Genetic Algorithm and Local Search applied to Vehicle Routing Problem

Pedro de Mendonça Maia

***Abstract*— The vehicle routing problem (VRP) is a combinatorial optimization problem with many applications in logistics areas. Its goal is to determine routes for the available vehicles to attend the demands of a set of customers with minimum cost. Exact models do not perform well on large instances of the VRP because of the problem's high complexity and heuristics are recommended. In this paper, a Genetic Algorithm combined with Local Search procedures is used solve the VRP. Experiments are performed on a traditional data set to evaluate the proposed algorithm.**

I. Introduction

The *Vehicle Routing Problem* (VRP) consists of determining optimal routes for fleet of vehicles to serve geographically scattered customers. This problem can be defined as a graph $G = (V, E)$, where $V = v_0, ..., v_n$ are the vertices and $E = (v_i, v_j) : v_i, v_j \in V$ is the set of edges. Each edge has an associated time $t_{i,j}$ corresponding to the travel time from v_i to v_j. v_0 represents the depot where there is a fleet of k vehicles located. The other vertices are the customers with specific demands d_i. A solution of the VRP consists of k routes starting and ending at the depot where each customer is visited once by exactly one vehicle. An example of solution for a VRP problem with five vehicles is in Figure 1.

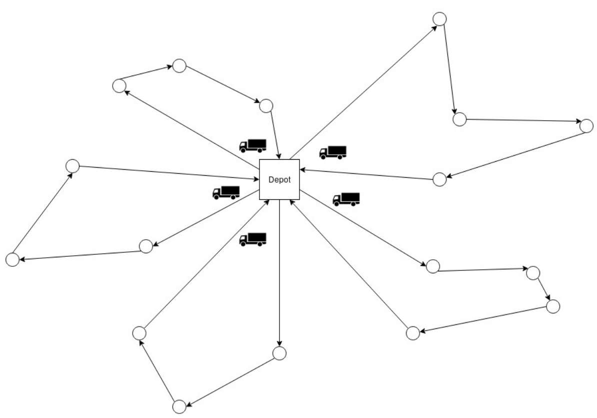

Fig. 1. Example of VRP solution with 5 vehicles and 17 customers

VRP objective is to find the routes with minimum total travel time subject to a variety of constraints. The most common constraint of the VRP is to consider a capacity Q for each vehicle. This problem is called *Capacitated Vehicle Routing Problem* (CVRP) and the sum of demands of the customers served by each vehicle cannot exceed its capacity. Total traveled distance or cost can be used instead of travel time.

The VRP can be used to solve a variety of logistic problem with different sizes and characteristics. However, the possible solutions of the VRP increase exponentially as the number of customers and/or vehicles grow. Exact optimization algorithms take a long time to achieve the optimal solution in large real world problems, but solution time is relevant in most of these cases, so heuristic approaches a commonly used to find good solutions in reasonable time.

In this paper a *Genetic Algorithm* (GA) is proposed to solve the VRP. In order to increase convergence, local searches are applied to each route every few iterations. The Solomon data set ([1]) is used to validate the model.

II. Related Work

The initial version of Vehicle Routing Problem was introduced by [2] as the "Truck Dispatching Problem" in 1959 as an extension of the well known *Traveling Salesman Problem* (TSP). They showed the exponential combinatory characteristics of the problem and presented a method to find the optimal solution. [3] generalized the problem and presented what can be considered the first VRP in the literature as it is used today.

Many authors identified the applicability of the VRP in many real world problems and it gained a lot of attention in the literature since then. [4] present the number of papers on routing problems in the literature and shows a continuous growth since 1970s. Initially the authors presented exact methods to solve the VRP ([5], [6]), but the computational time was high and some applications required fast solutions, so heuristics methods appeared ([7], [8], [9]). The increasing processing capacity of the computers since 1990s intensified the development of heuristics to find acceptable solutions of the VRP in shorter time ([10], [11], [12]).

Heuristics currently dominate the VRP literature. [13] reviewed methods used on the VRP between 2009 and 2016 and according to their survey less than 20% of the papers used exact methods. Evolutionary algorithms([14], [15], [16]) and metaheuristcs ([17], [18], [19]).

However, different methods to solve the VRP could not overcome all the limitations identified by some applications. In specific cases, some change in the definition of the VRP became necessary and new variations of the problem emerged. Models with heterogeneous vehicle capacities ([20], [21], [22]), multi-depots ([23], [24], [25]), delivery and pick-ups ([26], [27], [28]) and others were proposed (check [13] for more information).

III. Methodology

This paper uses a Genetic Algorithm to solve the traditional CVRP with vehicle's maximum work hour. Evolutionary algorithms alone usually are not able to find good solutions for combinatorial problems, so during the GA execution some local search methods are executed to improve convergence.

The solution model, local search procedures and GA workflow are explained in the following sections.

A. VRP Model

The typical representation of the VRP considers only one permutation vector, where the first customers are assigned to the first vehicle until the vehicle capacity or route time is reached. Then the next customers are assigned to the second vehicle and so on, until all customers are assigned to some vehicle. This representation is interesting because the number of necessary vehicles are calculated only after the optimization. However, it does not allow unbalanced routes that might be in the optimal solution. For this reason, in this paper another representation is proposed.

The VRP solution is considers a list of vehicles. Each vehicle is composed by the sequence of customers it visits. Every customer is assigned to exactly one vehicle. All the vehicles start and end the route at the depot. In the example of Figure 2, the vehicle $V01$ leaves the depot, visits customers $C07$, $C02$, $C11$, $C01$ in this order and returns to the depot.

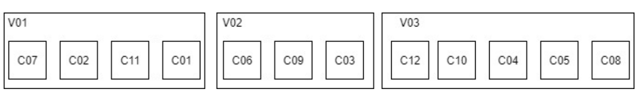

Fig. 2. Example Solution with three vehicles

B. Genetic Algorithm

The genetic algorithm is an evolutionary algorithm based on population. The population is composed by a set of solutions (individuals). A selection method is use to choose individuals to make crossover and create the next generation. Individuals with lower route times (most fitted) have higher probability of selection. Every new individual has a probability p_m to suffer mutation.

The general diagram of the algorithm is in Figure 3.

Every generation has a population of n individuals, so every iteration has to generate n individuals. The selection method chooses a pair of parents to generate a pair of children with probability p_c using the crossover method. If the crossover does not happen, the pair of parents is passed to the next generation.

The algorithm steps adopted in this paper are described in the following sections.

1) Initial Population: Individuals are created randomly. A vector of a random permutation of customers is generated and balanced sub-vectors are assigned to each vehicle.

2) Fitness calculation: Solution cost is given by the sum of total travel times and service times of each vehicle. Whenever a capacity or driver work hours of any vehicle is exceeded, a penalty is added to the solution cost.

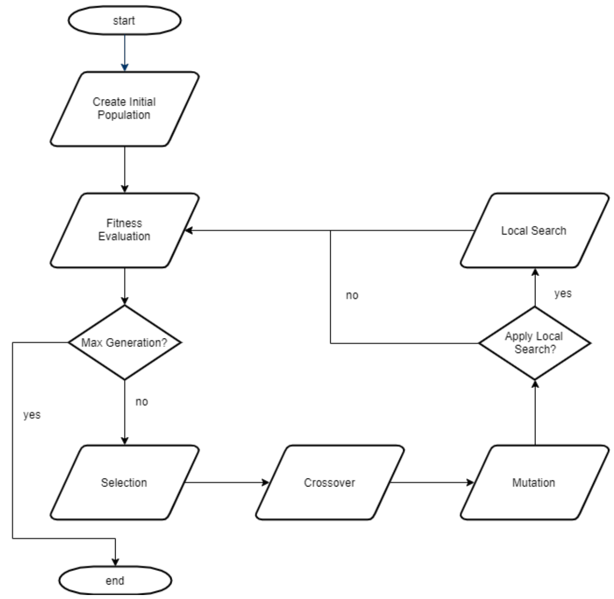

Fig. 3. Algorithm Diagram

3) Selection: Two methods of selection are used:

- **Tournament:** Two individuals are picked randomly and the most fitted is selected;
- **Roulette Wheel:** One individual from the population is selected with probability proportional to the fitness.

These methods are chosen randomly to select each parent used in each crossover.

4) Crossover: The crossover method used is the Best Cost Route Crossover (BCRC) [29]. It is a method for solutions modeled as permutation. An adaptation in the solution model is necessary to use the BCRC. The number of customers served by each vehicle are stored and all the customers are concatenated into a single vector, preserving the order. The BCRC is then executed and at the end the customers of the solution vector are assigned back to the vehicles according to the stored quantities.

The BCRC step are as follow:

- Randomly select two cut points for each parent:
  P_1: 3-1-7-—5-6—-4-2-8-9
  P_2: 1-9-8-—2-4-6-5—-7-3
- Randomly choose one sub-vector from each parent and remove those customers from the other parent. Suppose the customers 5 and 6 are selected from P_1 to remove from P_2 and customers 7 and 3 from P_2 to remove from P_1.
  P_1: 1-—5-6—-4-2-8-9
  P_2: 1-9-8-—2-4—-7-3
- Each removed customer is re-inserted in the best position
  P_1: 1-5-6-4-2-8-3-7-9
  P_2: 1-5-9-8-2-4-7-3-6

5) Mutation: Three methods of mutation are used:

- **Multi-Shift:** The following process is repeated two times: One random customer is transferred from a random route to another route. In the destination it is inserted in the best position;

- **Multi-Swap:** The following process is repeated two times: Two random routes are selected and one random customer from each is transferred to the other. In the destination the customers are inserted in the best positions;
- **Ejection Chain:** One random customer is transferred from each route to the next route. In the destination the customers are inserted in the best positions.

C. Local Search Methods

Intra-route local search algorithms are used to improve convergence on some generations of the algorithm, because crossover and mutation operators usually can't find the best sequence of customers in a route. A local search procedure is randomly selected to run on each route of each individual. The following local search procedures can be used:

- **Or-Opt:** Removes a customer from the route and re-insert in the best position;
- **Or-Opt2:** Removes two consecutive customers from the route and re-insert the sub-sequence in the best position;
- **Or-Opt3:** Removes two consecutive customers from the route and re-insert the sub-sequence in the best position;
- **2-Opt:** Selects two customers of the route and reverse their order in the route;
- **Intra-Swap:** Swaps the position of two customers in the route.

The *Best Improvement* version of each algorithm is used. For each route, all possible movements in neighborhood of the chosen method are tested and the best is stored. The local search ends when there is no improvement.

IV. Experimental Evaluation

The computational experiments were executed on the Solomon benchmark problems [1]. The data set is composed by 56 instance with 100 customers.

The problems are divided into six classes: clustered customers with tight schedule (C1 problems), clustered customers with loose schedule (C2 problems), randomized customers with tight schedule (R1 problems), randomized customers with loose schedule (R2 problems), mixed clustered and random customers with tight schedule (RC1 problems) and mixed clustered and random customers with loose schedule (RC2 problems).

The difference between the problems within each class are the time windows, but this constraint is not considered in this paper, so tests were executed on the first problem of each class.

In this paper, the first 50 customers are considered, and 5 vehicles are used for all the problems.

In the tests, the parameters considered in the algorithm are:

- Crossover probability: $p_c = 0.8$
- Mutation probability: $p_m = 0.3$
- Population size: $n = 100$
- Number of generations: 100
- Number of generations with Local Search: 10

The algorithm was executed 30 times on each problem to evaluate the performance.

V. Results

Figures 4, 5, 6, 7, 8, 9 show solutions obtained by the algorithm for each problem. Visual inspection shows good results for most problems. Only in a few cases some parts of some routes could be manually improved.

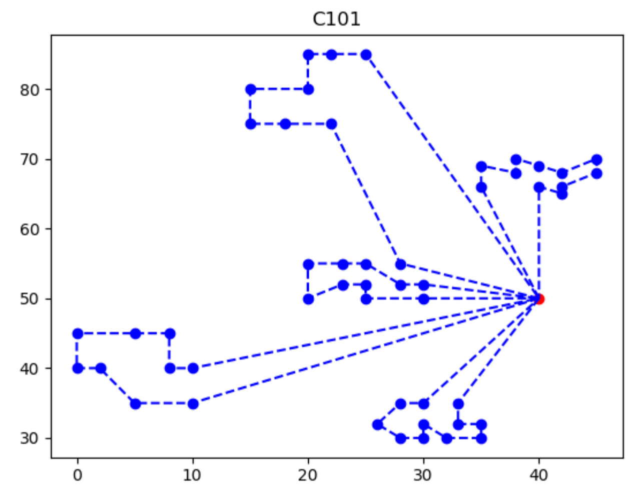

Fig. 4. Result problem C101

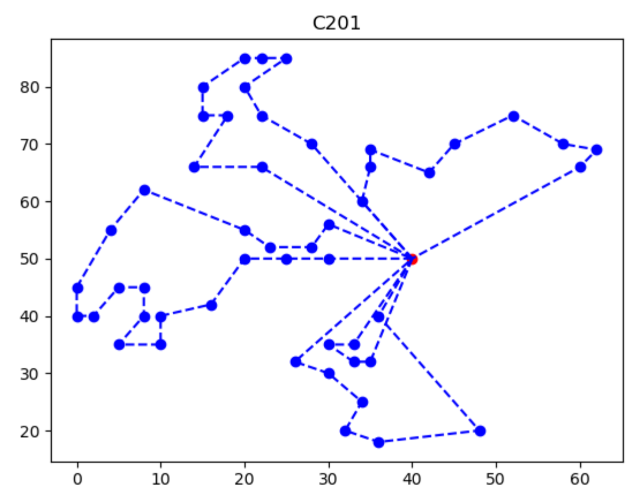

Fig. 5. Result problem C201

Figures 10, 11, 12 show how the results obtained for each problem varied. All classes of problems had low variance, what indicates a consistency of the algorithm.

An interesting result is that the same solution was found for all the mixed problems (RC101 and RC201) and by visual inspection of Figures 8, 9 one can see this is the best solution possible.

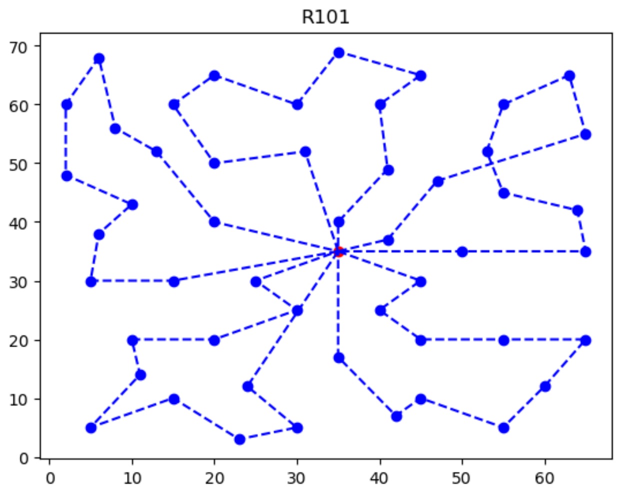

Fig. 6. Result problem R101

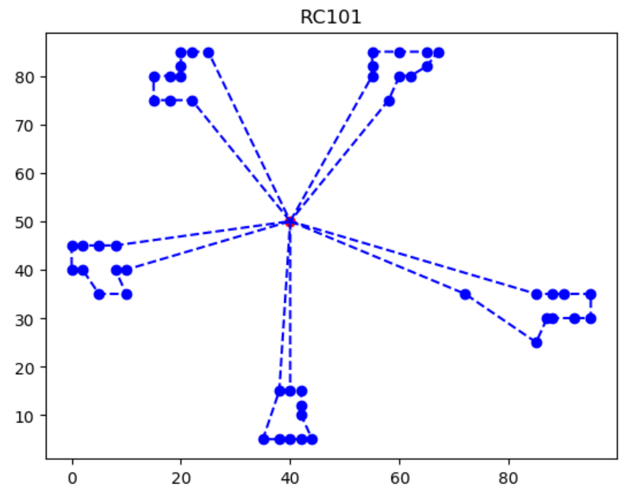

Fig. 8. Result problem RC101

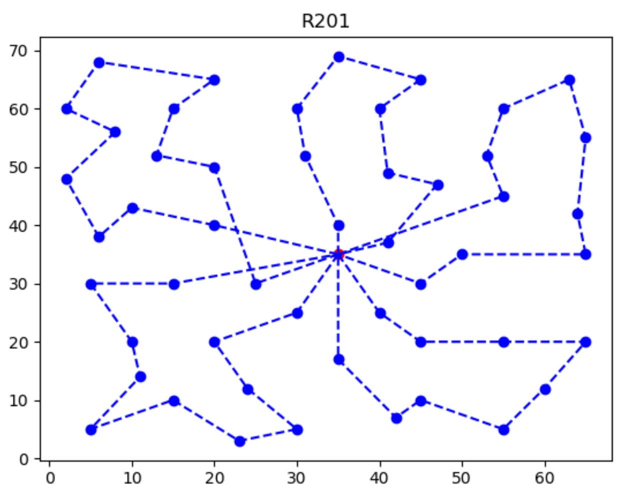

Fig. 7. Result problem R201

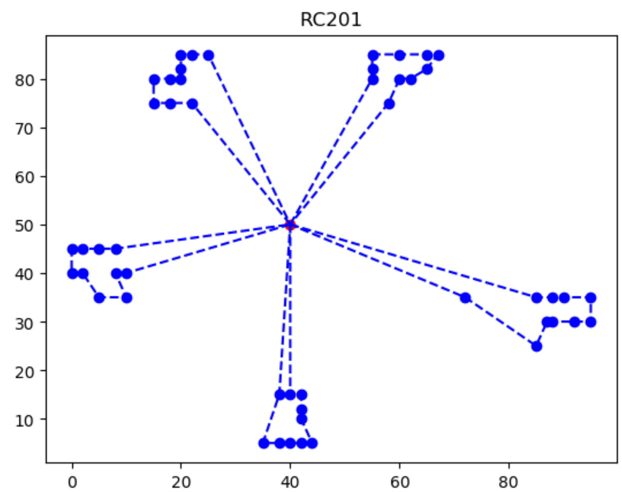

Fig. 9. Result problem RC201

Problem	Mean Distance	Execution Time (s)
C101	366.90	138.64
C201	461.65	138.47
R101	603.98	139.92
R201	586.84	144.11
RC101	518.50	135.11
RC201	518.50	139.81

TABLE I
MÉDIA DOS RESULTADOS OBTIDOS PARA CADA PROBLEMA

The results from Figures 4, 5, 6, 7, 8, 9 combined with the low variability in results from Figures 10, 11, 12 show the algorithm is consistent and able to find good results.

Table I show the mean distances and execution time for each problem. As expected, all execution times are similar, because all the problems have same dimension and the algorithm performs approximately the same steps on each case.

VI. Conclusions

This paper presented a combination of Genetic Algorithm with Local Search procedures to solve the VRP problem. The VRP is a challenging optimization problem in the literature with many applications in real world problems.

Evolutionary algorithms alone do not perform well on combinatorial problems like the VRP, so local search methods were introduced to improve the convergence. The proposed method was applied in a famous base of the literature and satisfactory results were obtained.

In future works, more constraints can be added to the problem, like time windows. Furthermore, the number of

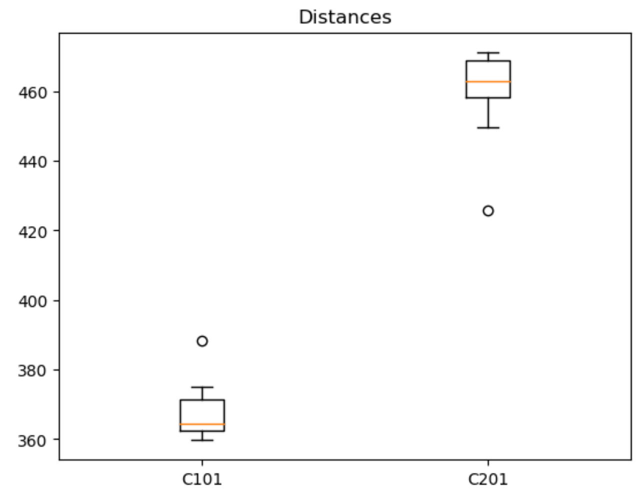

Fig. 10. Clustered problems result dispersion

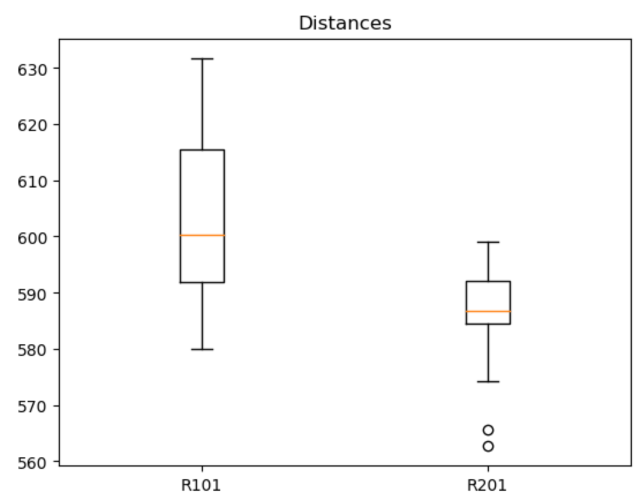

Fig. 11. Random problems result dispersion

vehicles might be reduced dynamically, adding some mutation structure to remove a vehicle. Other VRP variations can be tested as well, like stochastic, dynamic and multi-depots.

References

[1] M. M. Solomon, "Vehicle routing and scheduling with time window constraints: Models and algorithms," tech. rep., 1984.

[2] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[3] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research*, vol. 12, no. 4, pp. 568–581, 1964.

[4] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers and Industrial Engineering*, vol. 57, no. 4, pp. 1472–1483, 2009.

[5] D. Marks and R. Stricker, "Routing for Public Service Vehicles," *SCE Journal of the Urban Planning and Development Division*, vol. 97, pp. 165–178, 1971.

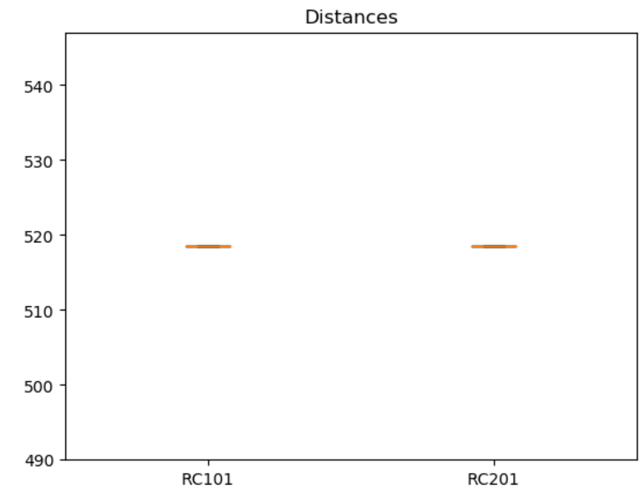

Fig. 12. Mixed problems result dispersion

[6] A. Levin, "Scheduling and Fleet Routing Models for Transportation Systems," *Transportation Science*, vol. 5, no. 3, pp. 232–255, 1971.

[7] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen, "Implementing vehicle routing algorithms," *Networks*, vol. 7, no. 2, pp. 113–148, 1977.

[8] T. M. Cook and R. A. Russell, "a Simulation and Statistical Analysis of Stochastic Vehicle Routing With Timing Constraints," *Decision Sciences*, vol. 9, no. 4, pp. 673–687, 1978.

[9] J.-J. Jaw, A. R. Odoni, H. N. Psaraftis, and N. H. M. Wilson, "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows," *Time*, vol. 2, no. 3, pp. 243–257, 1986.

[10] P. Trudeau and M. Dror, "Stochastic inventory routing: Route design with stockouts and route failures," *Transportation Science*, vol. 26, no. 3, pp. 171–184, 1992.

[11] R. W. Eglese, "Routeing winter gritting vehicles," *Discrete Applied Mathematics*, vol. 48, no. 3, pp. 231–244, 1994.

[12] J. Brandão and A. Mercer, "Tabu Search Algorithm for The Dynamic Vehicle Routing Problem," *European Journal of Operational Research*, vol. 100, no. 97, pp. 180–191, 1997.

[13] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers and Industrial Engineering*, vol. 99, pp. 300–313, 2016.

[14] M. Barkaoui and M. Gendreau, "An adaptive evolutionary approach for real-time vehicle routing and dispatching," *Computers and Operations Research*, vol. 40, no. 7, pp. 1766–1776, 2013.

[15] A. Garcia-Najera and J. A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," *Computers and Operations Research*, vol. 38, no. 1, pp. 287–300, 2011.

[16] T. C. Chiang and W. H. Hsu, "A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows," *Computers and Operations Research*, vol. 45, pp. 25–37, 2014.

[17] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Computers and Operations Research*, vol. 36, no. 12, pp. 3215–3223, 2009.

[18] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the Pollution-Routing Problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 346–359, 2012.

[19] Y. Marinakis, "Multiple phase neighborhood search-GRASP for the capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6807–6815, 2012.

[20] J. Brandão, "A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem," *Computers and Operations Research*, vol. 38, no. 1, pp. 140–151, 2011.

[21] S. Liu, "A hybrid population heuristic for the heterogeneous vehicle routing problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 54, pp. 67–78, 2013.

[22] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems," *European Journal of Operational Research*, vol. 234, no. 3, pp. 658–673, 2014.

[23] B. Yu, Z. Z. Yang, and J. X. Xie, "A parallel improved ant colony optimization for multi-depot vehicle routing problem," *Journal of the Operational Research Society*, vol. 62, no. 1, pp. 183–188, 2011.

[24] S. Salhi, A. Imran, and N. A. Wassan, "The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation," *Computers and Operations Research*, vol. 52, pp. 315–325, 2014.

[25] W. Tu, Z. Fang, Q. Li, S. L. Shaw, and B. Y. Chen, "A bi-level Voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 61, pp. 84–97, 2014.

[26] D. G. Pandelis, C. C. Karamatsoukis, and E. G. Kyriakidis, "Finite and infinite-horizon single vehicle routing problems with a predefined customer sequence and pickup and delivery," *European Journal of Operational Research*, vol. 231, no. 3, pp. 577–586, 2013.

[27] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows," *Computers and Industrial Engineering*, vol. 83, pp. 111–122, 2015.

[28] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "The load-dependent vehicle routing problem and its pick-up and delivery extension," *Transportation Research Part B: Methodological*, vol. 71, pp. 158–181, 2015.

[29] S. Kumar Nagle, "Study of Crossover operators of Genetic Algorithm& Development of New Crossover Operator to Solve Heterogeneous Vehicle Routing Problem with Time Windows," Tech. Rep. 2.