A GRASP Algorithm Based on New Randomized Heuristic for Vehicle Routing Problem

Abdesslem Layeb, Meryem Ammi and Salim Chikhi

Computer Science Department, University of Constantine, Algeria

This paper presents a novel GRASP algorithm based on a new randomized heuristic for solving the capacitated vehicle routing problem. This problem is characterized by a fleet of homogenous vehicle capacity that will start from one depot, to serve a number of customers with demands that are less than the vehicle capacity. The proposed method is based on a new constructive heuristic, and a simulated annealing procedure as an improvement phase. The new constructive heuristic uses four steps to generate feasible initial solutions, and the simulated annealing enhances these solutions found to reach the optimal one. We tested our algorithm on two sets of benchmark instances and the obtained results are very encouraging.

Keywords: optimization problems, CVRP, constructive heuristics, nearest neighbor heuristic

1. Introduction

The Vehicle Routing Problem (VRP), proposed by Dantzig and Ramser [1], is an important research area in operation search given its importance in logistic planning. In this paper we consider the Capacitated Vehicle Routing Problem (CVRP), in which each customer has a known demand less than the vehicle capacity. All the vehicles start from one depot, serve the demands of a set of customers, and return to the same depot. The aim is to determine the total traveled distance, in a way that each customer is visited exactly once by a single vehicle, subject to the restriction that the total load on any vehicle associated with a given route does not violate the vehicle capacity. Although the problem appears to be easy to state, it belongs to NP-hard problems category, and needs a huge time and space to resolve optimally, especially with hard VRP instances [2, 3].

Several methods were proposed to solve the CVRP that can be classified in three main classes. The first class contains the exact algorithms which are able to find the exact solution of the CVRP, although they have an exponential complexity [3]. The most popular algorithms of this class are based on the Branch & Bound [6], Branch & Cut [7], column generation [8], and set partitioning [9]. Unfortunately, the exact methods are impracticable for large CVRP instances. The second class contains heuristic techniques or constructive methods, in which the solutions are constructed progressively. The main features of the constructive heuristics are the simplicity and the rapidity to find good acceptable solutions. The best known heuristics for CVRP are the saving heuristic of Clarke and Wright [10], the sweep algorithm [11], the insertion heuristic [12], the petal algorithm [13] and the nearest neighbor algorithm. The main inconvenience of heuristics is that they do not guarantee the optimality and sometimes the solutions are too far from the best solutions. In order to improve the quality of the heuristic solutions, improvement methods are used [14]. The improvement methods like K-opt approach [15] try to find an enhanced solution from an initial one which is not created by a constructive heuristic. These algorithms delete some arcs and replace them with new ones in order to find a better solution. This process is repeated until no possible improvement in the value of the objective function is found. Finally, the third class contains metaheuristics like genetic algorithm [16], particle swarm optimization [17], ant colony [18], local search methods [19, 20]. Compared to the classical heuristics, the metaheuristic methods

explore efficiently the solution space. However, they are slower than the constructive heuristics and depend on the initial solutions. In addition to the previous methods, several hybrid methods have emerged in recent years [21, 22], which hybridize between exact methods, metaheuristics or constructive heuristics.

Greedy Randomized Adaptive Search Procedure (GRASP) is a metaheuristic algorithm commonly applied to several combinatorial optimization problems. A GRASP is an iterative procedure, where each GRASP iteration consists of two phases: construction and local search. The construction phase is a greedy procedure used to build a feasible solution [23, 24]. The second phase is a local search used to enhance the solution found in the constructive phase. The two stages are repeated several times independently and the best overall solution is selected as the final result. This technique was successfully used to solve the VRP problems [25,26].

The purpose of this paper is to present new GRASP algorithm based on a new heuristic and the simulated annealing for solving the CVRP. The idea of the new heuristic is based on the density between demands and distances of customers, and it was first proposed for the knapsack problem [27]. The first step of the proposed heuristic consists of computing the density matrix of the problem. Then, a giant tour is constructed subject to the order given by the density matrix; the costumer with the high density in the current route is selected to be the next visited customer. In the stochastic version, a non visited customer can be selected randomly with some probability, even if its density is smaller. In the next phase, the giant tour is split into feasible routes subject to the vehicle capacity. Finally, each route is reordered by the use of the nearest neighbor heuristic. In the second phase of our algorithm, we apply the simulated annealing based on a set of operators such as the swap inter- and intra-routes, and the 2-opt move intra- and inter-routes. The computational results over a set of known instances show the effectiveness of the proposed heuristic.

The rest of the paper is organized as follows. In Section 2, a formulation of the tackled problem is given. In Section 3, the proposed method is described. Experimental results are discussed in Section 4. Finally, a conclusion and future works are drawn.

2. Capacitated Vehicle Routing Problem Formulation

The Capacitated Vehicle Routing Problem (CVRP) consists of a number of customers with known demands, served by a number of vehicles departing from one depot. The objective function of the CVRP that should be minimized is the total cost distance travelled by all the vehicles. The problem is solved subject to the following constraint [24, 25]:

- m identical vehicles of capacity cap.

- n customers with demands $d_i > 0$, $i = 1, \ldots, n$

- All vehicles routes start and end at the depot

- Each vehicle serves a subset of customers

- Each customer is served once and by one vehicle

Moreover, there is an integer programming formulation of the CVRP. Having the following variables:

- i: is the index of customer ($i = 1, 2, \ldots, n$), n is the number of customers, and the index of depot is $i = 0$;

- k: is the index of vehicle ($k = 1, 2, \ldots, m$), m is the number of vehicles;

- d_i: the demand of customer i;

- $Dist_{ij}$: the distance cost between customer i and customer j;

- cap: the capacity of vehicle;

- y_{ik}: binary variable: its value is 1 if the customer i is delivered by the vehicle k; otherwise it is 0;

- x_{ijk}: binary variable: its value is 1 if the vehicle k travels directly from customer i to customer j, otherwise it is 0.

The objective function is:

$$\min \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} Dist_{ij} * x_{ijk} \qquad (1)$$

Subject to

$$\sum_{i=1}^{n} d_i * y_{ik} \leq cap \quad pk = 1 \ldots m \qquad (2)$$

$$\sum_{k=1}^{m} y_{ik} = 1 \quad i = 1 \ldots n \qquad (3)$$

$$\sum_{i=1}^{n} x_{ijk} = y_{kj} \quad j = 1 \ldots m, \forall k \qquad (4)$$

$$\sum_{j=1}^{n} x_{ijk} = y_{ki} \quad i = 1 \ldots n, \forall k \qquad (5)$$

where equation (1) indicates that the objective function of the VRP problem is to minimize the total distance; the constraint equation (2) controls that the load of each vehicle does not exceed its capacity; the constraint equation (3) signifies that each customer's demand must be satisfied by one vehicle; the constraint equations (4) and (5) ensure that there is only one vehicle which arrives and departs from a customer.

3. The Proposed Algorithm

To solve the Capacitated Vehicle Routing Problem, we propose a new algorithm based on Greedy Randomized Adaptive Search Procedure (GRASP). GRASP is a two-phase metaheuristic technique used to solve hard combinatorial optimization problems. The first phase is a constructive phase used for the construction of initial solutions while the second phase is generally a local search method used for the improvement of the solutions found in the first phase [23, 24]. A generic GRASP approach is described in Figure 1.

The main difference between our algorithm and the other GRASP approaches used for solving the CVRP is the use of a novel randomized heuristic to construct the initial solutions. In order to show how GRASP concepts have been tailored to the problem at hand, we need first to present the novel heuristic used to generate the initial solutions.

```
Input: VRP data, Number_iteration
Output: Best_solution

Initialization;
repeat
  a. Solution=Constructive_phase(VRP data);
  b. Local_Search(Solution);
  c. Best_solution= Update_solution(Solution);
until (Number_iteration)
End
```

Figure 1. The GRASP metaheuristic scheme.

3.1. The constructive phase

Constructive heuristics construct a feasible solution by adding one element to the current partial solution until a complete one is found. In this paper, we present a new heuristic algorithm for the capacitated vehicle routing problem. The proposed method is inspired by the density ordered heuristic of the knapsack problem [27]. It follows the Route-first Cluster-second strategy [3, 4], which consists in constructing first a giant tour followed by the split of this tour into feasible routes. The main steps of the proposed heuristic are described by the following algorithm:

```
Input: VRP problem instance
Output: VRP solution

1. Compute the Density Matrix of the VRP instance.
2. Construct the giant tour subject to the density matrix.
3. Split the giant tour subject to customers' demands and vehicle capacity.
4. Improve each route by the nearest neighbor heuristic
```

Figure 2. The heuristic scheme.

We explain in more details the proposed heuristic. Given a VRP instance, with one depot, n customers, each customer has a static demand d_i, and m vehicles of identical capacity *cap*. Our heuristic starts with the computation of the density values between every pair of customers (including the depot). The density value is the difference between the vehicle capacity and the customers' demands; the result is divided by the distance between a pair of customers, as described by the following equation:

$$Density(i,j)$$
$$= \frac{|(cap - (d_i + d_j)|^p}{(dist(i,j)^k * Density(i, depot) * Density(j, depot))}$$
$$Density(i,i) = 0$$
$$i = 1 : n; \quad j = 1 : n; \quad k, p \in \{1, 2, 3, 4\} \qquad (6)$$

Where, *demand(i)* is the demand of the customer i, *dist(i,j)* is the distance between the customers i and j, *Density (i,depot)* is the density

between the customer i and the depot, k and p are integer parameters between 1 and 4 that affects the solution' quality. If p has a big value, the customer with the lowest demand over the vehicle's capacity will be selected to be the next visited one. Also if k has a big value, the choice of the next visited customer will be the customer whose angular distance is very small.

Secondly, we construct a giant tour by using the density matrix as ordering criteria. We suppose having one vehicle with unlimited capacity used to visit all customers which are ordered by their density. First, we begin with the customer i_s having the highest density value to the depot $(Max(density(depot, i_s)))$. Then, we select a second customer i_{next} to visit, who has the highest density value with the previous visited customers $i_s(Max(density(i_s, i_{next})))$. However, in the stochastic version, we can select a random unserved customer with some probability, which leads to generate diverse solutions. The process is repeated until all the customers are inserted into the giant tour. The third step of our heuristic consists of splitting the giant tour into feasible routes subject to the order defined in the previous step and the vehicle capacity. For each vehicle, the current customer in the giant tour is selected if its demands can be satisfied by the available vehicle capacity; else, the next customer is selected. The process is repeated until all the customers are inserted into feasible routes.

Finally, in order to enhance the solution quality, the nearest neighbor heuristic is applied on each sub tour if a better solution is found. The nearest neighbor heuristic works as follows: the vehicle starts at the depot and then visits the customer nearest to the depot. From there, it visits the nearest customer that has not been visited so far, etc., until all the customers in one route are visited, and therefore the vehicle returns to the depot. However, we can use other sophisticated heuristics in order to improve efficiency of each route, such as the k-opt approach, swap, etc.

3.2. The local search phase

The Stochastic Local Search (SLS) methods were demonstrated to be useful in solving many complex problems. The Simulated Annealing (SA) algorithm [28, 29] is among the most used local search methods in solving many hard combinatorial optimization problems. The Simulated Annealing is inspired by the annealing process in the metallurgy [29]. The simulated annealing starts from a given solution, and applies iteratively some perturbations on this solution in order to get a new one. The perturbation is accepted if it has the effect of reducing the objective function (or energy) of the problem; Otherwise, SA can accept uphill moves with the probability $exp(\Delta E/T)$, where ΔE is the difference between the objective value of the current solution and the modified one, and T is the temperature parameter. The temperature T decreases during the search process. Consequently, at the first steps of SA, the probability of accepting uphill moves is great, and decreases progressively until it is difficult to accept new uphill moves. The simulated annealing was among the first local search algorithms developed for the routing problems [28]. In our paper, we use a standard SA procedure with a random neighborhood structure composed of 7 neighborhoods operators. Four of them operate within the routes called intra-route improved operators, and the others called inter-route improved operators, operate between two routes. In the following, we present the different neighborhoods operators used within the simulated annealing procedure.

3.2.1. Intra-route operators

Several intra-route operators have been proposed in the literature; generally, they are taken from the TSP improvement operators. Among the well-known intra-route improvement operators, we have used the following operators, (it should be noted that in one run we apply randomly one operator to improve one route).

- **Move operator**

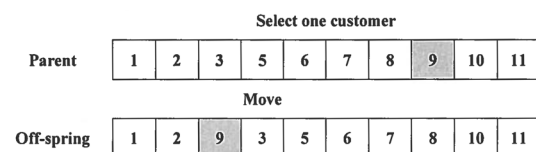   One random customer is moved from its current position to a new random one (Figure 3).

	Select one customer									
Parent	1	2	3	5	6	7	8	9	10	11
	Move									
Off-spring	1	2	9	3	5	6	7	8	10	11

Figure 3. Move operator.

- **Swap operator**

  The Swap operator consists of selecting randomly two positions along the individual and swaps their contents, as shown in following figure.
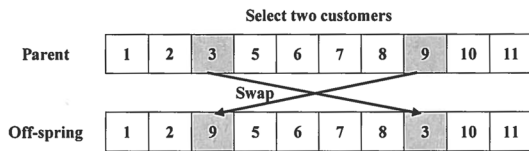

Figure 4. Swap operator.

- **Inversion operator**

  The inversion operator is closely related to the 2-opt operator, it aims to inverse a subsequence between two customers in the route (Figure 5)
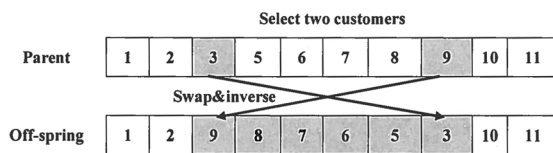

Figure 5. Inversion operator.

- **Intra-route two-opt operator**

  The 2-opt operator is among the most robust improvement operators for TSP problems. It is an iterative operator, at each step we delete two edges and add two new edges, and the process is repeated until no future improvement can be done. The 2-opt improvement operator operates as shown in Figure 6.
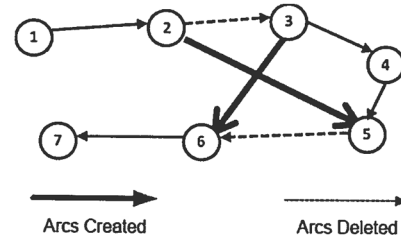

Figure 6. Intra-route 2-opt operator.

3.2.2. Inter-route operators

This kind of operators is developed specifically for VRP; they involve more than one route.

- **1-0 exchange operator**
  A customer is deleted from one route and inserted in another route, in order to get good solution, the deleted customer is inserted in the best position in the second route (Figure 7a).

- **1-1 exchange operator**
  Two customers are deleted from two different routes, and each deleted customer is inserted in the other route. In order to get good solution, the deleted customer is inserted in the best position in the second route (Figure 7b).

- **Inter-route two-opt operator**
  The Inter-route two-opt operator removes one edge from two different routes and re-links the first node of the edge with a second node of the second edge and vice versa (Figure 7c).
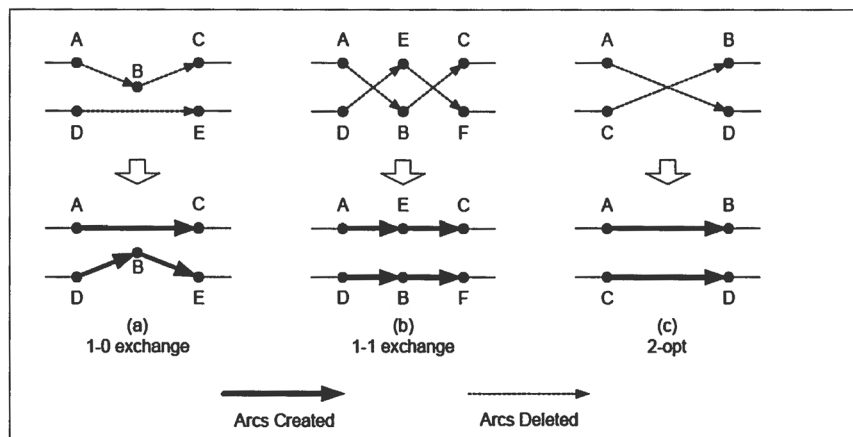

Figure 7. Inter-route operators.

The previous operators are integrated in the simulated annealing procedure, the choice of the simulated annealing is to escape from the local minima, uphill moves can be accepted in order to avoid being trapped in local minima.

At the beginning, the SA parameters like the temperature T is set to T_0. Then, an initial solution $VRPs$ is constructed by using the randomized heuristic described in Section 2.1. The current best solution $VRPs_{best}$ and the best objective function $Cost_{best}$ are set to be $VRPs$ and $Cost(VRPs)$, respectively.

At each iteration, a new solution $VRPs'$ is generated from the neighborhood of $VRPs$ by using randomly one of the improvement operators described previously If the new solution is better than the current best, then we save the current solution as the best current one. However, the uphill move can be accepted with the probability $exp(-\Delta/T)$, where Δ is the difference between solutions costs $Cost(VRPs')$ and the $Cost_{best}$. At each iteration the current temperature T is decreased by using cooling scheme $T \leftarrow \alpha T$, where $0 < \alpha < 1$. Moreover, to increase the convergence of the proposed algorithm, each new best solution is improved by using a local search procedure based on 2-opt operator

The algorithm is terminated when the cool temperature cool T_F is reached. The proposed simulated annealing used for VRP is described in the following figure.

4. Experimental Results

The proposed algorithm called GRASPVRP is implemented in Matlab 7 and tested on Core Duo 2 GHZ with 2 GB of memory. In order to evaluate the performances of our algorithm, we have used two data sets. The first test data sets used in this experiment study are taken from the VRP benchmarks http://www.branchandcut.org/VRP/data/. The VRP instances are taken from series A, B, E, F, G, P and M. All the tests used, are presented in the following tables with the same format name (X-nNC-kNV) where X denotes the name of the data sets (A, B, E, F, G, P, M), NC is the number of customers including the depot, and NV represents the number of vehicles. For example, the data set "A-n80-k10" refers to an instance having 80 customers (including the depot) and 10 vehicles. Respectively, *A* and *B* contain small VRP tests. However, the other data sets contain generally hard VRP instances. The second data sets contain 14 benchmark problems proposed by Christofides

---

Input: Solution VRPs , VRP data

Initialisation:
Select an initial temperature T=T0.
Set $VRPs_{best}$= VRPs and $Cost_{best}$ = $Cost($VRPs$)$.

Repeat
VRPs'=apply randomly one inter-route or intra-route improvement operator to VRPs.
Calculate the objective function of VRPs'.
If $Cost($VRPs'$)>$ VRPs$_{best}$ **then**
VRPs = VRPs'
VRPs$_{best}$ = VRPs'
$Cost_{best}$ = $Cost($VRPs'$)$
Improve VRPs$_{best}$ by 2-opt local search algorithm
Else
Generate a random number r
If r < Exp(($Cost($VRPs$)$- $Cost($VRPs'$))$/T) **then**
$VRPs$=$VRPs'$

Update the temperature T.

Until a termination criterion is reached.

Output: improved VRP solution

Figure 8. The simulated annealing for VRP scheme.

[30]. The Christofides instances 6 to 10, 13 and 14 have, besides the capacity constraint, maximum route length limits and non-zero service times.

The obtained results are summarized in Tables 1, 2, 3 and 4. Finally, statistical tests of Freidman were carried out to test the significance of accuracy' difference of each method in this experiment.

4.1. Results and discussion of the proposed heuristic

In the first part of experiments, we have reported the best results found by the deterministic version of the proposed heuristic HDVRP, the randomized one (RHDVRP) and the proposed GRASP based on RHDVRP (GRASPVRP). The number of iterations in RHDVRP and GRASPVRP and algorithm is 30.

In the test data set A (Table 1), the GRASVRP results are the nearest to the best known solutions. Indeed, there is no significant difference between the results of GRASPVRP and those of the best known ones. It is clear that the RHDVRP and GRASPVRP are better than the deterministic heuristic HDVRP, as it is confirmed by Freidman test (Figure 9).

	HDVRP	RHDVRP	GRASPVRP	Best known
A-n80-k10	2180	2085	1763	1763
A-n63-k10	1708	1667	1315	1314
A-n69-k9	1485	1436	1159	1159
A-n65-k9	1580	1500	1177	1174
A-n60-k9	1694	1618	1354	1354
A-n62-k8	1461	1461	1288	1288
A-n54-k7	1485	1426	1167	1167
A-n45-k7	1447	1395	1146	1146
A-n45-k6	1159	1068	950	944
A-n33-k6	913	818	742	742
A-n39-k5	1004	947	822	822
A-n34-k5	947	911	778	778

Table 1. Results for A instances.

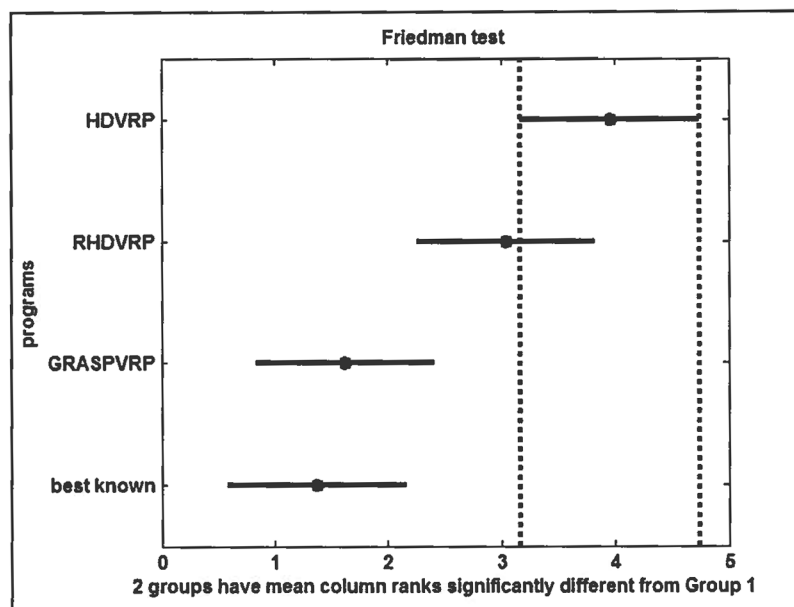|	HDVRP	RHDVRP	GRASPVRP	Best known
B-n78-k10	1591	1414	1221	1221
B-n67-k10	1431	1263	1032	1032
B-n66-k9	1730	1506	1316	1316
B-n64-k9	1403	1107	861	861
B-n50-k8	1573	1503	1312	1312
B-n57-k7	1643	1457	1153	1153
B-n44-k7	1191	1060	909	909
B-n45-k6	972	743	678	678
B-n38-k6	1060	970	805	805
B-n45-k5	869	826	751	751
B-n35-k5	1130	1075	955	955

Table 2. Results for B instances.

Figure 9. Freidman test compares HDVRP, RHDVRP, GRASPVRP, and best known on A data set.

In the second experiment, we have tested our algorithm on the *B* data set (Table 2), the proposed GRASPVRP is able to find the exact solutions of all the B benchmarks. The performance of RHDVRP in this experiment is better than its performance in the previous experiment; it is successful in the Freidman test, (Figure 10).

In the third experiment held on the hard instances category (Table 3), the GRASPVRP results are the nearest to the best known ones. On the other hand, the deterministic version of the HDVRP and its randomized version RHDVRP are not successful in this experiment (Figure 11).

	HDVRP	RHDVRP	GRASPVRP	Best known
E-n101-k8	1072	1056	815	815
E-n101-k14	1424	1319	1071	1071
F-n135-k7	1631	1319	1162	1162
F-n72-k4	365	357	237	237
G-n262-k25^E	8423	8095	5592	5543
M-n151-k12^E	1531	1444	1021	1015
M-n200-k17^E	1850	1757	1292	1281
M-n200-k16^E	1876	1730	1310	1284
M-n101-k10	1120	1048	820	820
P-n101-k4	865	838	681	681
P-n76-k4	788	737	593	593
P-n70-k10	1049	1030	827	827

Table 3. Results for hard instances (the results G and M instances are taken from *xtremeroute* website).
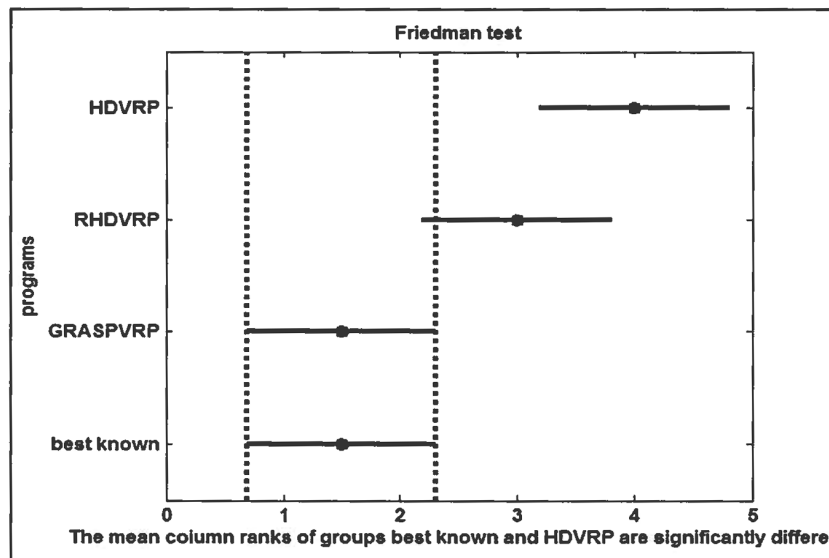

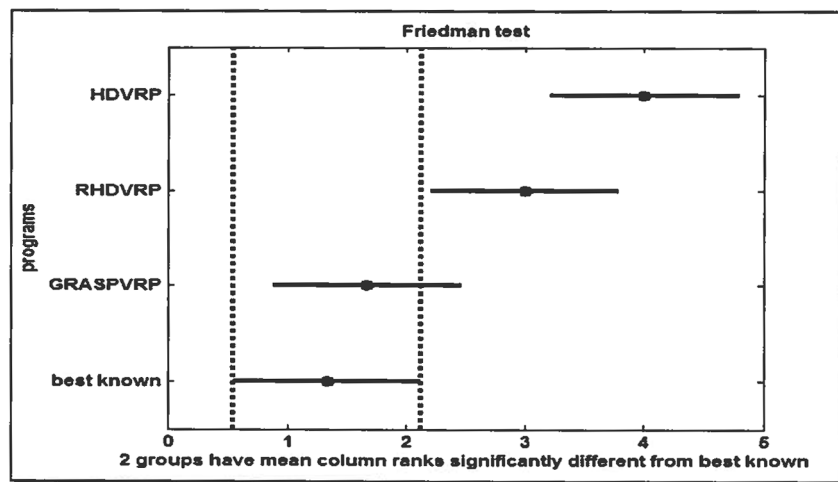*Figure 10.* Freidman test compares HDVRP, RHDVRP, GRASPVRP, and best known on B data set.

Figure 11. Freidman test compares HDVRP, RHDVRP, GRASPVRP, and best known on hard CVRP instances.
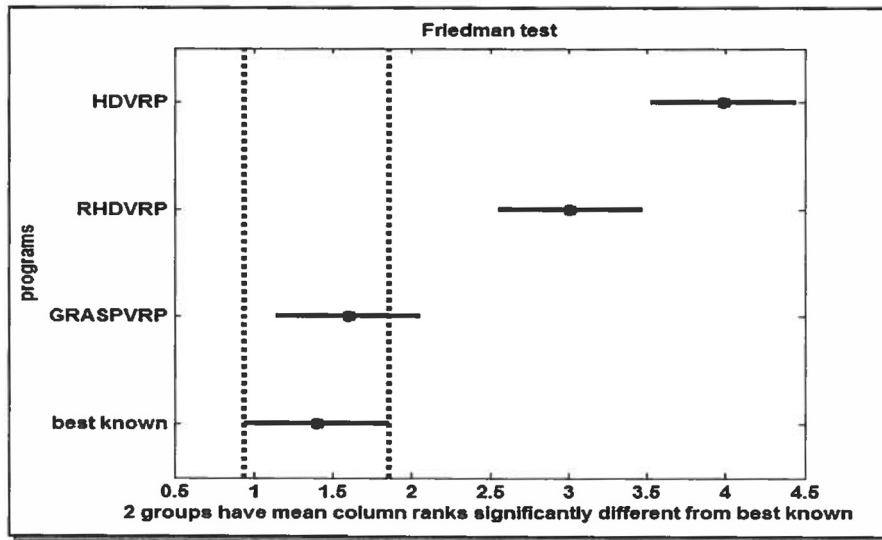
Figure 12. Freidman test compares HDVRP, RNNVRP, NNVRP, RHDVRP, and best known on all instances.

Figure 12 shows the performances of all programs on the entire tests. As it is shown in the Friedman test, there is no significant difference between the GRASVRP results and the best known solutions. However, the results of HDVRP and its randomized version RHDVRP are far from the best solutions.

In this section, we report the results of GRASPVRP compared against the best results reported so far in the table below, on the benchmark of Christofides instances. We have compared our algorithm with six popular and efficient algorithms, namely, simulated annealing

with Tabu (OSA) [28], Tabu search for VRP (TS) by Toth and Vigo [31] adaptive memory programming method called (SEPAS) by Tarantilis [22], a savings ants (SA)[32], GRASP algorithm of Prins (GRELS) [33] and finally a GRASP algorithm based on Neighborhood Search (MPNS-GRASP) [26].

As it can be drawn from Table 4, the GRASPVRP has found the optimal solution for 6 out of 14 instances of Christofides. For the rest of the instances, our results do not show a significant difference between them and those reported as the best known, as shown in Figure 13, the

Instance	OSA	TS	SA	SEPAS	GRELS	MPNS-GRASP	GRASP-VRP	BKS
C1	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**
C2	835.32	838.6	840.61	**835.26**	**835.26**	836.39	836.18	**835.26**
C3	827.53	828.56	828.21	**826.14**	**826.14**	**826.14**	**826.14**	**826.14**
C4	1044.35	1033.21	1037.57	**1028.42**	1029.48	1032.24	1031.8	**1028.42**
C5	1334.55	1318.25	1306.91	1311.48	1294.09	1314.25	1308.6	**1291.29**
C6	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**
C7	**909.68**	920.72	917.5	**909.68**	**909.68**	**909.68**	914.5	**909.68**
C8	866.75	869.48	**865.94**	**865.94**	**865.94**	**865.94**	874.89	**865.94**
C9	1164.12	1173.12	1173.94	**1162.55**	**1162.55**	1175.86	1177.9	**1162.55**
C10	1420.84	1435.74	1415.53	1407.21	1401.46	1412.11	1412.6	**1395.85**
C11	**1042.11**	1042.87	1043.46	**1042.11**	**1042.11**	**1042.11**	**1042.11**	**1042.11**
C12	**819.56**	**819.56**	**819.56**	**819.56**	**819.56**	821.12	**819.55**	**819.56**
C13	1550.17	1545.51	1546.84	1544.01	1545.43	1548.53	1548	**1541.14**
C14	**866.37**	**866.37**	**866.37**	**866.37**	**866.37**	868.62	**866.37**	**866.37**

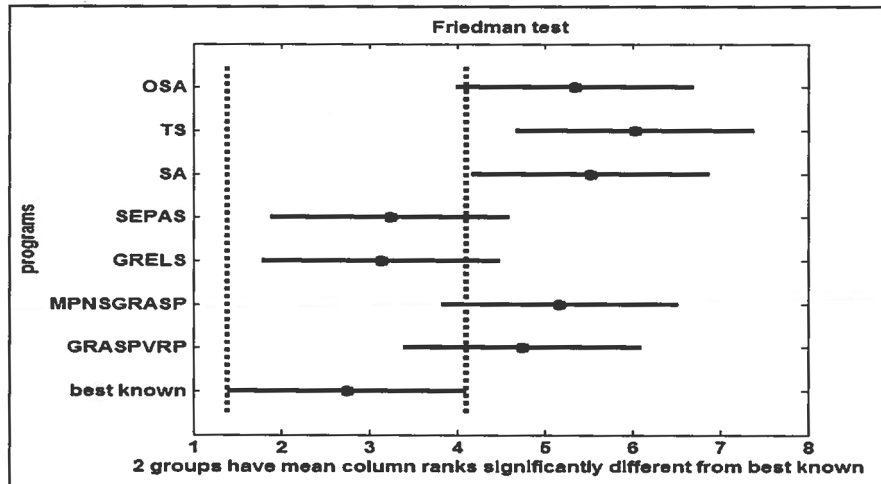*Table 4.* Results for Christofides instances.

Figure 13. Freidman test compares GRASP program against other programs on Christofides instances.

results of our algorithm are close to the best known. These results show that GRASPVRP is better than some other approaches, such as Tabu Search (TS) and Simulated Annealing (OSA), the SEPAS program based on adaptive memory programming, our algorithm ranks the third as reported by the Friedman test.

5. Conclusion

In this paper we presented a new GRASP algorithm to solve the capacitated vehicle routing problem. The proposed approach is based, firstly, on a new constructive heuristic; the first phase of this heuristic is the construction of a giant tour subject to the density value of each customer. The tour is split according to the customers' demands and the available vehicle capacity. Secondly, to enhance the quality of the solution found by the construction phase, we have used an adaptive simulated annealing based on several neighborhood operators as the second phase of the GRASP algorithm. The obtained results on two different benchmarks are very encouraging and demonstrate the quality of the proposed approach.

References

[1] G. B. DANTZIG, J. H. RAMSER, The Truck Dispatching Problem. *Management Science*, **6**(1) (1959), 80–91.

[2] P. TOTH, D. VIGO, (EDS.), The Vehicle Routing Problem. *SIAM Monographs on Discrete Mathematics and Applications, SIAM*, (2002a) Philadelphia

[3] G. LAPORTE, Fifty years of vehicle routing. *Transportation Science* **43**(4) (2009), 408–416

[4] G. LAPORTE, F. SEMET, (2001). Classical heuristics for the capacitated VRP. P. Toth, D. Vigo, eds.,The vehicle routing problem, Society for Industrial and Applied Mathematics, Philadelphia, PA. pp. 109–128

[5] M. CHARIKAR, S. KHULLER, B. RAGHAVACHARI, Algorithms for capacitated vehicle routing. *SIAM Journal on Computing*, **31**(3) (2001b) 665–682.

[6] P. TOTH, D. VIGO, (2001). Branch-and-bound algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM: Philadelphia, pp. 29–52.

[7] J. LYSGAARD, A. LETCHFORD, R. EGLESE, A new branch-and-cut algorithm for capacitated vehicle routing problems. *Math. Program*, **100** (2004), 423–445.

[8] R. BALDACCI, A. MINGOZZI, A unified exact method for solving different classes of vehicle routing problems. *Math. Program*, **120** (2009), 347–380.

[9] R. BALDACCI, N. CHRISTODES, A. MINGOZZI, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, **115**(2) (2008) 351–385.

[10] G. CLARKE, J. W. WRIGHT, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**(4) (1964), 568–581.

[11] B. GILLETT, L. MILLER, A Heuristic Algorithm for the Vehicle Dispatch Problem. *Operations Research*, **22** (1974), 340–349.

[12] A. M. CAMPBELL, M. SAVELSBERG, Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science*, **38** (2004), 369–378.

[13] D. M. RYAN, C. HJORRING, F. GLOVER, Extensions of the Petal Method for Vehicle Routing. *Journal of the Operational Research Society*, **44** (1993), 289–296.

[14] G. A. P. KINDERWATER, M. W. P. SAVELSBERGH, Vehicle routing: handling edge exchanges. In E. H. L. Aarts and J. K. Lenstra (eds), *Local Search in Combinatorial Optimization Wiley*, (1997) Chichester, England.

[15] S. LIN, Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, **44** (1965), 2245–2269.

[16] B. M. BAKER, M. A. AYECHEW, A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, **30** (2003), 787–800.

[17] T. J. AI, V. KACHITVICHYANUKUL, Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers & Industrial Engineering*, **56** (2009), 380–387.

[18] J. BELL, E. MCMULLEN, R. PATRICK, Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, **18**(1) (2004), 41–48.

[19] M. GENDREAU, A. HERTZ, G. LAPORTE, A tabu search heuristic for the vehicle routing problem. *Management Science*, **40** (1994), 1276–1290.

[20] P. CHEN, H.-K. HUANG, X.Y. DONG, Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, **37** (2010), 1620–1627.

[21] J. BERGER, M. BARKAOUI, (2003). A hybrid genetic algorithm for the capacitated vehicle routing problem. In Cantú-Paz, E., ed.: GECCO03. LNCS 2723, Illinois, Chicago, USA, Springer-Verlag, pp. 646–656

[22] C. D. TARANTILIS, E. E. ZACHARIADIS, C. T. KIRANOUDIS, A hybrid metaheuristic algorithm for the integrated vehicle routing and three dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, **10** (2009), 255–271.

[23] A. LAYEB, M. SELMANE, M. BENCHEIKH EL-HOUCINE, A new greedy randomized adaptive search procedure for multiple sequence alignment, to appear in *International Journal of Bioinformatics Research and Applications, Inderscience publisher*, **9**(4) (2013), 323–335.

[24] A. V. MOURA, R. A. SCARAFICCI, A GRASP strategy for a more constrained School Timetabling Problem. *International Journal of Operational Research*, **7**(2) (2010), 152–170.

[25] CH. DUHAMEL, PH. LACOMME, CH. PRINS, C. PRODHON, A GRASP×ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, **37**(11) (2010), 1912–1923.

[26] Y. MARINAKIS, Multiple Phase Neighborhood Search GRASP for the Capacitated Vehicle Routing Problem. *Expert Systems with Applications*, **39**(8) (2012), 6807–6815.

[27] KOHLI, RAJEEV, R. KRISHNAMURTI, A Total-Value Greedy Heuristic for the Integer Knapsack Problem. *Operations Research Letters*, **12**(2) (1992), 65–71.

[28] I. OSMAN, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, **41** (1993), 421–451

[29] S. KIRKPATRICK, C. D. GELATT, M. P. VECCHI, Optimization by Simulated Annealing. *Science*, **220** (1983), 671–680.

[30] N. CHRISTOFIDES, (ED.) ET AL., *Combinatorial Optimization*. John Wiley & Sons, Chicester, pp. 315–338.

[31] P. TOTH, D. VIGO, The granular tabu search and its application to the VRP. INFORMS. *J. Comput.*, **15** (2003), pp. 333–346.

[32] M. REIMANN, K. DOERNER, R. F. HARTL, D-Ants: Savings based ant divide and conquer the vehicle routing problem. *Computers & Operations Research*, **31** (2004), 563–591.

[33] C. PRINS, (2009) A GRASP x evolutionary local search hybrid for the Vehicle Routing Problem, Bio-inspired algorithms for the Vehicle Routing Problem, F.B. Pereira and J. Tavares (eds), Studies in Computational Intelligence, Springer, pp. 35–53,

Contact addresses:
Abdesslem Layeb
Computer Science Department
University of Constantine
Algeria
e-mail: `Layeb.univ@gmail.com`

Meryem Ammi
Computer Science Department
University of Constantine
Algeria
e-mail: `meryem.misc@gmail.com`

Salim Chikhi
Computer Science Department
University of Constantine
Algeria
e-mail: `chichi@umc.edu.dz`

ABDESSLEM LAYEB is an Associate Professor in the Department of Computer Science at the University of Constantine, Algeria. He is a member of MISC Laboratory. He received his PhD in Computer Science from the University of Constantine, Algeria. He is interested in combinatorial optimisation methods and their applications in solving several problems.

MERYEM AMMI is a PhD student in Computer Science at the University of Constantine. She is a member of the SCAL team of MISC Laboratory. Her research areas include soft computing and artificial life techniques and their application in solving vehicle routing problems.

SALIM CHIKHI received his PhD in Computer Science from the University of Constantine, in 2005. He is currently a Professor at the University of Constantine, Algeria. He is the Leader of the SCAL team of MISC Laboratory. His research areas include soft computing and artificial life techniques and their application in several domains.