GVR DELIVERS IT ON TIME

Jorge Tavares[1], Francisco B. Pereira[1,2], Penousal Machado[1,2], Ernesto Costa[1]

[1]Centro de Informática e Sistemas da Universidade de Coimbra, 3030 Coimbra, Portugal
[2]Instituto Superior de Engenharia de Coimbra, Quinta da Nora, 3030 Coimbra, Portugal
jast@student.dei.uc.pt, {xico, machado, ernesto}@dei.uc.pt

ABSTRACT

Genetic Vehicle Representation (GVR) is a new two-level representational scheme designed to encode all the information required by potential solutions for the vehicle routing problem. In a previous paper we described a set of experiments performed with several instances from the Capacitated Vehicle Routing Problem (CVRP). In this preliminary investigation, GVR proved to be both effective and robust.

In this work we extend the application of this new genetic representation to the vehicle routing problem with time windows, a variant that adds additional time constraints to the original definition. We present the results of a comprehensive set of tests that show that GVR is also efficient with this alternative, allowing the evolutionary computation algorithm to reach optimal solutions for some well know benchmarks.

1. INTRODUCTION

The Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem, which can be described as follows: given a fleet of vehicles with uniform capacity, a common depot, and several customer demands (represented as a collection of geographical scattered points), find the set of routes with overall minimum route cost which service all the demands. All the itineraries start and end at the depot and they must be designed in such a way that each customer is served only once and just by one vehicle. VRP is NP-hard, and therefore difficult to solve.

Due to the nature of the problem it is not viable to use exact methods for large instances of the VRP (for instances with few nodes, the branch and bound technique [1] is well suited and gives the best possible solution). Therefore, most approaches rely on heuristics that provide approximate solutions. Some specific methods have been developed to this problem (see, e. g., [2], [3]). Another option is to apply standard optimization techniques, such as tabu search [4], simulated annealing [4], [5], constraint programming [6], or ant colony optimization [7].

In the past few years there have also been some applications of evolutionary computation (EC) techniques to the VRP (see [8] for a good overview). Most researchers rely on hybrid approaches that combine the power of an EC algorithm with the use of specific heuristics (see, e.g., [9], [10]) or use simplified versions of the problem. One common simplification is to pre-set the number of vehicles that is going to be used in the solution [11], [12]. When applied alone, the success of EC techniques has been limited [13], [14].

Considering that the representation adopted for individuals plays a crucial role in the performance of an EC algorithm, in a previous paper we proposed a new representational scheme, Genetic Vehicle Representation (GVR) [15]. It was designed to deal efficiently with the two levels of information that a solution must encode: clustering of the demands (i.e., allocation of all the demands to different vehicles) and specification of the delivery ordering for each one of the routes. This representation also enables an easy adjustment of the number of vehicles required for one possible solution. The search process relies on standard EC techniques. It is important to notice that specific heuristics are not used, as well as any kind of simplification to the problem. Experiments performed with several CVRP instances from some well-known benchmarks showed that this approach is both effective and robust, since it allowed the discovery of new best solutions [15].

In this paper we apply GVR to one of the most important extensions to the VRP, the Vehicle Routing Problem with Time Windows (VRPTW). This variant introduces additional constraints to the original definition, since each customer must be served within a specific time window (i.e., for each node there is both an earliest and a latest time allowed for delivery).

The paper has the following structure: in section 2 we give a formal definition of the VRPTW. Section 3 comprises a description of the proposed EC model. In section 4 we present and examine the most important experimental results achieved. As a final point, in section

5, we illustrate some overall conclusions and propose directions for future work.

2. THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

The VRPTW can be formally described in the following way: there is one central depot 0, which uses k independent delivery vehicles, with identical delivery capacity C, to service demands d_i from n customers, i = 1, …, n. The vehicles must accomplish the delivery with a minimum total length cost, where the cost c_{ij} is the distance from customer i to customer j, with i, j $\in$ [1, n]. The distance between customers is symmetric, i.e., $c_{ij}=c_{ji}$ and also $c_{ii}=0$. The travel time from customer i to customer j is $t_{ij} = c_{ij}$. Time constraints specify that associated with each customer i there is a time window $[e_i, l_i]$ during which this customer has to be served. This way, a vehicle must arrive to i no sooner than e_i (the earliest arrival time) and no later than l_i (the latest arrival time). Additionally, there is a service time f_i for each customer and a limit T, defining the maximum travel time permitted for any vehicle.

A solution for the VRPTW would be a partition $\{R_1,\ldots,R_k\}$ of the n demands into k routes, each route R_q satisfying both $\sum_{p \in R_q} d_p \le C$ and the time constraints. Associated with each R_q is a permutation of the demands belonging to it, specifying the delivery order of the vehicles. In figure 1 we present an illustration of the problem, viewed as a graph, where the nodes represent the customers.

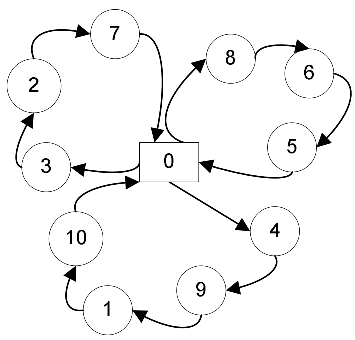

Figure 1: Vehicle Routing Problem.

3. EVOLUTIONARY COMPUTATION MODEL

3.1. Genetic Vehicle Representation

A candidate solution to an instance of the VRPTW must specify the number of necessary vehicles, the partition of the demands through all these vehicles and also the delivery order for each route. We adopted a representation where the genetic material of an individual contains several routes, each one of them composed by an ordered subset of the customers. All demands belonging to the problem being solved must be present in one of the routes. As an example, the chromosome from figure 2 represents the solution presented in figure 1.

Route 1	3	2	7	
Route 2	4	9	1	10
Route 3	8	6	5	

Figure 2: An example of a GVR chromosome.

The information encoded in the chromosome must be interpreted in such a way that it yields a legal solution. Two specific situations must be considered:

A vehicle exceeds its capacity: when this happens we split the route that exceeds capacity in several ones. An example illustrates this adjustment: assume that the original route {a, b, c, d, e, f} causes the vehicle to exceed its capacity at node d. When this situation occurs, the itinerary is divided in two sections: {a, b, c} and {d, e, f}, and a new vehicle is added to the solution. If necessary, further divisions can be made in the second section.

Time constraints are not satisfied: in this case, three different types of violation may occur: early arrival at a customer, late arrival at a customer or late arrival at the depot. The first situation is easily solved, since it's only required that the vehicle waits until it meets the earliest arrival time of the window. To resolve the other two cases, a new section is created on the itinerary, providing a valid route by adding a new vehicle to the solution.

Notice that these changes only occur at the interpretation level and, therefore, the information codified in the chromosome is not altered.

3.2. Genetic Operators

The EC algorithm processes the individuals in a straightforward way. Assuming that the population size is N, in each generation N parents are chosen and N descendants are obtained through the application of genetic operators to the elements of the selected set.

We consider two categories of operators: crossover and mutation. They must be able to deal with the two levels of the representation. Thus, they should be capable to change the delivery order within a specific route and to modify the allocation of demands to vehicles. In this last situation, they can, not only switch customers from one route to another, but also modify the number of vehicles belonging to a solution (adding and removing routes).

Another important requirement is that the genetic operators must always generate legal solutions.

The crossover operator used in our approach does not promote a mutual exchange of genetic material between two parents. Instead, when an individual from the selected set is submitted to this kind of operation, it receives a fragment of genetic material (more precisely, a route) from another parent and inserts it as the first route. After insertion, a repair process checks the original routes from the receiving individual and removes all customers that also appear in the inherited itinerary. This ensures that the new chromosome is legal, since there will be no repeated points. During the operation, the donor is not modified. The example from figure 3 illustrates how crossover acts. As you can see, it can, not only add a new route to a solution, but also remove some existing itineraries (in the example, original route 3 disappears).

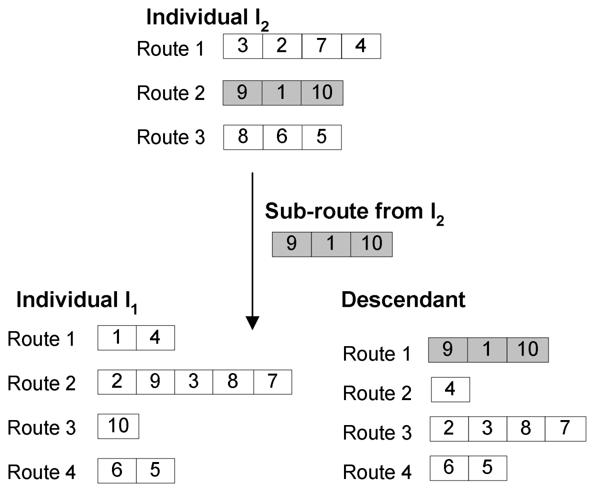

Figure 3: Example of crossover.

Descendants resulting from crossover can be subject to mutation. We consider four operators, based on proposals usually applied to order-based representations:

Swap: selects two customers and swaps them. Selected points can belong to the same or to different routes.

Inversion: selects a sub-route and reverses the visiting order of the customers belonging to it.

Insertion: selects a customer and inserts it in another place. The route where it is inserted is selected randomly. It is possible to create a new itinerary with this single customer. In all experiments reported in this paper, the probability of creating a new route is $1/(2\times V)$, where V represents the number of vehicles of the current solution. This way, the probability of creating a new route is inversely proportional to the number of vehicles already used. In figure 4 we show an example of this operation.

Displacement: selects a sub-route and inserts it in another place. This operator can perform intra or inter displacement (whether the selected fragment is inserted in the same or in another route). Just like in the previous operator, it is also possible to create a new route with the subsequence (the probability of this occurrence is calculated in the same way).

Swap and inversion do not change the number of routes of an individual. On the contrary, insertion and displacement have the ability to remove and to add vehicles to a solution. All genetic operators described have a specific probability of application to a single individual, not genes as in standard EC models.

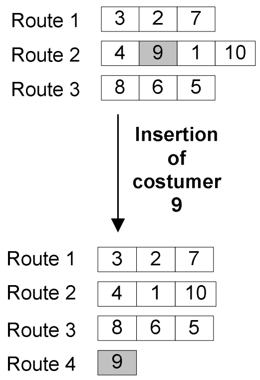

Figure 4: Example of insertion mutation applied over customer 9.

4. EXPERIMENTAL RESULTS

To evaluate our approach we performed an extensive collection of tests with 12 instances from Solomon's VRPTW benchmarks [16]. We choose four instances from each one of the three different sets of problems (C, R and RC) belonging to group 1 (this group contains instances with a short scheduling horizon, allowing only a few customers per route). All sets consist of 100 customers plus the depot, with different time windows width. Location of customers in instances from set R is randomly generated, whilst in set C, they are clustered. For instances of set RC, a mix of both distributions is used.

The settings of the EC model are the following: Number of generations: 50000; Population size: 200; Tournament selection with tourney size 5; Elitist strategy; Crossover rate: {0.25, 0.5}; Mutation rates: swap: 0.1; inversion: 0.1; insertion: {0.25, 0.5}; displacement: {0.25, 0.5}. For every set of parameters we performed 30 runs with the same initial conditions and with different random seeds. All initial populations were randomly generated, according to the algorithm in [15]. The values for different parameters were set heuristically.

Instances	GVR Best		Optimal		Heuristics Best		Previous EC Best	
	NV	Dist	NV	Dist	NV	Dist	NV	Dist
C101	10	827.3	10	827.3	10	828.94	10	828.94
C102	10	827.3	10	827.3	10	828.94	10	868.80
C103	10	826.3	10	826.3	10	828.06	11	939.46
C104	10	834.7	10	822.9	10	824.78	10	963.72
R101	21	1666.3	20	1637.7	19	1645.79	20	1676.86
R102	19	1486.1	18	1466.6	17	1486.12	17	1549.00
R103	15	1244.2	14	1208.7	13	1292.68	15	1311.81
R104	12	1024.8	n/a	n/a	10	982.01	10	1090.00
RC101	18	1671.2	15	1619.8	14	1696.94	17	1728.30
RC102	15	1502.5	14	1457.4	12	1554.75	14	1569.00
RC103	13	1353.8	11	1258.0	11	1261.67	14	1519.83
RC104	11	1179.9	n/a	n/a	10	1135.48	11	1263.00

Table 1: Summary of best solutions found. NV is the number of vehicles. Dist is the distance.

In table 1, we present, for all instances, the best solutions discovered by our approach and compare them with the optimal known values, the best solutions found by other heuristics and by previous EC models. (these values were collected from [1], [2], [4], and [16]). From the observation of table 1, it can be seen that GVR is an efficient representation:

Outperformed previous EC approaches: in all instances tested, GVR discovered solutions with better quality than those found by any other EC model.

Competitive against specific heuristics: in the majority of the instances (seven in twelve) it achieved better solutions than those obtained by specific heuristic algorithms. In the other situations, results found by GRV are less than 5% worst (the only exception is instance RC103, with 7%).

Reached optimal solutions: in most of the instances of the clustered set, namely C101, C102 and C103, the optimal solution was discovered.

In table 2 we show, for all settings tested, the value of the best solution, as well as the average of best solutions found in each of the 30 runs. A brief perusal of the results show that GVR is a robust approach. For instances belonging to different groups, our EC model was able to find good solutions. Moreover, the quality of the results achieved with different settings is similar. These results suggest that the GVR approach is not very sensitive to tuning details, such as the application rates of genetic operators.

The influence of crossover is not clear. In the experiments performed with CVRP, we verified that this operator was crucial to the discovery of good solutions [15]. On the contrary, in this variant with time windows, the importance of exchanging information between individuals is vague. A more carefully and thorough analysis of the role of crossover is required and will be carried out in a future publication.

	Displacement = 0.5						Displacement = 0.25					
	Insertion = 0.25			Insertion = 0.5			Crossover = 0.25			Crossover = 0.5		
Instances	NV	Best	Avg	NV	Best	Avg	NV	Best	Avg	NV	Best	Avg
C101	10	827.3	847.14	10	827.3	876.58	10	827.3	832.63	10	827.3	845.18
C102	10	827.3	872.03	10	827.3	896.15	10	827.3	866.24	10	827.3	869.44
C103	10	826.3	885.58	10	827.3	896.47	10	826.3	900.49	10	826.3	909.42
C104	10	850.9	914.31	10	860.7	939.07	10	854.8	916.82	10	834.7	941.96
R101	21	1677.8	1720.41	22	1695.1	1726.28	21	1683.4	1722.08	21	1666.3	1723.70
R102	18	1508.1	1541.52	19	1508.2	1550.24	18	1477.8	1538.89	19	1486.1	1536.98
R103	15	1244.2	1302.77	15	1267.3	1310.07	15	1244.0	1300.86	15	1267.7	1321.56
R104	12	1024.8	1078.02	12	1031.4	1077.61	12	1014.3	1071.29	13	1044.3	1085.37
RC101	18	1708.4	1777.86	18	1689.9	1768.54	18	1671.2	1769.07	18	1686.5	1770.28
RC102	16	1527.9	1597.78	15	1502.5	1589.36	16	1519.6	1612.80	16	1536.3	1611.79
RC103	13	1358.2	1407.81	13	1361.4	1407.01	13	1379.4	1423.77	13	1353.8	1415.48
RC104	12	1215.9	1273.92	12	1207.9	1267.29	12	1205.5	1273.40	11	1179.9	1278.32

Table 2: Best and average of the best solutions, found in each of the 30 runs.

5. CONCLUSIONS AND FUTURE WORK

In this paper we applied GVR, a new generic evolutionary approach to the VRPTW. The two-level representational scheme proved to be effective and robust on this variant of the problem as shown by the experimental results. Even though results can be considered as preliminary, GVR enabled us to discovery competitive solutions for a set of instances from well-known benchmarks.

As future work we intend to test GVR on all instances of Solomon's VRPTW benchmarks, in order to perform a thorough study of its robustness and effectiveness. A detailed analysis on the importance of genetic operators, with special relevance to crossover is fundamental, since its role is unclear.

6. REFERENCES

[1] J. Desrosiers, O. Madsen, M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows," *Transportation Science*, Vol. 33, No. 1, pp. 101-116, 1999.

[2] S. Thangiah, J. Potvin, and T. Sun, "Heuristic Approaches to Vehicle Routing with Backhauls and Time Windows," *Int. Journal of Computers and Operations Research*, pp. 1043-1057, 1996.

[3] P. Prosser, and P. Shaw, "Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems," *Technical Report RR/96/201*, Department of Computer Science, University of Strathclyde, Glasgow, 1997.

[4] K. C. Tan, L. H. Lee, Q. L. Zhu, and K. Ou, "Heuristic Methods for Vehicle Routing Problem with Time Windows," *Artificial Intelligent in Engineering*, pp. 281-295, 2000.

[5] R. Bent, and P. V. Hentenryck, "A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows," *Technical Report, CS-01-06*, Brown University, 2001.

[6] P. Shaw, "Using Constraint Programming and Local Search Methods to solve Vehicle Routing Problems," *Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming (CP '98))*, M. Maher and J. Puget (eds.), pp. 417-431, 1998.

[7] L. M. Gambardella, E. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds.), pp. 1043-1057, 1999.

[8] O. Bräysy, "Genetic Algorithm for the Vehicle Routing Problem with Time Windows," *Arpakannus 1/2001, Special Issue on Bioinformatics and Genetic Algorithms*, pp. 33-38, 2001.

[9] S. R. Thangiah, "Vehicle Routing with Time Windows Using Genetic Algorithms," *Application Handbook of Genetic Algorithms: New Frontiers*, Volume II, L. Chambers (ed.), pp. 253-277, CRC Press, 1995.

[10] J. Potvin, D. Dubé, and C. Robillard, "Hybrid Approach to Vehicle Routing Using Neural Networks and Genetic Algorithms," *Applied Intelligence*, Vol. 6, No. 3, pp. 241-252, 1996.

[11] K. Zhu, "A New Genetic Algorithm for VRPTW," *International Conference on Artificial Intelligence*, Las Vegas, USA, 2000.

[12] S. J. Louis, X. Yin, and Z. Y. Yuan, "Multiple Vehicle Routing With Time Windows Using Genetic Algorithms," *Proceedings of the Congress of Evolutionary Computation (CEC-99)*, pp. 1804-1808, 1999.

[13] T. Duncan, "Experiments in the Use of Neighbourhood Search Techniques for Vehicle Routing," *Report AIAI-TR176*, University of Edinburgh, 1995.

[14] P. Machado, J. Tavares, F. B. Pereira, and E. Costa, "Vehicle Routing Problem: Doing it the Evolutionary Way," *To appear at GECCO-2002 Proceedings*, 2002.

[15] F. B. Pereira, J. Tavares, P. Machado, and E. Costa, "GVR: a New Genetic Representation for the Vehicle Routing Problem," *Submitted to AICS 2002 13th Irish Conference on Artificial Intelligence and Cognitive Science*, Limerick, Ireland, 2002.

[16] http://web.cba.neu.edu/~msolomon/problems.htm