A Genetic Algorithm Based Approach for Vehicle Routing Problem with Simultaneous Delivery and Pick-up and Time Windows in Home Health Care

Habibeh Nazif
Department of Mathematics, Payame Noor University, IRAN
h_nazif@pnu.ac.ir

Abstract
This paper proposes a genetic algorithm based approach for solving a special vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. It concerns the delivery of drugs and medical devices from the home care company’s pharmacy to patients’ homes, delivery of special drugs from a hospital to patients, pickup of bio samples and unused drugs and medical devices from patients. Each patient is visited by one vehicle and each vehicle visits each node at most once. Patients are associated with time windows and vehicles with capacity. The problem can be considered as a special vehicle routing problem with simultaneous delivery and pickup and time windows. Computational example is presented with parameter settings in order to illustrate the proposed approach. Moreover, the proposed approach is tested on test instances derived from existing VRPTW benchmarks. From the results, it can be concluded that the proposed algorithm is competitive when compared with the best-known solutions in the literature.

Keywords: Genetic algorithms, Home health care logistics, Vehicle routing, Pickup and delivery, Time windows.

1. Introduction

This paper addresses a vehicle scheduling problem encountered in home health care logistics. Home hospitalization organizations have been created for patients requiring long and regular health cares in order to provide quality health service at their home while reducing the bed requirements at hospitals. Home health care services are provided in France by Home Health Care (HHC) companies. Each day, a HHC company has various logistic activities including delivering drugs and medical devices from its pharmacy (also called depot in this paper) to patients at their home. It also takes some special drugs, such as chemotherapy drugs and blood products, from hospitals to patients. On the other hand, the HHC also needs to pick up materials from patients and deliver to different locations. Blood samples of the patients are collected and delivered to a medical lab. Medical wastes, unused drugs and medical devices are collected and brought back to the HHC or the depot. As HHC companies are usually small but serve rather large number of patients with dispersed locations, it is crucial to carefully design the routes of the HHC vehicles in order to reduce its operating cost while improving the service quality to patients.

The design of HHC vehicle routes is related to the vehicle routing problem with simultaneous pickup and delivery and time windows (VRPSDPTW) introduced by Hokey (1989). The VRPSDPTW is a hard and challenging problem in the field of vehicle routing problem (VRP). In the classical VRPSDPTW all delivery goods are loaded at the depot and all pickup goods have to be transported to the depot. In our HHC vehicle scheduling problem besides the depot, goods can be transported from a hospital to patients and from the patients to a lab. Clearly, the composition of vehicles’ loads in our case is more complex than VRPSDPTW. Furthermore, different from the classical VRPSDPTW, each route of our problem must satisfy some precedence constraints, e.g., for a patient who needs drugs provided by the hospital, the vehicle visiting the patient has to visit the hospital first. Such special constraints are similar to the pairing and precedence constraints in classical pickup and delivery problem (PDP), in which each customer request is defined by an origin location and a destination, the origin must visited before the destination by the same vehicle. However, the PDP is less complicated than our problem, since the origins as well as the destinations of transportation requests in the PDP are locations other than the depot, and a customer in the PDP only has either pickup or delivery request. Since both the VRPSDPTW and PDP are NP-hard problems, our problem is more complex than these problems and is also NP-hard.
The VRPSDPTW is an extension of the VRPSDP, and has been much less studied than the VRPSDP. One exact method for the VRPSDP was designed by Dell’Amico et al. (2006). Dethloff (2002) proposes an extension of the cheapest insertion heuristic to the VRPSDP. Several tabu search (TS) algorithms for VRPSDP were proposed in Alfredo Tang Montané and Galvão (2006), Chen and Wu (2005), Bianchessi and Righini (2007) and Crispim and Brandão (2005). Recently, Ai and Kachitvichyanukul (2009), Gajpal and Abad (2009) and Subramanian et al. (2010) have proposed several metaheuristics to solve VRPSDP. For the most complex one, VRPSDPTW, only Angelelli and Mansini (2002) proposed an exact method and Mingyong and Erbao (2010) and Wang and Chen (2012) proposed genetic algorithms (GAs). For the the pickup and delivery problem with time windows (PDPTW), several exact approaches have been designed. Dumas et al. (1991), Savelsbergh and Sol (1998), Xu et al. (2003) and Sigurd and Pisinger (2004) used branch and price schemes for the PDPTW. Cordeau (2006) and Ropke et al. (2007) developed branch and cut approach for the PDPTW. Ropke and Cordeau (2009) introduced a new branch and cut and price algorithm for the PDPTW. Meanwhile, many heuristics have been proposed for the PDPTW. Jaw et al. (1986), Madsen et al. (1995), Diana and Dessouky (2004) and Lu and Dessouky (2006) presented various insertion-based heuristics for solving the PDPTW. Toth and Vigo (1997), Nanry and Wesley Barnes (2000) and Cordeau and Laporte (2003) solved the PDPTW by means of tabu search heuristics. Li and Lim (2001), Pankratz (2005), Ropke and Pisinger (2006) and Parragh et al. (2010) designed simulated annealing, genetic algorithm, adaptive large neighborhood search heuristic, and variable neighborhood search heuristic for solving the PDPTW. For the HHC vehicle scheduling problem, only Liu et al. (2013) proposed two meta-heuristics, tabu search and genetic algorithm. The GA is based on a permutation chromosome, a split procedure and local search. The TS is based on route assignment attributes of patients, an augmented cost function, route re-optimization, and attribute based aspiration levels. They tested their approaches by using a range of test instances, which are designed based on existing VRPTW benchmarks to reflect different realistic scenarios. The results showed that, both TS and GA provide good solutions in a reasonable time span, and TS requires relatively more computational time. In this paper, we first formulate our problem as a mixed integer programming model (MIP) based on model given by Liu et al. (2013), and then develop a genetic algorithm (GA) based approach, which uses permutation based representation and ensures feasibility for solving the problem more efficiently and effectively. The performance of the proposed approach is evaluated with several test problems and the results are discussed in details.

2. Mathematical formulation

This paper addresses the daily scheduling problem of vehicles of a home health care company for delivery of drugs and medical devices and for pickup of biological samples and medical wastes or unused drugs. This section presents a mixed integer programming formulation of the problem that will serve to assess the efficiency of the proposed algorithm of this paper.

The problem can be defined as follows. Let \( G = (V, A) \) be a directed graph with a set \( V = \{0, 1, \ldots, n, n+1\} \cup \{h,m\} \) of nodes and a set \( A = \{(i,j): i, j \in V, i \neq j\} \) of arcs. Nodes 0 and \( n + 1 \) represent the origin and destination depots which are in practice the pharmacy of the home health care company. Each vehicle starts at node 0 and ends at node \( n + 1 \). Nodes \( N = \{1, \ldots, n\} \) correspond to patients’ homes. Node \( h \) and \( m \) represent the locations of a hospital and a medical lab. Each patient \( i \in N \) has four types of delivery and pickup requirements: \( d_{i1} \), \( d_{i2} \), \( p_{i1} \) and \( p_{i2} \), where \( d_{i1} \) represents the amount of materials (drugs/medical devices) to deliver from the depot 0 to patient \( i \), \( d_{i2} \) the amount of materials (special drugs) to deliver from the hospital to patient \( i \), \( p_{i1} \) the amount of materials to pick up from patient \( i \) and bring back to the depot \( n + 1 \), and \( p_{i2} \) the amount of materials (biological samples) to pick up from patient \( i \) and bring to the medical lab \( m \). Each type of requirements is called a demand. Different materials are assumed to be compatible and can be loaded in the same vehicle. \( D_1 \subseteq N \) denotes the set of patients needing type 1 delivery service, i.e. patients \( i \) with \( d_{i1} > 0 \). Similarly, \( D_2 \), \( P_1 \), \( P_2 \) denote sets of patients needing type 2 delivery, type 1 and type 2 pickup services. A patient may require different types of demands. For example, for a patient \( i \in D_2 \cap P_2 \), the company has to pick up the quantity \( p_{i2} \) from node \( i \) and deliver to the lab and deliver the quantity \( d_{i2} \) from the hospital to this node. For notation convenience, we set zero-demands for nodes 0, \( n + 1 \), \( h \), \( m \).
A time window \([a_i, b_i]\) is associated with each node \(i \in V\), where \(a_i\) and \(b_i\) represent the earliest and latest time. A vehicle is allowed to arrive before \(a_i\) and wait until the patient becomes available, but arrivals after \(b_i\) are prohibited. The depot node also has a time window, representing the earliest and latest times when the vehicles may leave from and return to the depot. Each arc \((i, j) \in A\) is associated with a routing cost \(c_{ij}\) and a travel time \(t_{ij}\).

The service time for a patient \(i\) is assumed to be included in the travel time \(t_{ij}\). A fleet \(K\) of identical vehicles, initially located at the depot, is available to serve the patients. Each vehicle has a capacity of \(Q\).

The problem consists in determining a set of at most \(K\) routes of minimal overall cost in order to serve all delivery and pickup demands of all patients under the obvious time window and vehicle capacity constraints and the following assumptions:

**Assumption A.** Each route starts from and ends at the depot and visits each location at most once;

**Assumption B.** Each patient is visited by exactly one vehicle for all its demands;

**Assumption C.** Each route makes a hospital visit before visits to its \(D_2\)-patients;

**Assumption D.** Each route makes a lab visit after all visits to its \(P_2\)-patients.

A typical route is as follows. The vehicle starts the depot with all materials for its \(D_1\)-patients, visits some patients for \(D_1\)-delivery and any pickup, visits the hospital to load all materials for its \(D_2\)-patients, visits other patients, then goes to the lab to deliver materials of all \(P_2\)-patients, visits other patients before returning to the depot with all materials of its \(P_1\)-patients. Four types of decision variables are used:

- \(x_{ij}^k\) binary variable equal to 1 if vehicle \(k\) travels directly from node \(i\) to node \(j\);
- \(B_{ij}^k\) time at which vehicle \(k\) begins to serve at node \(i\);
- \(y_{ij}^k\) quantity of \(P_2\)- and \(P_2\)-pickup carried along arc \((i, j)\) by vehicle \(k\);
- \(w_{ij}^k\) quantity of \(D_1\)- and \(D_2\)-delivery carried along arc \((i, j)\) by vehicle \(k\).

The MIP problem is formulated as follows:

\[
\begin{align*}
\text{Min} & \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ij}^k \\
\text{St.} & \quad \sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 & \forall i \in N \\
& \quad \sum_{j \in V} x_{ij}^k \leq \sum_{j \in V} x_{jm}^k & \forall i \in P_2, k \in K \\
& \quad \sum_{j \in V} x_{ij}^k \leq \sum_{j \in V} x_{lj}^k & \forall i \in D_2, k \in K \\
& \quad \sum_{i \in V} x_{0i}^k \leq 1 & \forall k \in K \\
& \quad \sum_{i \in V} x_{iN+1}^k \leq 1 & \forall k \in K \\
& \quad \sum_{j \in V} x_{ij}^k = \sum_{j \in V} x_{ji}^k & \forall i \in N \cup \{m, h\}, k \in K \\
& \quad \sum_{j \in V} x_{ij}^k \leq 1 & \forall i \in \{m, h\}, k \in K \\
& \quad \sum_{i \in V} x_{ij}^k \geq \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} y_{ij}^k & \forall i \in V, j \in V, i \neq j, k \in K \\
& \quad \sum_{i \in V} x_{ij}^k \geq \sum_{i \in V} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} w_{ij}^k & \forall j \in N \cup \{h\} \\
& \quad \sum_{i \in V} x_{ij}^k \geq \sum_{i \in V} \sum_{k \in K} \sum_{j \in V} w_{ij}^k & \forall j \in N \cup \{m\} \\
& \quad \sum_{i \in V} x_{ij}^k \geq \sum_{i \in V} \sum_{j \in D_2} x_{ij}^k & \forall k \in K \\
& \quad \sum_{i \in V} x_{ij}^k \geq \sum_{i \in V} \sum_{j \in P_2} x_{ij}^k & \forall k \in K \\
\end{align*}
\]
The objective function (1) minimizes the total routing cost. Constraints (2) ensure that all the demands are satisfied. Constraints (3) guarantee that a vehicle visiting a $P_2$-patient also visits the lab. Constraints (4) are similar but for hospital visit. Constraints (5) and (6) force the route of each vehicle to start and end at the depot. Constraints (7) ensure the flow balance of the vehicles, i.e., if a vehicle visits a node it must leave this node. Constraints (8) indicate that each vehicle can only visit the hospital and the lab once. Constraints (9) impose the consistency of the visiting times. Constraints (10) and (11) are flow equations for pickup and delivery demands. Constraints (12) impose that all delivery demands of $D_2$-patients are loaded at the hospital. Constraints (13) impose the unloading of all pickup demands of $P_2$-patients at the lab. Constraints (14) ensure that each $P_2$-patient is visited before a lab visit; constraints (15) ensure the hospital visit before any visit to a $D_2$-patient. Finally, constraints (16) and (17) impose the time window and vehicle capacity constraints. This MIP model is nonlinear because of constraints (9), (14) and (15) that can be linearized as follows:

$$B^k_m \geq (B^k_i + t_{im}) \sum_{j \in V} x^k_{ij} \quad \forall i \in P_2, k \in K$$ (14)

$$B^k_i \geq (B^k_i + t_{hi}) \sum_{j \in V} x^k_{ij} \quad \forall i \in D_2, k \in K$$ (15)

$$a_i \leq B^k_i \leq b_i \quad \forall i \in V, k \in K$$ (16)

$$y^k_{ij} + w^k_{ij} \leq Q x^k_{ij} \quad \forall i \in V, j \in V, k \in K$$ (17)

$$x^k_{ij} \in \{0, 1\} \quad \forall i \in V, j \in V, i \neq j, k \in K$$ (18)

3. Proposed genetic algorithm

The HHC vehicle scheduling problem belongs to the class of NP-hard problems, for that reason the exact solution methods become highly time-consuming as the problem instances increase in size. Therefore, due to the combinatorial nature of the VRP and the GAs’ efficiency in solving combinatorial problems, a GA based approach is developed to solve the problem. GAs can easily be adapted to various types of problems therefore many different GA approaches exist depending on the problems studied. There are several ways to maintain the population and several GA operators. However, all GA approaches must have a good genetic representation of the problem, an initial population generator, appropriate fitness function, and genetic operators such as crossover and mutation in order to work effectively. The generalized procedure of the GA approach is as follows:

begin
  t  0; ←
  initialize P(t) by encoding;
  evaluate P(t) by decoding;
  While (not terminating condition) do
    create C(t) from P(t) by crossover operator;
    create C(t) from P(t) by mutation operator;
    evaluate C(t) by decoding;
    select P(t + 1) from P(t) and C(t) by selection mechanism;
end
In this study, a permutation representation is used for genetic representation of the problem. The chromosome is a permutation of all patients that give order of visits in different routes. The GA starts with the generation of an initial population. In the proposed methodology, initial population generation process is based on random permutation. Due to the characteristics of our problem, the fitness value is calculated as the total routing cost. The solutions are evaluated in terms of their fitness value which is identical to the fitness of individuals. The individuals with better fitness value survive while the ones with worse fitness value die. This means that the more costly solutions are removed from the population while others are survived. We use an exact split algorithm proposed by Liu et al. (2013) for fitness evaluation of our problem. The algorithm splits the permutation into sub-strings of patients to be visited by a vehicle and to insert hospitals and labs. The roulette wheel selection is used in the proposed methodology for probabilistically selection of individuals based on their performance. The roulette wheel selection method scales the fitness value of the members within the population so that the sum of the rescaled fitness values equals to 1. In roulette wheel method, the probability of choosing an individual is directly proportional to its fitness value.

The main genetic operator is crossover, which simulates a reproduction between two parents. It works on a pair of solutions and recombines them in a certain way generating one or more offspring. The offspring share some of the characteristics of the parents and by this way the characteristics are passed on to the future generations. Crossover operator is not able to produce new characteristics. The other genetic operator is mutation, which is applied to a single solution with a certain probability. Mutation operator makes small random changes in the solution. These random changes will gradually add some new characteristics to the population, which could not be supplied by the crossover. In this study, partial-mapped crossover (PMX) and swap mutation are used for genetic operations of permutation based chromosomes. After crossover and mutation, each offspring is evaluated in terms of fitness value mentioned before. The procedure of PMX is as follows:

**Step1:** select two positions along the string uniformly random. The sub-strings defined by two positions are called mapping sections.

**Step2:** exchange two sub-strings between parents to produce offspring

**Step3:** determine mapping relations between two mapping sections

**Step4:** legalize offspring with mapping relationship

The swap mutation works as follows: two elements in the offspring are randomly selected, and then their positions are exchanged to create a new offspring.

### 4. Computational experiments

This section reports the results of a series of computational experiments for comparison of the proposed genetic algorithm and application of the commercial solver CPLEX 12.3 for the mathematical formulation of our problem. We evaluate the performance of the proposed approach with the test instances proposed by Liu et al. (2013) which are derived from existing VRPTW benchmarks of Solomon (1987). Liu et al. (2013) selected eighteen Solomon VRPTW instances to generate their test instances. Each Solomon instance contains 100 customers over a service region defined on a 100×100 grid. These VRPTW instances are divided into three classes that differ by the geographical distribution of the customers: they are clustered in the C type instances, randomly located in the R type instances, and partly clustered, partly randomly located in the RC type instances. Meanwhile, each class is divided into two series: in the 100-series instances time windows are tighter, and in the 200-series instances time windows are wider. To test different characteristics of instances, Liu et al. (2013) selected 6 C type instances, 6 R type instances and 6 RC type instances. Among six instances of each type, both the 100-series instances and 200-series ones exist.

For each Solomon instance, they derived six new instances for the problem with 4Z demands as follows. First, they randomly chose Z patients from the Solomon instance as the \( P \) patients in their new instance, each of which has a demand equal to 50% of the customer’s demand given in the Solomon instance. Then, \( ZP_2 \), \( ZD_1 \) and \( ZD_2 \) patients are randomly selected from the Solomon instance. Clearly, one patient may be selected more than once and the number of patients is less than then number of requirements (4Z).
The coordinates of the depot is inherited in their instances, and the locations of lab and hospital are (10,15) and (40, 50). For each patient, the time window in the Solomon instance is used directly. Time windows for the depot, lab and hospital are selected as follows to avoid infeasible solutions. The depot’s time window of the Solomon instance is multiplied by 1.2 and assigned to the depot, lab and hospital in their instance. For each Solomon instance, this constructing procedure is repeated six times, generating two small (40 demands), two moderate (80 demands), and two large (120 demands) instances. Concerning the vehicle capacity and vehicle number, their values have been reduced compared to the ones considered for the VRPTW, because they are loose for the problem.

4.1. Illustrative example and parameter settings

In this illustrative example, we use moderate (80 demands) instances. Initially, to determine the appropriate population size and number of generations for GA, a pilot study is conducted. The proposed approach is applied with combinations of population size 30, 50, 100 and number of generations 500, 1000 and 1500. The other parameters used in GA are crossover rate 0.80 and mutation rate 0.03. Results of 10 iterations over all relevant problem instances are summarized in Table 1. For each combination, best objective function value (column I), average of the best values (column II), worst of the best values (column III), average objective function value of the population (column IV) as well as computation time as average CPU times (s) (column V) are given.

<table>
<thead>
<tr>
<th>Popsize</th>
<th>Number of generations</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>500</td>
<td>1435</td>
<td>1462</td>
<td>1658</td>
<td>2544</td>
<td>226.1</td>
</tr>
<tr>
<td>50</td>
<td>500</td>
<td>1468</td>
<td>1699</td>
<td>1730</td>
<td>2317</td>
<td>266.3</td>
</tr>
<tr>
<td>100</td>
<td>500</td>
<td>1696</td>
<td>1733</td>
<td>1803</td>
<td>2609</td>
<td>303.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1000</td>
<td>1446</td>
<td>1455</td>
<td>1624</td>
<td>2680</td>
<td>306.7</td>
</tr>
<tr>
<td>50</td>
<td>1000</td>
<td>1494</td>
<td>1545</td>
<td>1601</td>
<td>2814</td>
<td>348.5</td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>1520</td>
<td>1671</td>
<td>1699</td>
<td>3110</td>
<td>395.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1500</td>
<td>1442</td>
<td>1470</td>
<td>1549</td>
<td>2759</td>
<td>383.1</td>
</tr>
<tr>
<td>50</td>
<td>1500</td>
<td>1420</td>
<td>1536</td>
<td>1601</td>
<td>2692</td>
<td>406.7</td>
</tr>
<tr>
<td>100</td>
<td>1500</td>
<td>1439</td>
<td>1678</td>
<td>1666</td>
<td>2833</td>
<td>457.5</td>
</tr>
</tbody>
</table>

From these results, it can be concluded that population size 30 combined with number of the generations 1000 results in good solutions considering the average of best objective function values. Additionally, combination of population size 50 with number of generation 1500 results the best solution found by the proposed approach.

5. Results and discussion

During the computational experiments, parameter values which were previously determined are used as follows: population size= 50; maximum number of generations= 1500; crossover rate= 0.8; mutation rate=0.03. The HHC vehicle scheduling problem is formulated as MIP model and CPLEX 12.3 solver is used for obtaining upper bounds of the objective function. The best integer feasible solutions given by CPLEX are used as an upper bound (UB). In addition, the best-known solutions found by Liu et al. (2013) can be considered as lower bounds (LB).

Table 2 shows the average routing costs by grouping problem instances according to the number of demands and the type of the instance. The results are obtained from 10 independent runs for each problem instance of the proposed approach plus the one obtained with CPLEX. Column ‘GA-Best’ is the average over all relevant problem instances of the best solutions among 10 independent runs of GA approach for each instance. Column
‘CPU’ is the average CPU time in seconds of one run among relevant test instances. The LB, UB and results of GA based approach are shown in Fig. 1.

Table 2. The results of the computational experiments

<table>
<thead>
<tr>
<th>Demand</th>
<th>Type</th>
<th>GA-Best</th>
<th>LB Best-known solution</th>
<th>UB CPLEX</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>902.7</td>
<td>881.9</td>
<td>1093.0</td>
<td>101.7</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>R</td>
<td>898.4</td>
<td>875.3</td>
<td>1078.7</td>
<td>85.1</td>
</tr>
<tr>
<td>RC</td>
<td>1008.9</td>
<td>954.4</td>
<td>1191.6</td>
<td>98.4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1441.9</td>
<td>1424.3</td>
<td>1866.5</td>
<td>420.1</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>R</td>
<td>1435.6</td>
<td>1409.1</td>
<td>1780.1</td>
<td>404.6</td>
</tr>
<tr>
<td>RC</td>
<td>1601.5</td>
<td>1577.6</td>
<td>1916.2</td>
<td>381.2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1520.2</td>
<td>1503.9</td>
<td>1973.9</td>
<td>658.7</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>R</td>
<td>1532.8</td>
<td>1505.4</td>
<td>2072.6</td>
<td>674.8</td>
</tr>
<tr>
<td>RC</td>
<td>1710.4</td>
<td>1691.3</td>
<td>2202.4</td>
<td>583.0</td>
<td></td>
</tr>
</tbody>
</table>

As seen from Table 2, our approach is robust and can find good solutions for different test instances in different runs within significantly CPU time. Fig. 1 shows that the proposed GA provides solutions which are between lower bound and upper bound for all of test instances. It means that for each combination of types and sizes, our algorithm significantly dominate the CPLEX solver. Considering these results and CPU time, it can be stated that GA is efficient and performs well for test instances of different types and different sizes.

6. Conclusion

This paper investigates a special simultaneous pickup and delivery problem with time windows in HHC, an extension of the classical VRPSDPTW. The problem is of interest because of its theoretical complexity and of the important applications in the home health care industry. We formulate the problem as an integer programming model to minimize the total vehicle cost for serving all patients demands. We also propose a genetic algorithm based approach to solve this problem. Experiments are conducted by using a range of test instances, which are designed based on existing VRPTW benchmarks to reflect different realistic scenarios. In
general, proposed GA can provide good solutions in a reasonable time span and is competitive when compared with the best-known solutions in the literature.

For further studies, the HHC may choose the visiting days for each patient. Additionally, the proposed approach may be applied to real-world home health care logistic applications.

References


