

Solving the Capacitated Vehicle Routing Problem and the Split Delivery Using GRASP Metaheuristic

Joseph Gallart Suárez and Manuel Tupia Anticona

Engineering department, Pontificia Universidad Católica del Perú, Av. Universitaria 1801,
San Miguel, Lima, Perú
{jgallart,tupia.mf}@pucp.edu.pe

Abstract. This paper presents two GRASP metaheuristic algorithms for the vehicle routing problem, considering the capacity and shared demand of the customers. In this paper the solution obtained is compared with a greedy solution and two hybrid solutions (greedy and random). The results obtained show that the GRASP algorithm obtains a better quality solution for this kind of problem.

Keywords: Capacitated Vehicle Routing Problem, Split Delivery Vehicle Routing Problem, Metaheuristic, GRASP.

1 Introduction

The Vehicle Routing Problem (VRP) is probably the best known node routing problem nowadays. It is generally defined as a graph $G = (V, D)$ with a set of nodes $V = \{v_0, v_1, v_2 \dots v_n\}$ a set of edges D , where v_0 represents the depot, with K identical delivery vehicles with Q capacity, the other nodes represent the consignee customers with a demand q_i and each edge (i, j) has a symmetrical cost $c_{ij} = c_{ji}$ [1].

The problem consists in determining a group of K routes for K vehicles, where total cost is minimal, beginning and ending in a depot; such that each node is visited just one time and each delivery vehicle does not surpass its Q capacity [2].

A particular variant of the previously explained scenery is to add delivery vehicle capacity as a restriction (CVRP), besides, in this paper, split delivery is added as a second restriction (SDVRP); this last consideration applies when the consignee customer has bigger demand than the delivery vehicle capacity, thus a consignee customer can be visited by more than one delivery vehicle.

Since the introduction of the proposed problem by Dantzig and Ramser [3], the evolution of solutions, both exact and heuristic, has improved notoriously, among heuristic solutions the works of Laporte, Mercure and Nobert [4]; Fischetti, Toth and Vigo [5]; and Fisher [6] stand out, proposing Ramification and quotation algorithms. We have to mention that Fisher's method solves up to 71 destiny nodes (customers). However, this demands a big computational cost [2].

In the exact ramification and cut algorithm category the work of Cornuéjols, Fonlupt and Naddef [7]; Naddef and Rinaldi [8]; Toth and Vigo [9] stand out, however, this kind of algorithm depends a lot on the particular structure of the VRP since it can complicate the solution if cuts are applied to a structure that does not favor the algorithm [10].

Within the most outstanding heuristic constructive methods we have Clarke and Wright's saving algorithm [11], and the saving algorithm based on matching, as the works of Desrochers and Verhoog [12] and Altinkemer and Gavish [13].

In the category of algorithms with heuristic insertion methods the works by Mole and Jameson [14]; Christofides, Mingozzi and Toth [15]; Solomon [16] stand out, these algorithms start with initially empty routes and evaluate the best form to insert a node in any route iteratively, remaining with a pair (node, route) that represents the best insertion.

Nowadays, we have metaheuristic methods that obtain very good solutions for the proposed problem, such is the case of the ant colony algorithm [17], genetic algorithms [18] or the Tabu search algorithm [19].

In this document, we present two GRASP metaheuristic algorithms that allow us to solve the previously mentioned problem and 3D bin packing problem (3D-BPP) within the bins of the delivery vehicles, besides, we try to improve the solution, optimizing it through a 2-Opt local search algorithm. The combination of the two previously mentioned problems constructs a real transport and package packing problem.

This paper is organized as follows: in section 2 the algorithms we used are described, in section 3 the tests and the results obtained are shown, finally, the conclusions obtained and future work are shown in section 4.

2 Description of the Algorithms

2.1 GRASP Algorithm

The GRASP algorithm is an aleatory and adaptative short-sighted search procedure that helps us to find good quality solutions for combinatory optimization problems [20]. This algorithm is a multistart method where each iteration contains two phases: construction and improvement.

In the construction phase, a solution is constructed by relaxing the greedy criteria. In order to do so, an element is iteratively added in each step, the election of each element (S_i) is determined by three elements: the objective function (f_o), the viability range (E) and the restricted list of candidates, which is defined by three factors: the relaxation constant (α), the best value of the objective function (β) and the worst value of the objective function (τ) [20]. The restricted list of candidates (RCL) to minimize cases and to maximize the objective function respectively is detailed in equations 1 and 2.

$$\text{RCL} = \{S_i \in E: \beta \leq f_o(S_i) \leq \beta + \alpha(\tau - \beta)\}$$

Equation 1. Restricted list of candidates in case minimization of the objective function is searched

$$\text{RCL} = \{S_i \in E: \tau + \alpha(\beta - \tau) \leq f_o(S_i) \leq \beta\}$$

Equation 2. Restricted list of candidates in case maximization of the objective function is searched

An element is extracted aleatorily out of this group of candidates, thus forming a possible solution.

In the improvement phase the solution generated by the construction phase is optimized; in order to do so, an iterative search is performed replacing successively the present solution by one that is in the group of neighborhood solutions. When a solution that improves the present one is not found, it is said that the solution is locally optimal.

These two phases are repeated several times and the best solution found after all GRASP iterations is considered the best quality solution [20].

In this paper we present 2 GRASP algorithms that are detailed bellow:

2.1.1 First GRASP Algorithm

The first GRASP algorithm provides an initial solution for the capacitated vehicle routing problem, taking split delivery as second restriction. Within the construction phase a second GRASP algorithm is applied that solves the 3D bin packing problem which helps us to optimize packing in the vehicle, considering vehicle load weight and the order of delivery of the consignees.

This algorithm receives two entry parameters: a graph (conformed by the starting node and the consignee nodes) and the delivery vehicle list; the algorithm returns the delivery route to be followed by each vehicle.

For the construction of the first GRASP algorithm we use an objective function that has three entry parameters: the traveled distance of the edge, the traffic constant (a value between 1 and 5) and the cost function of the second GRASP algorithm¹. The objective function to be minimized is shown in equation 3.

$$F_{(objective\ GRASP-1)} = \frac{Distance * TrafficConstant}{F_{(cost\ GRASP-2)}}$$

Equation 3. Objective function of the first GRASP algorithm

To know the quality of the solution of the algorithm we apply a cost function that has four entry parameters: the summation of the distance traveled by all the vehicles in the route, the average of the traffic constant of all the avenues that form the delivery vehicle route, the quantity of vehicles used to perform the delivery and the average of the cost function of the second GRASP algorithm². The cost function to be minimized is shown in equation 4.

$$F_{(Cost\ GRASP-1)} = \frac{\sum Distance * TrafficConstant * Quantity\ of\ vehicles}{\overline{F_{(cost\ GRASP-2)}}$$

Equation 4. Cost function of the first GRASP algorithm

¹ The cost function of the second algorithm is detailed in section 2.1.2.

² The cost function of the second algorithm is detailed in section 2.1.2.

2.1.2 Second GRASP Algorithm

The second GRASP algorithm is applied in the construction phase of the first GRASP algorithm. This algorithm provides a solution for the 3D bin packing problem. This algorithm receives two entry parameters: the delivery vehicle and the list of packages of the consignees; the algorithm returns the vehicle with the configuration of packed packages according to the space (in three dimensions) and the load weight of such vehicle.

For the construction of the second GRASP algorithm an objective function that has six entry parameters is used: package measurements (width, length and height), the width and the length of the cut region of the vehicle to be examined (*WRVehicleCut*, *LRVehicleCut*) and, finally, vehicle height. The objective function to be maximized is shown in equation 5.

$$F_{(Objective\ GRASP-2)} = \frac{widthPackage * lengthPackage * heightPackage}{WRVehicleCut * LRVehicleCut * heightVehicle}$$

Equation 5. Objective function of the second GRASP algorithm

In this second algorithm a cost function that has 4 entry parameters is applied: the volume used to pack the packages in the vehicle (*Vol Tp*), total storage space volume of the vehicle (*Vol Veh*), the summation of the weight of all the packages packed in the vehicle (*Weight Tp*) and the load weight that the vehicle supports (*Weight Veh*), in this function the priority is to optimize the volume, that is why such factor is doubled. The cost function to be maximized is shown in equation 6.

$$F_{(Cost\ GRASP-2)} = \left(\frac{(Vol\ Tp / Vol\ Veh) * 2 + Weight\ Tp / Weight\ Veh}{3} \right) * 100\%$$

Equation 6. Cost function of the second GRASP algorithm

2.2 Local Search Algorithm: 2-OPT

The 2-OPT algorithm is a local search based improvement algorithm, the procedure to be followed is as follows: two groups of edges $\{x_1, \dots, x_k\}$ and $\{y_1, \dots, y_k\}$ are determined, such that their interchange reduces the solution cost. The arcs “x” must be part of the route and both groups must be disjunct, besides, when eliminating the arcs “x” and adding the “y” arcs they must form a closed route [21].

This algorithm receives three entry parameters: a graph (conformed by the start node and the consignee nodes) and the list of delivery vehicles with the route of the consignees; the algorithm returns a list of vehicles with the optimized routes.

3 Test Execution and Results

The algorithms were implemented in Java and the tests were executed in an Intel Core 2 Duo Computer with 2.0 GHz of processing speed, 2 GB RAM and with a Windows XP SP2 operating system.

A certain quantity of graphs and random nodes were used following the pattern described by Christofides, Mingozzi and Toth for CVRP problems [2]. 300 random graphs with 50, 100 and 200 consignee nodes with random packages and random weight and measures were generated. In Table 1, the parameters used by the two GRASP algorithms are shown.

Table 1. Parameters of the GRASP algorithms

	n=50	n= 100	n=200
#Iterations to GRASP-1	50	50	40
#Iterations to GRASP-2	100	100	80
Relaxation constant to GRASP-1 (α)	0.2	0.2	0.2
Relaxation constant to GRASP-2 (α)	0.8	0.6	0.8

The objective of the tests is to compare the two GRASP algorithms and 2-Opt optimization (acronym G-2OPT) against greedy algorithms and hybrid algorithms (greedy and random) described in Table 2.

Table 2. Acronyms of the algorithms

First algorithm	Second algorithm	Acronym
Greedy algorithm	Greedy algorithm	GG
Greedy algorithm	Random algorithm	GR
Random algorithm	Greedy algorithm	RG

In Tables 3, 4 and 5 the average results obtained by the algorithms according to their execution time (in seconds) and cost (detailed in equation 4) are shown.

Table 3. Result for 50 consignee nodes

	G-2OPT	GG	GR	RG
Average time(s)	306.30	403.20	429.84	532.36
Average cost	2175.29	2283.96	3225.82	4385.58

Table 4. Result for 100 consignee nodes

	G-2OPT	GG	GR	RG
Average time (s)	772.44	925.53	961.17	1216.25
Average cost	17583.11	18516.72	20223.74	29546.84

Table 5. Result for 200 consignee nodes

	G-2OPT	GG	GR	RG
Average time (s)	1849.56	2620	2838	3563.11
Average cost	96193.21	118110.66	144171.91	214684.65

The three tables indicate that the two GRASP algorithms with 2-OPT optimization obtain better quality solutions than the rest of algorithms, in the 3 tables the Greedy algorithm obtains better solutions than the two Hybrid algorithms. In Table 3 the proposed algorithm (G-2OPT) and the greedy algorithm obtain very close solutions (in cost and time), however, for 100 and 200 nodes the proposed algorithm renders a solution of better quality.

We can observe that the Random-Greedy algorithm (RG) presents a higher cost and more execution time than the rest of algorithms, this is because it parts from an random solution, therefore it demands higher cost and more time to find a solution.

4 Conclusions and Future Work

In this work we implemented 2 GRASP algorithms with 2-OPT optimization to solve the capacitated vehicle routing problem, considering the split delivery restriction and solving, besides, the 3D bin packing problem, the combination of both problems let us have a more realistic model of the transport and package loading problem.

Computing tests were performed to study the aplicability and the quality of the solution of the two algorithms proposed, the computational results indicate that the two GRASP algorithms with 2-OPT optimization obtain better solutions than a Greedy algorithm and against two hybrid algorithms.

As future work we have to implement other metaheuristic algorithms, as the Ant Colony and the Genetic algorithms, compare them with the solution that was proposed and to verify if it obtains the best solutions for the proposed problem.

References

1. Prins, C.: A GRASP X Evolutionary Local Search Hybrid for the Vehicle Routing Problem. In: *Bioinspired Algorithms for the Vehicle Routing Problem*, vol. 1, p. 35 (2008)
2. Universidad de Málaga (2009), <http://neo.lcc.uma.es/radi-aeb/WebVRP/>
3. Marinakis, Y., Migdalas, A.: *Operational Research* 7(1), 27–46 (2007)
4. Laporte, G., Mercure, H., Nobert, Y.: An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks* 16, 33–46 (1986)
5. Fischetti, M., Toth, P., Vigo, D.: A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research* 42, 846–859 (1994)
6. Fisher, M.L.: Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research* 42, 626–642 (1994)
7. Cornuéjols, G., Fonlupt, J., Naddef, D.: The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming* 33, 1–27 (1985)
8. Naddef, D., Rinaldi, G.: The graphical relaxation: A new framework for the symmetric traveling salesman polytope. *Mathematical Programming* 58, 53–88 (1993)

9. Toth, P., Vigo, D.: *The Vehicle Routing Problem*. Society of Industrial and Applied Mathematics (SIAM), Philadelphia (2002)
10. Claudio Contardo Vera: *Formulación y solución de un problema de ruteo de vehículos con demanda variable en tiempo real, trasbordos y ventanas de tiempo*. Universidad de Chile (2005)
11. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, 568–581 (1964)
12. Desrochers, M., Verhoog, T.W.: A new heuristic for the fleet size and mix vehicle routing problem. *Computers & OR* 18, 263–274 (1991)
13. Altinkemer, K., Gavish, B.: Parallel Savings Based Heuristics for the Delivery Problem. *Operations Research* 39(3), 456–469 (1991)
14. Mole, R.H., Jameson, S.R.: A sequential route-building algorithm employing a generalized savings criterion. *Operations Research Quarterly* 27, 503–511 (1976)
15. Christofides, N., Mingozzi, A., Toth, P.: *The vehicle routing problem*. Wiley, Chichester (1979)
16. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35 (1987)
17. Dong, L.W., Xiang, C.T.: Ant Colony Optimization for VRP and Mail Delivery Problems. In: 2006 IEEE International Conference on Industrial Informatics, pp. 1143–1148 (2006)
18. Ombuki-Berman, B., Hanshar, F.T.: Using Genetic Algorithms for Multi-depot Vehicle Routing. *Studies in Computational Intelligence* 161, 77–99 (2009)
19. Wassan, N.A., Wassan, A.H., Nagy, G.: A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization* 15(4) (2008)
20. Resende, M.G.C., Velarde, J.L.G.: GRASP: Greedy Randomized Adaptive Search Procedures. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 19, 61–76 (2003)
21. Olivera, A.: *Heurísticas para Problemas de Ruteo de Vehículos*. Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay (2004)