

An Exact Algorithm for the Open Vehicle Routing Problem with Hiring Cost

Efraín Ruiz y Ruiz, Cecilia Gutiérrez Mota, Pamela Chiñas Sánchez, José L. Dena Sánchez

División de Estudios de Posgrado
Tecnológico Nacional de México/ I.T. Saltillo
Saltillo, Coahuila, Mexico

Abstract—This paper proposes an exact algorithm for solving the open vehicle routing problem with hiring cost (OVRPHC). Given a central depot, vehicles depart to deliver goods demanded by a set of clients. There is a cost associated with every vehicle used for deliveries. Every client is visited by a vehicle part of a homogeneous fleet, which delivers the required amount of a given product or products. Vehicles depart from the depot and are not required to return to it, finishing their routes after delivering the products to the last client. There is an associated cost for the distance traveled by vehicles. The problem's objective function is to minimize the total traveling and hiring costs while considering vehicle capacity constraints. To solve the problem, a node-based formulation and a branch-and-cut algorithm based on a three-index formulation are proposed. A set of 14 benchmark instances was used to test the proposed algorithm. The results obtained in the computational experiments, show the effectiveness of the approach.

Keywords:- *Vehicle routing; OVRP; branch-and-cut; combinatorial optimization*

I. INTRODUCTION

As competition between companies continues growing due to the incorporation of new competitors, the reduction of prices, the reduction of market share and others factors, companies continue working on reducing cost while improving the efficiency of the available resources. Logistic costs are said to represent about 20 % of the product cost. One of the most important activities in logistics is the delivery of the goods produced to the clients which are usually made by a fleet of vehicles usually owned by the company. To reduce costs and improve efficiency some companies have decided to outsource the delivery activity using a transport company or by renting vehicles. Is in this context where the open vehicle routing problem takes its importance since vehicles are not required to return to the depot after servicing the last client.

At present there are an important number of companies which offer delivery services that rely on outsourcing, allowing them to hire vehicles to make deliveries, saving the purchase of vehicles and their maintenance. These companies face the challenge of solving routing designs problems to optimize the use of the vehicles as well as save money. The basic problem that arises to optimize the use of vehicles to deliver goods is the vehicle routing problem (VRP). There are many extensions and generalizations of the vehicle routing problem, which consider additional constraints that, affect the route design process. Among the most important variants of the VRP, we can mention the ones with time windows, pickup and delivery, back-hauls, multi-depot, open routes, etc.

Normally in VRP problems vehicles are required to return to the depot after they serve the last client, but that is not the case in the open vehicle routing problem (OVRP), in which routes end after servicing the last client. The difference between VRP and OVRP is shown in Figures 1 and 2.

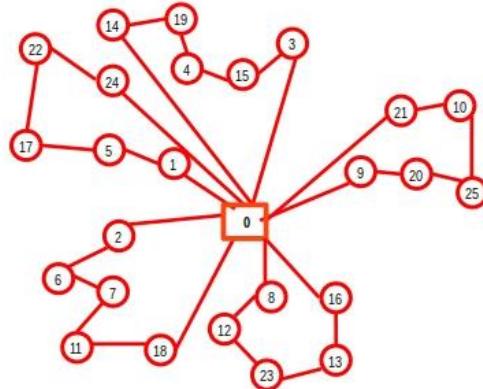


Figure 1. VRP example

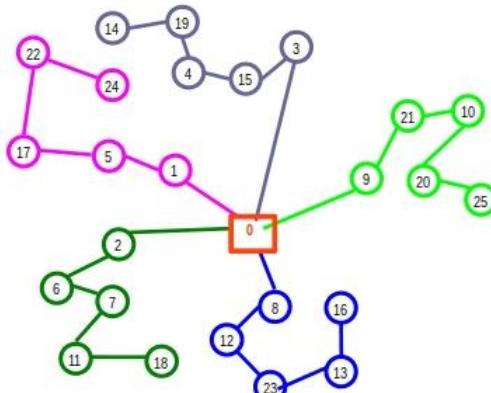


Figure 2. OVRP example

The OVRP has been studied in different applications like the one in [14] about an air express courier problem, the train services planning at the Channel tunnel [4], or the one in [16] among other applications. The OVRP considers capacity constraints over the vehicles and maximum length constraints over the routes. Since the OVRP is of type NP-hard (by the

reduction of the Hamiltonian path problem), many of the previous research works have concentrated on using heuristic methods to solve instances of the problem. Among such works we can mention the one by [3], who proposed a variable neighborhood search, the Hybrid Evolutionary Strategy by [13], the ant colony optimization-based metaheuristic by [7], the work by Repoussis et al [10] which proposes a hybrid evolution strategy, the Bumble Bees Mating Optimization by [8], the work by [17] and the ILP improvement by [12]. Although the difficulty of the problem, also exact methods have been developed like the branch-and-cut algorithm by [5]. In this paper, the Open Vehicle Routing Problem with Hiring Costs (OVRPHC) is presented and studied. This new combinatorial optimization problem is an extension of the open vehicle routing problem (OVRP). In the OVRP many authors use a hierarchical objective function that first minimizes the number of vehicles and then minimizes the total traveled distance. Other authors consider only the minimization of the total traveled distance.

The OVRPHC considers a hiring cost for each vehicle, which is included in the objective function. In this sense is no longer required to use a hierarchical objective function. A formulation and a cutting plane algorithm to solve the problem are presented. Two sets of benchmark instances from the literature are used to perform the computational experiments. The size of the instances varies from 32 to 51 vertices. Every test instance was solved using the cutting plane algorithm. The cutting plane algorithm obtains lower gaps between the lower bound and the best integer solution found than the formulation without cuts. Finally, is worth mentioning that the cutting plane algorithm finds the optimal solution for 26 of the 27 test instances.

II. PROBLEM DESCRIPTION AND FORMULATION

As with many routing problems the OVRPHC can be defined over a graph. As in the VRP, clients are represented as vertices as well as the depot. Edges or arcs represent the distance between the depot and any customer and between any customer to the rest of the customers. Then the OVRPHC is defined as follows:

Let $G = (V, A)$ be a graph where V is a set of vertices and A a set of arcs. Set V is defined as $V = \{v_0, v_1, \dots, v_n\}$, where v_0 is the depot and $V^+ = \{v_1, \dots, v_n\} \subset V$ is a set of n clients. Each client v_i , $i = 1, \dots, n$, has an associated demand w_i that must be delivered. Each arc $a = (i, j) \in A$ has an associated cost $c_{ij} > 0$. Feasible solutions to the OVRPHC are open routes with origin at v_0 that satisfy capacity and route length constraints. Finally, there is a set K of available vehicles, which have an associated hiring cost hc_k .

The OVRPHC is to find a set of open routes with origin at v_0 that minimize the hiring and routing costs and satisfy the capacity and route length constraints. Since the OVRPHC is an extension of the OVRP, it is also of type NP-hard.

The OVRPHC can be formulated in many ways. Following, we present a node-based formulation for the problem.

A. Node-based formulation

To solve instances of the OVRPHC, we propose a Mixed Integer Linear Programming (MILP) formulation. Therefore, the following variables are defined:

$$y_k = \begin{cases} 1, & \text{if vehicle } k \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{if arc } (i, j) \text{ is traversed by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

$$T_i = \text{load in vehicle after visiting client } i \in V^+$$

Using these variables, the OVRPHC is formulated as follows:

$$\text{Minimize } \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} + \sum_{k \in K} hc_k y_k \quad (1)$$

Subject to:

$$\sum_{k \in K} \sum_{i \in V} c_{ij} x_{ijk} = 1 \quad \forall j \in V^+ \quad (2)$$

$$\sum_{k \in K} \sum_{\substack{i \in V \\ i \neq j}} c_{ji} x_{jik} \leq 1 \quad \forall j \in V^+ \quad (3)$$

$$\sum_{j \in V} x_{0jk} \leq y_k \quad \forall k \in K \quad (4)$$

$$T_i - T_j \sum_{k \in K} Q x_{ijk} \leq Q - d_j \quad (i, j) \in A \quad (5)$$

$$x_{ijk} \in \{0, 1\} \quad i \in V, j \in V, k \in K \quad (6)$$

$$y_k \in \{0, 1\} \quad k \in K \quad (7)$$

The Objective Function (1) minimizes the total cost including routing and hiring costs. Constraints (2) impose that every client is visited, while constraints (3) state that vehicles can only depart one time from a client. Constraints (4) impose that vehicles can only depart from the depot if they are used. Finally, constraints (5) are the well-known Miller-Tucker-Zemlin (MTZ) connectivity constraints and (6) are variable domain constraints.

III. BRANCH-AND-CUT ALGORITHM

Since the OVRPHC is of type NP-hard, it is difficult to find optimal solutions even for small instances of the problem. Therefore, we developed a branch-and-cut algorithm based on the separation of valid inequalities valid for routing problems, like the multistar, bin-packing and root-cutset inequalities.

A. Multistar inequalities

These inequalities were first proposed by [1]. These inequalities enhance the connectivity of vertices, considering the required capacity to serve a subset of vertices $S \in V^+$. Considering the variables used to formulate the OVRPHC, they are expressed as follows:

$$\begin{aligned} & \sum_{k \in K} \sum_{j \in S} Q x_{0jk} + \sum_{k \in K} \sum_{\substack{i \in S \\ j \in S}} (Q - d_i) x_{ijk} \\ & - \sum_{k \in K} \sum_{\substack{i \notin S \\ j \in S}} d_i x_{jik} \geq \sum_{j \in S} d_j \end{aligned} \quad (8)$$

B. Bin-packing inequalities

The Bin-packing inequalities can be expressed as connectivity or as sub-tour elimination constraints (SEC). The SEC version takes the following form:

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - VR(S) \quad (9)$$

where

$$VR(S) = \left\lceil \frac{\sum_{j \in S} d_j}{Q} \right\rceil$$

is the minimum required number of vehicles to supply the demand of the clients in set S .

C. Root-cutset inequalities

This family of inequalities were also proposed by [1]. These inequalities help enforce the connection with the depot and have proved to be helpful in network design and routing problems. They are expressed as follows:

$$\frac{VR(S)+1}{VR(S)} \sum_{k \in K} \sum_{i \in B} \sum_{j \in S} x_{ijk} + \sum_{k \in K} \sum_{i \in B} \sum_{j \in S} x_{ijk} \geq VR(S)+1 \quad (10)$$

where

$$B = \left\{ i \notin S : \left\lceil \frac{d(S) + d_i}{Q} \right\rceil > \left\lceil \frac{d(S)}{Q} \right\rceil \right\}, \quad \bar{B} = V^+ \setminus \{S \cup B\}$$

and

$$d(S) = \sum_{i \in S} d_i$$

D. Branch-and-cut algorithm

The branch-and-cut algorithm solves the linear relaxation of the model presented in the previous section. The branch-and-cut algorithm removes constraints (7) and (8) and replaces them with:

$$x_{ijk} \in [0, 1] \quad i \in V, j \in V, k \in K \quad (11)$$

$$y_k \in [0, 1] \quad k \in K \quad (12)$$

and the relaxed problem LR is obtained. The branch-and-cut algorithm solves LR to obtain the fractional solution X^* . Once a solution is obtained, the separation algorithm proposed in [1] is used to find multistar, bin-packing and root-cutset inequalities violated by the fractional solution X^* . The violated inequalities are then included in the relaxed problem LR and solved. This procedure is repeated until no violated inequalities are found. The final LR model is then used with a branch and bound algorithm to find the optimal solution to the problem.

IV. COMPUTATIONAL EXPERIMENTS

To assess the performance of the proposed branch-and-cut algorithm computational experiments were performed. To perform the computational experiments a set of 27 benchmark

instances from the literature was used. The characteristics of the instances are described in Table I, where n represents the number of clients and minK the capacity of the vehicles. For the computational experiments, three different values for the hiring cost (HC) were used. In total 81 computational experiments were run.

TABLE I. TEST INSTANCES

Instance	n	minK	Instance	n	minK
A-n32-k5	32	5	A-n48-k7	48	7
A-n33-k5	33	5	A-n53-k7	53	7
A-n33-k6	33	6	A-n54-k7	54	7
A-n34-k5	34	5	A-n55-k9	55	9
A-n36-k5	36	5	A-n60-k9	60	9
A-n37-k5	37	5	A-n61-k9	61	9
A-n37-k6	37	6	A-n62-k8	62	8
A-n38-k5	38	5	A-n63-k9	63	9
A-n39-k5	39	5	A-n63-k10	63	10
A-n39-k6	39	6	A-n64-k9	64	9
A-n44-k7	44	7	A-n65-k9	65	9
A-n45-k6	45	6	A-n69-k9	69	9
A-n45-k7	45	7	A-n80-k10	80	10
A-n46-k7	46	7	-	-	-

The formulation and the branch-and-cut algorithm were programmed in C++, using GUROBI 9.0 as the solver. The experiments were performed on a PC using Linux with an intel i7 6600 processor running at 3.4 GHz., and 16 GB of memory. For every test instance, a solution time limit of 72000 seconds was used.

The results of the computational experiments of the branch-and-cut algorithm are presented in Tables II, III, and IV. In both tables, column **gap** shows the percentile difference between the upper bound and lower bound obtained by the respective solution approach, and column **k** presents the number of vehicles used in the optimal/best solution found by the respective solution method. Finally, column **Time** shows the time used by the respective solution method.

TABLE II. BRANCH-AND-CUT ALGORITHM RESULTS (HC=0)

Instance	gap	k	Time	Instance	gap	k	Time
A-n32-k5	0.00	5	1.5	A-n48-k7	0.00	7	39.4
A-n33-k5	0.00	5	1.6	A-n53-k7	0.00	8	15.1
A-n33-k6	0.00	6	4.9	A-n54-k7	0.00	7	476.6
A-n34-k5	0.00	6	7.5	A-n55-k9	0.00	9	277.4
A-n36-k5	0.00	5	5.3	A-n60-k9	0.00	10	81.9
A-n37-k5	0.00	5	2.8	A-n61-k9	0.00	9	73.7
A-n37-k6	0.00	6	10.4	A-n62-k8	0.00	8	2756.5
A-n38-k5	0.00	6	2.1	A-n63-k9	0.00	10	3093.4
A-n39-k5	0.00	6	5.5	A-n63-k10	0.00	10	115
A-n39-k6	0.00	6	7.6	A-n64-k9	0.00	9	2610.1

A-n44-k7	0.00	7	60.9	A-n65-k9	0.00	9	598.2
A-n45-k6	0.00	7	6.6	A-n69-k9	0.00	9	557.6
A-n45-k7	0.00	7	62.9	A-n80-k10	4.75	11	7200.0
A-n46-k7	0.00	7	14.3		-	-	-

a.

TABLE III. BRANCH-AND-CUT ALGORITHM RESULTS (HC=50)

Instance	gap	k	Time	Instance	gap	k	Time
A-n32-k5	0.00	5	2.7	A-n48-k7	0.00	7	29.7
A-n33-k5	0.00	5	4.7	A-n53-k7	0.00	7	85.4
A-n33-k6	0.00	6	9.7	A-n54-k7	0.00	7	612.6
A-n34-k5	0.00	5	40.7	A-n55-k9	0.00	9	222.2
A-n36-k5	0.00	5	9.3	A-n60-k9	0.00	9	590.5
A-n37-k5	0.00	5	16.0	A-n61-k9	0.00	9	232.7
A-n37-k6	0.00	6	68.0	A-n62-k8	0.00	8	1390.0
A-n38-k5	0.00	5	169.2	A-n63-k9	0.00	9	1234.6
A-n39-k5	0.00	5	41.3	A-n63-k10	0.00	10	208.7
A-n39-k6	0.00	6	12.0	A-n64-k9	0.00	9	1730.1
A-n44-k7	0.00	7	78.7	A-n65-k9	0.00	9	1690.2
A-n45-k6	0.00	6	25.2	A-n69-k9	0.00	9	446.3
A-n45-k7	0.00	7	231.5	A-n80-k10	5.31	10	7200
A-n46-k7	0.00	7	19.8		-	-	-

TABLE IV. BRANCH-AND-CUT ALGORITHM RESULTS (HC=100)

Instance	gap	k	Time	Instance	gap	k	Time
A-n32-k5	0.00	5	3.1	A-n48-k7	0.00	7	55.1
A-n33-k5	0.00	5	9.1	A-n53-k7	0.00	7	385.1
A-n33-k6	0.00	6	5.6	A-n54-k7	0.00	7	452.4
A-n34-k5	0.00	5	45.3	A-n55-k9	0.00	9	434.1
A-n36-k5	0.00	5	13.1	A-n60-k9	0.00	9	341.1
A-n37-k5	0.00	5	15.0	A-n61-k9	0.00	9	298.8
A-n37-k6	0.00	6	9.2	A-n62-k8	0.00	8	2628.1
A-n38-k5	0.00	5	202.6	A-n63-k9	0.42	9	7200
A-n39-k5	0.00	5	61.7	A-n63-k10	0.00	10	154.8
A-n39-k6	0.00	6	16.4	A-n64-k9	0.42	9	1264.5
A-n44-k7	0.00	7	139.1	A-n65-k9	0.00	9	874.7
A-n45-k6	0.00	6	25.2	A-n69-k9	0.00	9	1134.8
A-n45-k7	0.00	7	34.1	A-n80-k10	2.03	10	7200
A-n46-k7	0.00	7	97.5		-	-	-

The results in Table I (with HC=0) show that the branch-and-cut algorithm is able to find the optimal solutions for 26 of the 27 benchmark instances when there is no hiring cost. The solution approach finds the optimal solution for instances of up to 69 vertices (clients). For the largest instance with 80 vertices (clients), the algorithm is unable to find the optimal solution and obtains a gap of 4.75% reaching the time limit.

Regarding the number of vehicles to be used, for eight of the test instances, the optimal solution found uses one more vehicle than the minimum required.

For a hiring cost of 50 (HC=50), the results in Table II show that the branch-and-cut algorithm again optimally solves 26 of 27 benchmark instances. Also, the largest instance with 80 vertices (clients) is not solved to optimality and obtains a gap of 5.31% after reaching the time limit. In this case, the number of vehicles used in the optimal solutions found by the algorithm is equal to the minimum required.

Finally, in Table III with (HC=100) the branch-and-cut algorithm finds the optimal solutions for 25 of the 27 benchmark instances. In this case, the algorithm is unable to find the optimal solution for instances “A-n64-k9” and “A-n80-k10”. Reaching in both cases the time limit. The gap for both instances is 0.42% and 2.03 respectively. As in the previous case (HC=50) the solutions found by the algorithm, use the minimum of required vehicles.

V. CONCLUDING REMARKS

The Open Vehicle Routing Problem with Hiring Cost, a new variant of the well-known Open Vehicle Routing Problem, was introduced and formulated. A branch-and-cut algorithm, based on the separation of valid inequalities was used to perform computational experiments. The results show that the algorithm is capable of finding optimal solutions for instances with up to 69 vertices (clients). For larger instances of 80 vertices, the solution approach is unable to find the optimal solution. Therefore, the approach is efficient for small and medium-sized instances and struggles with larger instances.

REFERENCES

- [1] Araque, J. R., Hall, L. A., & Magnanti, T. L. (1990). Capacitated trees, capacitated routing, and associated polyhedra.
- [2] Nicos Christofides. Combinatorial optimization, volume 1. 1979.
- [3] Krzysztof Fleszar, Ibrahim H Osman, and Khalil SHindi. A variable neighbourhood search algorithm for the open vehicle routing problem. European Journal of Operational Research, 195(3): 803–809, 2009.
- [4] Zhuo Fu, Richard Eglese, and Leon YO Li. A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society, 56(3):267–274, 2005.
- [5] Adam N Letchford, Jens Lysgaard, and Richard W Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. Journal of the Operational Research Society, 58(12):1642–1651, 2007.
- [6] Feiyue Li, Bruce Golden, and Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers & Operations Research, 34(10):2918–2930, 2007.
- [7] XY Li, Peng Tian, and Stephen CH Leung. An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems. Journal of the Operational Research Society, 60(7):1012–1025, 2009.
- [8] Yannis Marinakis and Magdalene Marinaki. A bumble bees mating optimization algorithm for the open vehicle routing problem. Swarm and Evolutionary Computation, 15:80–94, 2014.
- [9] R.C. Prim. Shortest connection matrix network and some generalizations. Bell System Technical Journal, 36:1389–1401, 1957.
- [10] Panagiotis P Repoussis, Christos D Tarantilis, Olli Br'aysy, and George Ioannou. A hybrid evolution strategy for the open vehicle routing problem. Computers & Operations Research, 37(3):443–455, 2010.
- [11] Efrain Ruiz, Maria Albareda-Sambola, Elena Fernández, and Mauricio GC Resende. A biased random-key genetic algorithm for the capacitated minimum spanning tree problem. Computers & Operations Research, 57:95–108, 2015.
- [12] 12. Majid Salar, Paolo Toth, and Andrea Tramontani. An ilp improvement procedure for the open vehicle routing problem. Computers & Operations Research, 37(12):2106–2120, 2010.