Lecture Notes in Management Science (2013) Vol. **5**: 84–92
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

Applying a savings algorithm for solving a rich vehicle routing problem in a real urban context

José Cáceres-Cruz [1], Daniel Riera [1], Roman Buil [2] and Angel A. Juan [1]

[1] *IN3-Computer Science Department, Open University of Catalonia, Barcelona, Spain*
jcaceresc@uoc.edu; drierat@uoc.edu; ajuanp@uoc.edu

[2] *Department of Telecommunications and Systems Engineering,*
Universitat Autònoma de Barcelona, Bellaterra, Spain
roman.buil@uab.cat

Abstract. Nowadays urban transportation is a strategic domain for distribution companies. In academic literature, this problem is categorized as a Vehicle Routing Problem, a popular research stream that has undergone significant theoretical advances but has remained far from practice implementations. In fact, a general combinatorial routing problem has emerged as Rich Vehicle Routing Problem for considering problems inspired in real situations. Intra-urban distribution required a special combination of routing characteristics. In this study, we consider a routing problem with asymmetric cost matrix, heterogeneous fleet of vehicles, service times, limited routes length, open routes, and balanced loads in routes' restrictions. Our objective function is to reduce the total traveling time. We present an algorithm based on a randomized Clarke & Wright's Savings heuristic. We execute our algorithm with data from a company that distributes prepared food to more than 50 customers in Barcelona. The results reveal promising improvements in different scenarios.

Keywords: rich vehicle routing problem; clarke and wright; heuristics

Introduction

Vehicle routing is a complex logistics management problem and represents a key phase for the logistic optimization. We have considered a variant where several restrictions are considered at the same time. The set of defined constraints are taken from a real case provided by a food distribution company located in Barcelona, Spain. The distribution inside cities has special conditions like little time for delivery, congestion, traffic lights, and different types of vehicles related to the size and velocity issues. In fact, the asymmetric nature of road network produces a remarkable effect in distribution (Rodríguez & Ruiz, 2012). There are many possible routes to visit a customer because the street direction creates a special network of available arcs. In reality, most companies use different types of vehicles to distribute their products for several reasons. Notice that the open routes feature creates some flexibility on selecting the ending point of routes (Li et al., 2007). For some enterprises, these constraints could be interesting. The purpose of this study is to apply a randomized savings algorithm based on a savings heuristic for a specific Rich Vehicle Routing Problem (RVRP) with several restrictions. These are asymmetric cost matrix, heterogeneous fleet of vehicles, service times, limited routes length, and optionally open routes or balanced loads in routes restrictions. This RVRP study searches to create alternative routing plans for the decision-making process. The paper is organized as follows: Section 2 describes the theoretical background and previous works. In Section 3 we present an overview of the proposed algorithm. Section 4 presents the data instances from the distribution company. Section 5 shows the results of applying the proposed methodology to a real context case. To conclude, Section 6 summarizes with final remarks.

The rich vehicle routing problem

The Vehicle Routing Problem (VRP) has been studied for over more than 50 years (Laporte, 2009). The simplest version is known as the Capacitated Vehicle Routing Problem (CVRP), defined by (Dantzig and Ramser, 1959). In CVRP, a directed graph $G = (V, A)$ is given, where $V = \{0, 1, \ldots, n\}$ is the set of $n + 1$ nodes and A is the set of arcs. *Node 0* represents the depot, while the remaining nodes $V' = V \setminus \{0\}$ correspond to the n customers. Each customer i in V' requires a known supply of q_i units, i.e., its demand, from a single depot (assume $q_0 = 0$). This demand is going to be served by exactly one visit of a single vehicle. In this basic form, there is a homogeneous fleet of m identical vehicles with capacity Q to serve these n customers. Each vehicle has also a time limit L for their single trip. A vehicle's trip is a sequence of customers, whose total demand cannot exceed Q that starts from and finishes at the depot with duration no greater than L (used to be a really big value in order to ignore its effect). CVRP aims at finding m trips (vehicles) so that all customers are serviced and the total distance traveled by the fleet is minimized. In the asymmetric version of the CVRP, the cost of each arc (i, j) can have a different value for the inversed direction arc (j, i), i.e. $\exists i, j\,;\, i \neq j;\, c_{ij} \neq c_{ji}$. In general, a heterogeneous vehicle fleet M is composed by m different vehicle types, i.e., $M = \{1, \ldots, m\}$. For each vehicle type, there are m_k vehicles, a number that might be very large or, essentially, unlimited. The m_k vehicles of type $k \in M$ have capacity Q_k, fixed cost F_k, and variable cost per arc (i, j) traveled c_{ijk} $(i \neq j)$. The number of trips performed by type k vehicles must not be greater than m_k. The cost of a route results is then related to the costs of arcs included in the route and the vehicle costs. In this paper, we consider a heterogeneous fleet with the following additional considerations regarding the available fleet and its costs: the number of vehicles of each type, m_k, is limited (fixed fleet) and their use must be determined. For each vehicle type, its fixed costs are ignored and its routing costs are vehicle-independent. However other routing constraints are considered. The special feature of open routes can be represented as the exclusion of the cost of all returning edges ($c_{i0} = 0$). Then the original condition is also affected since all routes do not finish in the depot.

Realistic routing constraints are summarized in (Drexl, 2012). Then, more flexible and "rich" VRP approaches are needed. A preliminary definition of the Rich VRP (RVRP) has been made by (Toth & Vigo, 2002). The authors define the potential of extending the "vehicle flow formulations, particularly the more flexible three-index ones". Years later, the enterprise needs have created a broad context for this concept. Also a wide classification of the Rich VRP variants is presented in a special issue published by (Hartl et al., 2006). On this, seven papers were selected for covering different aspects of ampleness and illustrating novel types of VRPs. The editors state that "VRP research has often been criticized for being too focused on idealized models with non-realistic assumptions for practical applications". Several studies have addressed the Rich VRP in various ways. For instance, (Goel & Gruhn 2008) considers real-life requirements -e.g. time window restrictions, a heterogeneous vehicle fleet with different travel times, travel costs and capacity, multi-dimensional capacity constraints, order/vehicle compatibility constraints, orders with multiple pickup, delivery and service locations, different start and end locations for vehicles, and route restrictions for vehicles. The authors use Variable and Large Neighborhood Searches. (Hasle & Kloster 2007) presents a generic solver based on a combined interaction of a Variable Neighborhood Descent and an Iterated Local Search. They address the capacitated constraint, the distance limitation, the pickup-and-delivery, the fleet size and mix problem as well as the time windows. They present the possibility to extend it for multi-depot and site-dependent problems. (Pellegrini et al 2007) have studied a case characterized by multiple objectives, constraints concerning multiple time windows, heterogeneous fleet of vehicles, maximum duration of the sub-tours, and periodic visits to the customers. They considered two versions of Ant Colony Optimization (ACO). The authors compared the results obtained with a Tabu Search algorithm and a Randomized Nearest Neighborhood heuristic. (Vidal et al. 2013) applies a Hybrid Genetic Search. The approach is addressed to any combination of periodic, multi-depot, site-dependent, and duration-constrained with time windows. Some real applications can be found in the literature. For instance, (Prescott-Gagnon et al. 2010) present a real-inspired oil distribution which presents a heterogeneous vehicle fleet, multiple depots, intra-route replenishment, time windows, driver shifts and optional customers. (Oppen et al. 2010) considers a real scenario which includes duration and capacity restrictions, heterogeneous fleets, time windows, multi-trips and multi-products issues.

Randomized based-Savings algorithm

Our approach is based on the algorithm called Simulation in Routing via the Generalized Clarke and Wright Savings heuristic (SR-GCWS) proposed by (Juan et al., 2010). A multi-start process is executed until a stopping condition is not satisfied (*maxTime* parameter as a specific execution period of time). At each iteration, a solution is constructed using a randomization version of the classical parallelized Clarke and Wright Savings (CWS) heuristic (Clarke and Wright, 1964). CWS is probably one of the most cited heuristic to solve the CVRP. This procedure uses the concept of savings. On general, at each step of the solution construction process, the edge with the most savings is selected if and only if the two corresponding routes can feasibly be merged using the selected edge. This heuristic starts with the creation of a dummy solution with independent round-trips to each customer (most expensive solution). Then these routes are progressively joined between them.

The original version of CWS is based on the estimation of possible savings originated from merging routes, i.e., for unidirectional or symmetric edges: $Sav(i, j) = c_{0i} + c_{0j} - c_{ij}$. These savings are estimated between all nodes, and then decreasingly sorted. Then the bigger saving (at the top of the list) is always taken, and used to merge the two associated routes. On the randomized version of this algorithm, we use a pseudo-geometric distribution to induce a biased randomization selection of savings. Moreover, this selection probability is coherent with the savings value associated with each edge, i.e., edges with higher savings will be more likely to be selected from the list than those with lower savings. Therefore, each combination of edges has a chance of being selected and merged with previously built routes.

We make an initial distinction regarding the open routes requirement. If it is the case, we set to 0 the cost of all edges going to the depot. The purpose of this is to ignore returning edges in the route construction process including the dummy solution. The savings construction is modified for being applied to both contexts the asymmetric and open routes contexts. First, the inversed edges must be also considered in the set of eligible options (multiplying the original quantity on the symmetric version by two), i.e., for two different nodes i and j: $Sav(i, j) = c_{i0} + c_{0j} - c_{ij}$ as well as for $Sav(j, i)$. Then the commented asymmetric savings concept for the open routes case will be $Sav(i, j) = c_{0j} - c_{ij}$. The edge for going to the depot is excluded from the merging or construction of routes. Therefore, all savings will be competing to be taken in the biased randomized process, and those with higher savings will define the orientation of routes. Once a saving edge is selected and successfully used to merge to given routes, the opposite edge must be also removed from the savings edge list, in order to save computational time. Likely the routes construction process will consider the direction of savings edges. Once a route takes a direction then all considered candidate routes to be merged with the first one must follow the same direction.

In **Fig 1**, a simplified example is depicted in order to give an idea of the route construction process under the given routing constraints. On this directed graph, we have two open routes and two possible savings edges to be considered (*A* and *B*). Then it is easy to appreciate that the savings value related to *B* is better than *A*. So the new route will be made considering saving edge B. Notice that resulting routes will join routes where the first customer is near to the last visit of other route.

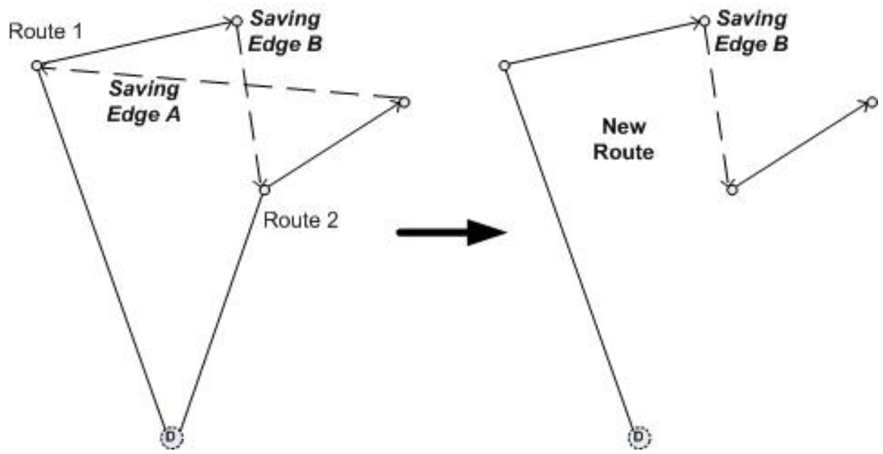

Fig. 1. Example of saving edge merging decision in an open routes context.

For considering the balanced loads in routes, we add another validation aspect in the merging step of the CWS process. Once the inputs are read, a maximum load limit per route is estimated using the total requested demand on the instance as well as a number of desirable routes indicated as a new parameter (*maxRoutes*). This last parameter can be set to two in order to try to find the minimum number of routes with balanced loads, as we did. This load limit is then adjusted with a percentage range (*perRoutes*) in order to allow a flexible criterion in the route construction. This value will serve as a basic limit for checking capacity when two routes are merged.

Inside of the construction process, the total route duration (traveling plus service times), the maximum load limit, and a candidate vehicle responsible of the new route are validated. The bigger vehicle between the two processing routes will be responsible of the new route. If a route does not have an assigned vehicle, then the first vehicle on the available vehicle list (decreasingly sorted by capacity) is selected. For this, several fictitious vehicles will be required mainly at the beginning of the CWS process. The fictitious vehicle should be defined using the minimum possible capacity on the instance. At the end, the fictitious vehicles must be discarded. Notice that any individual demand can be carried out by any truck (even the smallest and fictitious).

Once a solution is obtained, then it is improved with a local search method based on a memory cache (Juan et al., 2011). This technique keeps in memory the best known routes so far with the different combination of customers. This procedure compares and saves the best order for visiting the nodes on all solutions generated so far. The explained approach represents the final approach of the preliminary algorithm presented in Caceres-Cruz et al. (2013), which considers a reduced set of routing constraints.

Company instances

As the case of study, we used the information of a prepared food distribution company located in Barcelona, Spain. The company has provided us with the delivery address of their customers in six independent days along with their demands for those days. The transportation limits are defined inside of the city borders (urban distribution). On this context the orientation of streets, traffic lights and traffic itself have a remarkable influence in the daily routing planning.

The main interest of the company is to apply the proposed approach to bigger datasets using a web information tool. For this reason, the company just compile the information during a short period (as a sample) in order to produce a preliminary result. Therefore on a daily basis, this company receives requests from around 50 customers. So far, this information serves as input to manually design the company's routing planning. According to the size of the company it is not possible to employ a person specialized in mathematical software in order to apply exact methods. Therefore they prefer to have an approximated solution algorithm embed in a web tool which could be used to give automatic solution in little time. Regarding a future increase on demands, the company is mainly interested in building a set of alternative routing solutions. These solutions can include a subset of the previously specified restrictions. The restrictions can be separated as mandatory for all scenarios (asymmetric cost matrix, heterogeneous fleet of vehicles, service times at customers and limited routes length) and optional (open routes, and balanced loads). These last constraints create new scenarios for routing planning which are the main contributions of this study. In fact, the company is especially interested in the open routes option because their drivers can take delivery vehicles with them. So the time for going to the parking place and going to the depot point (on the next day) is not counted for the delivery process. For other part, the balanced loads constraint represents an equally working condition between drivers.

All orders arrive in the early morning, and between noon and 15:00 all must be delivered. Then there is a specific constraint: each vehicle must visit all customers of a route in a maximum period of 180 minutes. This route length restriction must to include the traveling time and the service time. So far, the company uses two types of vehicles, which are described in the **Table 1**. The columns of this table show the capacity (Q_k) and quantity (m_k) of available vehicles for each type (k). Actually the company used four vehicles, but they needed to determine if it is possible to reduce the total routing costs (traveling time) and also execute the same deliveries with fewer routes.

The main features of given six data instances are summarized in the **Table 2**. On the first column, we present the identification of each instance that represents a day. The second column shows the number of customers with demands. Third column is the total demand. And the last column represents the total service time of all the nodes on the instance. We have used a map-location service, like Google Maps to generate the asymmetric cost matrix between every pair of nodes. Although this kind of routing tool considers all possible streets of the city, the cost matrix will only represent the best traveling time between each two nodes. This will serve as an approximation solution for testing our approach. So real source information is required in order to produce a proper solution in a given day. Finally the company provides us with the service times' historic data of some routes. However this was incomplete. Notice that the company did not save exact information of all their routes, even within a whole day. Likely they do not apply any specific routing method. A person in charge designs the routing planning. Therefore we have randomly generated the respective values, using simulation theory (Monte Carlo Simulation). Then we have defined that the service time for each client follows a triangular distribution with the next parameters: min = 1, max = 12 and mode = 3 (the measure unit for all is minutes). The triangular probability distribution is often used to represent time in general simulation models.

Table 1: Composition of the current company fleet.

Vehicle Type k	Q_k	m_k
1	20	2
2	30	2

Table 2: General features of real instances.

Instance (day)	*Number of Customers*	*Total Requested Demand*	*Total Service Time (min)*
A	40	53	163
B	50	75	213
C	40	60	163
D	39	54	159
E	40	57	162
F	18	28	75

Results

Our algorithm was implemented as a Java application and used to run the six instances on an Intel Xeon E5603 at 1.60 Ghz and 8 GB RAM. For each instance, a single run with a total maximum time (*maxTime*) of 500 seconds was employed. For *perRoutes* parameter, we use a range value of 10%.

Table 3 shows the results obtained in experiments. We use the average information for comparing several scenarios: (a) Current company solutions; (b) previously Best found solutions in Caceres-Cruz et al. (2013); (c) solutions allowing only Open routes; (d) solutions only balancing the total load to 3 routes; (e) solutions balancing the total load to 3 routes and also allowing open routes; (f) solutions only balancing the total load to 2 routes; and (g) solutions balancing the total load to 2 routes and also allowing open routes. For each of these, we present the average number of routes, the average traveling time cost (minutes), the average total cost (minutes), average percentage of used capacity in assigned vehicles, average load per route, and average CPU time until the solution is found (seconds).

Table 3: Different solution scenarios combining real constraints.

Scenario	*Routes*	*Traveling Cost (min)*	*Total Cost (min)*	*Load per Route*	*% of Used Capacity*	*Time (sec)*
Current	4.00	192.83	348.67	54.50%	13.63	NA
Best	2.17	167.67	323.50	85.63%	25.17	130.50
Open	2.83	144.83	300.67	70.07%	18.94	174.98
Balanced 3 routes	3.17	182.00	337.83	65.00%	17.13	129.81
Open-Balanced 3 routes	3.67	147.50	303.33	59.25%	15.60	248.65
Balanced 2 routes	2.17	168.83	324.67	85.63%	25.17	162.34
Open-Balanced 2 routes	3.00	144.67	300.50	68.13%	18.17	232.76

As it can be appreciated, the Best scenario generated in the first experiments reduces total costs as well as routes, where the used percentage of vehicle capacities is the higher obtained value. The Open scenario (cheapest) reduce even more total costs because the returning path to the depot is not being considered. However the average number of routes slightly increases. Although, the balancing scenarios is focused on creating solutions with an equally criteria on route loads, the cost tend to increase. The algorithm finds better solutions when balancing to the smallest number of routes which is near to the Best scenario. For instance, when we mix the balance and open criteria, the best total cost is found with an average balance of loads. Notice that all generated solutions have better values for the percentage of used capacities of vehicles than the Current scenario. However, longer CPU times are needed to find solutions combining open and balancing constraints.

Conclusions

In this paper, we have presented an algorithm for solving a Rich Vehicle Routing Problem (RVRP) with real restrictions. The proposed approach integrates a randomized savings heuristic approach with a local search. Several scenarios are evaluated using data obtained from a distribution company. These results revealed promising improvements for the day-to-day company routing planning. Through this experience it was possible to support a food distribution company to provide a technique for generating a set of alternative routing solutions with different features. This work presents some challenges to design and implement routing algorithms for automatizing real company distribution processes.

Acknowledgments—This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2010-21644-C03). It has been developed in the context of the IN3-ICSO program and the CYTED-HAROSA network (http://dpcs.uoc.edu).

References

Caceres-Cruz J, Riera D, Buil R, Juan A and Herrero R (2013). Multi-start Approach for Solving an Asymmetric Heterogeneous Vehicle Routing Problem in a Real Urban Context". In Proceedings of 2013 ICORES. February 16-18. Barcelona, Spain.

Clarke G and Wright J (1964). Scheduling of vehicles from central depot to number of delivery points. Operations Research, 12(4):568–581.

Dantzig GB and Ramser JH (1959). The Truck Dispatching Problem. Management Science 6:80-91.

Drexl M (2012). Rich vehicle routing in theory and practice. Logistics Research, 5:47-63.

Goel A and Gruhn V (2008). A general vehicle routing problem. European Journal of Operational Research, 191(3):650-660.

Hasle G and Kloster O (2007). Industrial vehicle routing. In Hasle G, Lie KA and Quak E (eds), Geometric Modelling, Numerical Simulation, and Optimization, pages 397-435. Springer Berlin H.

Hartl RF, Hasle G and Janssens GK (2006). Special issue on rich vehicle routing problems. Central European Journal of Operations Research, 14:103-104.

Juan AA, Faulín J, Ruiz R, Barrios B and Caballe S (2010). The SR-GCWS Hybrid Algorithm for Solving the Capacitated Vehicle Routing Problem. Applied Soft Computing 10: 215–224.

Juan AA, Faulin J, Jorba J, Riera D, Masip D and Barrios B (2011). On the Use of Monte Carlo Simulation, Cache and Splitting Techniques to Improve the Clarke and Wright Savings Heuristics. Journal of the Operational Research Society 62: 1085-1097.

Laporte G (2009). Fifty Years of Vehicle Routing. Transportation Science 43(4): 408-416.

Li F, Golden B and Wasil E (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers & Operations Research, 34(10): 2918-2930.

Oppen J, Lokketangen A and Desrosiers J (2010). Solving a rich vehicle routing and inventory problem using column generation. Computers & Operations Research, 37(7):1308-1317.

Pellegrini P, Favaretto D and Moretti E (2007). Multiple ant colony optimization for a rich vehicle routing problem: A case study. In Apolloni B, Howlett R and Jain L (eds), Knowledge-Based Intelligent Information and Engineering Systems, vol. 4693 of Lecture Notes in Computer Science, pages 627-634. Springer Berlin Heidelberg.

Prescott-Gagnon E, Desaulniers G. and Rousseau LM (2010). Heuristics for an oil delivery vehicle routing problem. Les Cahiers du GERAD G201063, 10.

Rodríguez A and Ruiz R (2012). A study on the effect of the asymmetry on real capacitated vehicle routing problems. Computers & Operations Research, 39(9): 2142-2151.

Toth P and Vigo D (eds) (2002). The Vehicle Routing Problem. SIAM.

Vidal T, Crainic TG, Gendreau M and Prins C (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & Operations Research, 40(1):475-489.