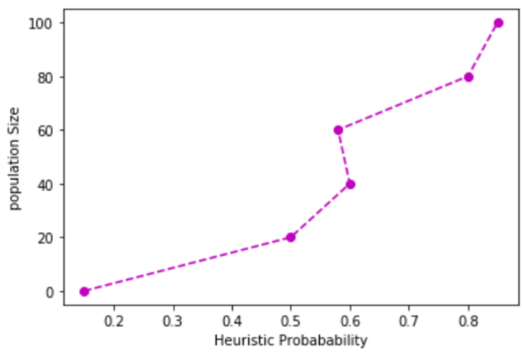

Fig. 4. Heuristic probability

In figure 4, the recorded custom insertion which were test

TABLE I
TOTAL DISTANCE TRAVELLED BASED ON MODIFIED GENETIC ALGORITHM

No.	**Probs**	**Nodes**	**Best known** [39]		**PSO** [39]		**ACO** [39]		**Modified Genetic Algorithm (MGA)**				
			TD	NV	TD	NV	TD	NV	NV	TD	time(T_r)	time(BS)	% Dev
1	C101	101	828.94	10	828.94	10	828.94	10	10	**820.98**	11157.97	3.10	−0.96
2	C102	102	828.94	10	828.94	10	828.94	10	10	**824.77**	7889.79	2.19	−0.50
3	C103	103	828.06	10	828.06	10	828.06	10	10	**825.41**	5015.33	1.39	−0.32
4	C104	104	824.78	10	824.78	10	824.78	10	10	**823.47**	4625.59	1.28	−0.16
5	C105	105	828.94	10	828.94	10	828.94	10	10	**819.95**	10283.75	2.86	−1.08
6	C106	106	828.94	10	828.94	10	828.94	10	10	**820.97**	9790.83	2.72	−0.96
7	C107	107	828.94	10	828.94	10	828.94	10	10	**821.47**	9377.16	2.60	−0.90
8	C108	108	828.94	10	828.94	10	828.94	10	10	**824.38**	8485.72	2.36	−0.55
9	C109	109	828.94	10	828.94	10	828.94	10	10	**823.75**	6516.58	1.81	−0.63
10	R101	101	1483.57	16	1642.87	20	1645.79	19	9	**933.3**	3686.91	1.02	−37.09
11	R102	102	1355.93	14	1472.62	18	1480.73	18	9	**1169.07**	9404.16	2.61	−13.78
12	R103	103	1133.35	12	1213.62	14	1213.62	14	8	**698.16**	9891.74	1.84	−38.40
13	R104	104	968.28	10	1007.24	9	982.01	10	8	**702.31**	4862.61	1.35	−27.47
14	R105	105	1262.53	12	1360.78	15	1360.78	15	8	**1113.91**	9972.59	2.77	−11.77
15	R106	106	1201.78	12	1241.52	13	1251.98	12	8	**703.94**	4375.67	1.22	−41.43
16	R107	107	1051.92	11	1076.13	11	1076.13	11	8	**695.34**	3602.06	1.00	−33.90
17	R108	108	948.57	10	963.99	9	948.57	10	8	**697.86**	3982.17	1.11	−26.43
18	R109	109	1110.40	12	1151.84	13	1151.84	13	8	**700.13**	8687.20	2.41	−36.95
19	R110	110	1080.36	11	1080.36	11	1080.36	11	8	**702.31**	7542.01	2.10	−34.99
20	R111	111	987.80	10	1053.50	12	1088.48	12	8	**703.60**	9017.18	2.50	−28.77
21	R112	112	953.63	10	953.63	10	953.63	10	8	**701.13**	3601.56	1.00	−26.48
22	C201	201	591.56	3	591.56	3	591.56	3	3	**409.24**	46177.97	12.83	−30.82
23	C202	202	591.56	3	591.56	3	591.56	3	3	**406.42**	37276.42	10.35	−31.30
24	C203	203	591.17	3	591.17	3	591.17	3	3	**407.02**	28928.30	8.036	−31.15
25	C204	204	590.60	3	590.60	3	590.60	3	3	**408.17**	18408.64	5.11	−30.89
26	C205	205	588.88	3	588.88	3	588.88	3	3	**407.67**	43410.51	12.06	−30.77
27	C206	206	588.49	3	588.49	3	588.49	3	3	**406.35**	40554.26	11.27	−30.95
28	C207	207	588.29	3	588.29	3	588.29	3	3	**407.99**	37548.86	10.43	−30.65
29	C208	208	588.32	3	588.32	3	588.32	3	3	**408.14**	37060.18	10.29	−30.63
30	R201	201	1148.48	9	1179.79	9	1148.48	9	2	**679.56**	10105.53	2.81	−40.83
31	R202	202	1049.74	7	1049.74	7	1079.36	6	2	**678.74**	7819.19	2.17	−35.34
32	R203	203	900.08	5	932.76	7	939.54	3	2	**680.39**	5709.81	1.59	−24.41
33	R204	204	772.33	4	772.33	4	772.33	4	2	792.95	6213.03	1.73	2.67
34	R205	205	959.74	4	970.89	6	970.89	6	2	**681.44**	7562.29	2.10	−29.00
35	R206	206	898.91	5	906.14	3	906.14	3	2	**683.58**	5750.57	1.60	−23.95
36	R207	207	814.78	3	814.78	3	814.78	3	2	**681.92**	4029.05	1.12	−16.31
37	R208	208	715.37	3	725.42	4	725.75	2	2	**679.40**	4193.44	1.17	−5.03
38	R209	209	879.53	6	879.53	6	879.53	6	2	**685.74**	3849.57	1.07	−22.03
39	R210	210	932.89	7	954.12	3	939.34	3	2	**683.97**	6473.33	1.80	−26.68
40	R211	211	761.10	4	885.71	2	888.73	5	2	**678.51**	3681.63	1.02	−10.85
41	RC101	101	1481.27	13	1623.58	15	1639.97	16	10	**1170.92**	13073.77	3.63	−20.95
42	RC102	102	1395.25	13	1482.91	14	1477.54	13	9	**1169.07**	9404.16	2.61	−16.21
43	RC103	103	1221.53	10	1262.02	11	1262.02	11	10	**1160.88**	6635.94	1.84	−4.97
44	RC104	104	1135.48	10	1135.48	10	1135.48	10	10	**1123.02**	5040.29	1.40	−1.10
45	RC105	105	1354.20	12	1518.60	16	1629.44	13	9	**1113.91**	9972.59	2.77	−17.74
46	RC106	106	1226.62	11	1384.92	12	1377.35	13	10	**1118.12**	7736.18	2.15	−8.85
47	RC107	107	1150.99	10	1212.83	12	1230.48	11	10	**1107.56**	4020.34	1.12	−3.77
48	RC108	108	1076.81	10	1117.53	11	1117.53	11	10	1115.28	1115.28	1.01	3.57
49	RC201	201	1134.91	6	1406.91	4	1286.83	9	2	**898.20**	9492.81	2.64	−20.86
50	RC202	202	1113.53	8	1113.53	8	1113.53	8	2	**896.78**	7279.51	2.02	−19.47
51	RC203	203	945.96	5	945.96	5	1049.62	3	2	**899.64**	5378.87	1.49	−4.90
52	RC204	204	796.11	4	798.41	3	798.46	3	2	894.07	4503.97	1.25	12.30
53	RC205	205	1168.22	8	1168.22	8	1168.22	8	2	**897.86**	8125.09	2.26	−23.14
54	RC206	206	1059.89	7	1084.30	8	1059.89	7	2	**904.24**	6524.38	1.81	−14.69
55	RC207	207	976.40	7	976.40	7	1053.58	6	2	**900.75**	4201.01	1.17	−7.75
56	RC208	208	785.93	4	816.10	5	806.87	5	2	**449.19**	3608.54	1.00	−42.85

runs with a population size of 100, shows that as the heuristic probability gets higher on every test run, it took more or less the same time to get a better solution.

VI. CONCLUSION

The modification to GA proposed in this paper yielded impressive results as compared with previously known solutions. The results as presented in Table I showed that out of 56 instances considered, the total distance for 53 instances were improved upon. The results were compared on the total distance travelled and the average total distance travel for each of the problem instances. The results obtained shows that the MGA used the least number of vehicles with comparable average travelled distance which was consistent for all implemented problem data sets.

The crossover operator searches for common parts between parent solutions and compared with other crossover operators that also deal with custom insert heuristics for constructing feasible solutions. Thus, on evaluating of the outcome, it became evident that, the choice of choosing a custom insertion heuristic method in most cases, find a better solution. Hence, the algorithm can be used to solve VRPTWSD and applied to other variants of VRP in the near future.

TABLE II
AVERAGE DISTANCE TRAVELLED

Author	C1	R1	C2	R2	RC1	RC2
[40]	916.67 /10.00	1356.92 /13.00	644.63 /3.00	1276.00 /3.18	1514.29 /13.00	1634.43 /3.71
[41]	902.9 /10.00	1381.9 /13.33	653.2 /3.13	1293.4 /3.27	1545.3 /13.25	1595.1 /3.88
[42]	930 /10.00	1317 /12.66	681 /3.00	1167 /2.91	1523 /12.38	1398 /3.38
[43]	955.39 /10.00	1386.46 /12.83	717.31 /3.00	1366.48 /3.09	1545.92 /12.50	1598.06 /3.38
[44]	843.84 /10.00	1242.40 /13.50	607.58 /3.13	977.12 /4.09	1408.76 /13.50	1111.37 /5.13
[45]	828.38 /10.00	1205.06 /12.31	—-	—-	1360.40 /12.00	—-
[46]	828.38 /10.00	1201.79 /12.33	—-	—-	1364.17 /11.95	—-
[47]	828.38 /10.00	1233.34 /12.42	596.63 /3.00	990.99 /3.09	1403.74 /12.00	1220.99 /3.38
[48]	828.38 /10.00	1211.53 /12.08	589.86 /3.00	949.27 /2.82	1361.76 /11.88	1097.63 /3.38
[49]	834.05 /10.00	1241.89 /12.50	591.78 /3.00	995.39 /2.91	1408.87 /12.38	1139.70 /3.38
[50]	832.88 /10.00	1253.24 /12.17	593.49 /3.00	1039.56 /2.82	1408.44 /11.88	1244.96 /3.25
[51]	828.9 /10.00	1242.7 /12.8	589.86 /3.00	1016.4 /3.00	1412 /13.00	1201.2 /3.7
[52]	828.38 /10.00	1221.1 /11.92	589.93 /3.00	975.43 /2.73	1389.89 /11.5	1159.37 /3.25
[53]	828.38 /10.00	1224 /11.92	590.9 /3.00	1012 /2.73	1417 /11.5	1195 /3.25
[39]	828.48 /10.00	1220.92 /12.5	590.6 /3.00	938.75 /3.1	1386.35 /12.12	1132.12 /3.38
[54]	828.38 /10.00	1187.32 /13.08	591.74 /3.00	897.95 /4.00	1348.22 /12.63	1036.65 /5.63
[55]	828.38 /10.00	1213.66 /12.08	589.86 /3.00	961.44 /2.73	1370.01 /11.75	1126.75 /3.25
[7]	839.28 /10.01	1192.76 /12.78	601.29 /3.05	916.62 /4.72	1362.02 /12.50	1054.60 /5.82
Modified Genetic Algorithm	**822.79** /10.00	**793.41** /8.17	**407.63** /3.00	**691.47** /2.00	**1134.85** /9.75	**842.59** /2.00

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," Management Science, Vol. 6, No. 1, 1959, pp. 80-91. doi:10.1287/mnsc.6.1.80

[2] Ibrahim, A., Ishaya, J., Lo, N., & Abdulaziz, R., Capacitated Vehicle routing problem with column generation and reinforcement learning techniques. Open J. Discret. Appl. Math. 2020, 3(1), 41-54; doi:10.30538/psrp-odam2020.0029

[3] M. Dror and P. Trudeau, "Savings by Split Delivery Routing", Transportation Science, vol. 23, no. 2, pp. 141-145, 1989.

[4] Ishaya, J., Ibrahim, A., & Lo, N. (2019), A Comparative Analysis of the Travelling Salesman Problem: Exact and Machine Learning Techniques. Open Journal of Discrete Applied mathematics, 2(3), 23-37.

[5] Clingerman, C., Hemphill, J., & Proscia, C. (2008). Analysis and Counter Examples Regarding Yatsenko's Polynomial-Time Algorithm for Solving the Traveling Salesman Problem. arXiv preprint arXiv:0801.0474.

[6] Augerat, P., Belenguer, J. M., Benavent, E., Corberan, A., & Naddef, D. (1998). Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research, 106(2-3), 546-557.

[7] Xu, S.-H., Liu, J. P., Zhang, F. H., Wang, L., and Sun, L. J. A Combination of Genetic Algorithm and Particles Warm Optimization for Vehicle Routing Problem with Time Windows. Sensors, 15:21033–21053, 08 2015. doi:10.3390/s150921033.

[8] Desaulniers, G., & Groupe d'etudes et de Recherche en Analyse des decisions (Montreeal, Quebec). (2000).

[9] De Magalhaes, J. M., & De Sousa, J. P. (2006). Dynamic VRP in Pharmaceutical Distribution-a Case Study. Central European Journal of Operations Research, 14(2), 177-192.

[10] Adebayo, K. J., Aderibigbe, F. M. and Dele-Rotimi, A. O., (2019), On Vehicle Routing Problems (VRP) with a Focus on Multiple Priorities. American Journal of Computational Mathematics, 9, 348-357.

[11] F. M. Aderibigbe, K. J. Adebayo, (2021), On Dynamical Situations in Vehicle Routing Problems (DSVRP), American Journal of Traffic and transportation Engineering, 6(1): 1-9. doi: 10.11648/j.ajtte.20210601.11.

[12] C. Archetti and M. G. Speranza, "Vehicle Routing Problems with Split Deliveries," International Transactions in Operational Research, vol. 19, pp. 3-22, 2012.

[13] G. Sierksma and G. A. Tijssen, "Routing Helicopters for Crew Exchanges on Off-Shore Locations," Annals of Operations Research, Vol. 76, 1998, pp. 261-286. doi:10.1023/A:1018900705946

[14] S. Song, K. Lee and G. Kim, "A Practical Approach to Solving a Newspaper Logistics Problem Using a Digital Map," Computers and Industrial Engineering, Vol. 43, No. 1-2, 2002, pp. 315-330. doi:10.1016/S0360-8352(02)00077-3

[15] G. B. Alvarenga, G. R. Mateus and G. de Tomi, "A Genetic and Set Partitioning Two-Phase Approach for the Vehicle Routing Problem with Time Windows," Computers & Operations Research, Vol. 34, No. 6, 2007, pp. 1561-1584. doi:10.1016/j.cor. 2005.07.025.

[16] P. Mullaseril, M. Dror and J. Leung, "Split-Delivery Routing Heuristics in Livestock Feed Distribution," Journal of the Operational Research Society, Vol. 48, 1997, pp. 107-116.

[17] Ursani, Z., Essam, D., Cornforth, D., & Stocker, R. (2011). Localized genetic algorithm for vehicle routing problem with time windows. Applied Soft Computing, 11(8), 5375-5390.

[18] Bortfeldt, A., & Yi, J. (2020). The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints. European Journal of Operational Research, 282(2), 545-558.

[19] Masum, A. K. M., Shahjalal, M., Faruque, F., & Sarker, I. H. (2011). Solving the Vehicle Routing Problem using Genetic Algorithm. International Journal of Advanced Computer Science and Applications, 2(7), 126-131.

[20] Nazif, H., & Lee, L. S. (2012). Optimised Crossover Genetic Algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling, 36(5), 2110-2117.

[21] Bae, H., & Moon, I. (2016). Multi-depot Vehicle Routing Problem with Time Windows Considering Delivery and Installation Vehicles, Applied Mathematical Modelling, 40(13-14), 6536-6549.

[22] Cooray, P. L. N. U., & Rupasinghe, T. D. (2017). Machine Learning Based Parameter Tuned Genetic Algorithm for Energy Minimizing Vehicle Routing Problem, Journal of Industrial Engineering.

[23] Min, J., Jin, C., & Lu, L. (2018), A Three-stage Approach for Split Delivery Vehicle Routing Problem Solving. In 2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS) (pp. 1-6). IEEE.

[24] Stodola, P. (2020). Hybrid Ant Colony Optimization Algorithm Applied to the Multi-depot Vehicle Routing Problem. Natural Computing, 1-13.

[25] Silva, M. M., Subramanian, A., & Ochi, L. S. (2015). An Iterated Local Search Heuristic for the Split Delivery Vehicle Routing Problem. Computers & Operations Research, 53, 234-249.

[26] Rajappa, G. P., Wilck, J. H., & Bell, J. E. (2016). An Ant Colony Optimization and Hybrid Metaheuristics Algorithm to Solve the Split Delivery Vehicle Routing Problem. International Journal of Applied Industrial Engineering (IJAIE), 3(1), 55-73.

[27] Tasan, A. S., & Gen, M. (2012). A Genetic Algorithm-based Approach to Vehicle Routing Problem with Simultaneous Pick-up and Deliveries. Computers & Industrial Engineering, 62(3), 755-761.

[28] Wen, M., (2010), Rich Vehicle Routing Problems and Applications. Kgs. Lyngby: DTU Management. Ph. D. Thesis, No. 8.

[29] Cordeau, J. F., Desaulnier, J. F., Desrosiers, S, J. Solomon, M. M. and Soumis, F., (2002), VRP with Time Windows, in the Vehicle Routing Problem, P. Toth and D. Vigo, eds., SIAM, Philadelphia, ch. 7, pp. 155–194.

[30] Russell, R. A. and Chiang, W. C., (2006), Scatter Search for the Vehicle Routing Problem with Time Windows, European Journal of Operational Research, 169(2): 606–622.
[31] Miller, D. L., Tucker, A. W. and Zemlin, R. A., (1960), Integer Programming Formulations of Traveling Salesman Problems, Journal of the ACM, 7 (1960), pp. 326–329.
[32] P. Damodaran, N. S. Hirani and M. C. V elez-Gallego, "Scheduling Identical Parallel Batch Processing Machines to Minimise Make Span Using Genetic Algorithms," European Journal of Industrial Engineering, Vol. 3, No. 2, 2009, pp. 187-206. doi:10.1504/EJIE.2009.023605
[33] Akhand, M. A. H., Zahrul Jannat Peya, Tanzima Sultana and Hafizur Rahman, M. M., (2017), Solving Capacitated Vehicle Routing Problem Using Variant Sweep and Swarm Intelligence, Journal of Applied Science and Engineering, Vol. 20, No. 4, pp. 511-524.
[34] Stefan I., Paolo T. and Daniele V., (2014), The Family of Vehicle Routing Problems, Chapter 1,
[35] Dror, M. and Trudeau, P., (1989), Savings by Split Delivery Routing, Transportation Science, 23, pp. 141–145.
[36] Dror, M. and Trudeau, P., (1990), Split Delivery Routing, Naval Research Logistic Quarterly, 37, pp. 382–402.
[37] Darwin, C. The Origin of Species by Means of Natural Selection. Academic Medicine, 26:243, 05 1951.
[38] J. Wilck IV and T. Cavalier, "A Genetic Algorithm for the Split Delivery Vehicle Routing Problem," American Journal of Operations Research, Vol. 2 No. 2, 2012, pp. 207-216. doi: 10.4236/ajor.2012.22024.
[39] Ombuki, B., Ross, B., and Hanshar, F. Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. Applied Intelligence, 24:17–30, 02 2006. doi:10.1007/s10489-006-6926-z.
[40] Thompson, P. and Psaraftis, H. Cyclic Transfer Algorithm for Multivehicle Routing and Scheduling Problems. Operations Research, 41:935–946, 10 1993. doi:10.1287/opre.41.5.935.
[41] Potvin, J. Y. and Rousseau, J. M. An Exchange Heuristic for Routeing Problems with Time Windows. Journal of The Operational Research Society - J OPER RES SOC, 46:1433–1446, 12 1995. doi:10.1038/sj/jors/0461203
[42] Russell, R. Hybrid Heuristics for the Vehicle Routing Problem with Time Windows. Transportation Science, 29:156–166, 05 1995. doi:10.1287/trsc.29.2.156.
[43] Antes, J. and Derigs, U. A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows. 12 1997.
[44] Prosser, P. and Shaw, P. Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems. 02 1997.
[45] Shaw, P. A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems. 09 1997.
[46] Shaw, P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, pages 417–431, 011998.
[47] Caseau, Y. and Laburthe, F. Heuristics for Large Constrained Vehicle Routing Problems. J. Heuristics, 5:281–303,10 1999. doi:10.1023/A:1009661600931.
[48] Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G. Record Breaking Optimization Results using Theruin and Recreate Principle. Journal of Computational Physics, 159:139–171, 04 2000. doi:10.1006/jcph.1999.6413.
[49] Cordone, R. and Calvo, R. A Heuristic for the Vehicle Routing Problem with Time Windows. Journal of Heuristics,7:107– , 03 2001. doi:10.1023/A:1011301019184
[50] Braysy, O. Fast local searches for the Vehicle Routing Problem with Time Windows. INFOR Information Systems and Operational Research, 40:319–330, 11 2002.
[51] Zhu, K. A Diversity-controlling Adaptive Genetic Algorithm for the Vehicle Routing Problem with Time Windows. pages 176– 183, 12 2003. ISBN 0-7695-2038-3. doi:10.1109/TAI.2003.1250187.
[52] Berger, J. and Barkaoui, M., A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. Computer and Operations Research, 31:2037–2053, 10 2004. doi:10.1016/S0305-0548(03)001631.
[53] Alvarenga, G. B., de Abreu Silva, R. M., and Sampaio, R. M. A Hybrid Algorithm for the Vehicle Routing Problem with Time Window. 2005.
[54] Garcıa-Najera, A. and Bullinaria, J. A. An Improved Multi-objectives Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows. Comput. Oper. Res., 38(1):287–300, 2011. doi:10.1016/j.cor.2010.05.004.
[55] VAIRA, G. and Kurasova, O. Genetic Algorithm for VRP with Constraints Based on Feasible Insertion. Informatica, 25: 155–184,032014. doi: 10.15388/Informatica.2014.09.