ON THE MULTI-DEPOT VEHICLE ROUTING PROBLEM

J. Orestes CERDEIRA
Inst. Superior de Agronomia
Tapada da Ajuda, 1399 Lisboa – Codex, Portugal

1 – Introduction

The multi-depot vehicle routing problem deals with situations in which vehicles based at several depots are required to visit customers so as to fulfill known customers' requirements. Each vehicle leaves the depot where it is based, eventually returning to that same depot after visiting certain customers. It is asked to design a route for each vehicle so that each customer will be entirely supplied by a single vehicle, and the total cost of the distribution will be minimum.

We consider the multi-depot vehicle routing problem with vehicles of unbounded capacity, in short MDVRP. Using covers of the nodes by trees and matchings, we present a way of determining lower bounds on the cost of the optimal solutions. We also describe a class of algorithms for this problem, which includes polynomial time algorithms, and we show that if the triangle inequality holds, the costs of the resulting solutions are not greater than twice the optimal costs. Computational experience is reported.

2 – Formulation

Each instance of the MDVRP is described by an undirected weighted graph $G = (V, E, c)$, a subset K ($|K| = k$) of the node set V ($|V| = n$), and a function $nveic : K \to \mathbf{Z}^+$. Each node of G represents either a customer or a depot. K denotes the set of nodes representing depots. E is the edge set, and associated with each edge e we have the cost c_e of traveling directly between the two locations (customers or depots) represented by the extreme nodes of e. We assume there are no edges having both extreme nodes in K. If v is a node of K, $nveic_v$ is the number of vehicles based at the depot represented by v.

The MDVRP consists in determining $\sum_{v \in K} nveic_v$ cycles in graph G such that

i) each node v is included in exactly one cycle, if $v \in V - K$, and in exactly $nveic_v$ cycles, if $v \in K$,

ii) no two nodes from K appear in the same cycle, and

iii) the sum of the costs of all edges in these cycles is minimum.

P. Gritzmann et al. (eds.), *Operations Research '91*

Let T_k be the set of the $0-1$ incidence vectors of the covers of the nodes of G by k node-disjoint trees (i.e. the sets of $n-k$ edges which leave no node isolated, and free of cycles), such that no two nodes of K are included in the same tree. Let also M_k be the set of the $0-1$ incidence vectors of the sets of $\sum_{v\in K} nveic_v$ edges which cover all the nodes in K. Denoting by E_v the set of edges of G incident in node v, the MDVRP can be formulated as follows:

$$\min_{x\in T_k,\, y\in M_k} \sum_{e\in E} c_e\,(x_e+y_e) \tag{1}$$

subject to:

$$\sum_{e\in E_v} x_e = nveic_v \qquad \forall\, v\in K \tag{2}$$

$$\sum_{e\in E_v} y_e = nveic_v \qquad \forall\, v\in K \tag{3}$$

$$\sum_{e\in E_v} y_e \le 1 \qquad \forall\, v\in V-K \tag{4}$$

$$\sum_{e\in E_v} (x_e+y_e) = 2 \qquad \forall\, v\in V-K \tag{5}$$

$$\text{the set of edges defined by } x+y \text{ includes exactly } \sum_{v\in K} nveic_v \text{ cycles.} \tag{6}$$

3 – Lower Bounds

For each node $v\in V-K$, let $\lambda_v\in\mathbf{R}$ be the Lagrange multiplier associated with the corresponding constraint (5), and let $\lambda_v=0$, $\forall\, v\in K$. If $e=[v_e,u_e]$ is an edge of G, let $\bar{c}_e=c_e-\lambda_{v_e}-\lambda_{u_e}$, and $\theta:\mathbf{R}^n\to\mathbf{R}$ be the function defined by

$$\theta(\lambda)=\min_{x\in T_k,\, y\in M_k}\sum_{e\in E}\bar{c}_e(x_e+y_e)+2\sum_{v\in V-K}\lambda_v \tag{7}$$

subject to: (2)–(4).

Given that (7), (2)–(4) is a relaxation of the MDVRP, for each feasible vector $\lambda\in\mathbf{R}^n$, $\theta(\lambda)$ is a lower bound on the value of (1) subject to (2)–(6). Moreover, determining $\theta(\lambda)$ can be achieved by solving the two following problems:

$$\min_{x\in T_k}\sum_{e\in E}\bar{c}_e\,x_e \tag{8}$$

subject to: (2), and

$$\min_{y\in M_k}\sum_{e\in E}\bar{c}_e\,y_e \tag{9}$$

subject to: (3), (4).

To solve problem (8), (2) there is an $O(n^2+(\sum_{v\in K} nveic_v)^2\, n)$ time algorithm [**1**,**2**]. Problem (9), (3), (4) can be solved as a matching problem in a bipartite graph in $O((\sum_{v\in K} nveic_v)^2\, n)$ time.

Therefore we have an $O(n^2+(\sum_{v\in K} nveic_v)^2\, n)$ time algorithm to obtain the lower bound given by (7), (2)–(4) on the cost of the optimal solution of the MDVRP.

4 – Upper Bounds

Let x^* be the solution that

$$\min_{x \in T_k} \sum_{e \in E} c_e \, x_e \quad \text{subject to: (2)} ,$$

which, as we said before, can be obtained in $O(n^2 + (\sum_{v \in K} nveic_v)^2 \, n)$ time. From x^* we assign customers to vehicles. Let $E(x^*)$ be the set of edges defined by x^*, and $\widehat{E}(x^*)$ the set of all edges of $E(x^*)$ except those incident in the nodes of K. The customers represented by nodes v, u will be supplied by the same vehicle based at the depot represented by r iff v, u, r are all included in the same connected component of the graph $(V, E(x^*))$, and v, u appear in the same component of $(V, \widehat{E}(x^*))$. Let S_{v_i} be the set of nodes representing customers which will be supplied by the i-th vehicle based at depot represented by v. If we use an algorithm to solve the traveling salesman problem in the subgraph of G induced by $S_{v_i} \cup \{v\}$, for $i = 1, ..., nveic_v$ and for all $v \in K$, we obtain a feasible solution of the MDVRP. Suppose now the cost vector c satisfies the triangle inequality, i.e., $c_{[v,u]} + c_{[u,r]} \geq c_{[v,r]}$, $\forall \, v, u, r \in V$. There are several polynomial time algorithms for the traveling salesman problem which determine tours whose costs are not greater than twice the costs of the minimum spanning trees (see for example [**6**, **5**]). With any choice of such an algorithm, the solutions of the MDVRP will be determined in polynomial time. Moreover, the costs of those solutions will be not greater than twice the optimal costs.

5 – Computational Results

In the computational tests we carried out, we use graphs with 50, 60, 70, 80, 90 and 100 nodes. In each of the six cases, ten different cost vectors satisfying the triangle inequality were considered. Each of these matrices results from applying Floyd's algorithm [**3**], for finding the shortest paths between all pairs of nodes, to a symmetric matrix with integer entries randomly selected in the range $[0, 100]$. For each graph we let the number of depots k be equal to $\lfloor \frac{n}{20} \rfloor$ and $\lfloor \frac{n}{15} \rfloor$. In each case we consider the three problems emerging by letting the number of vehicles, NV, be equal to $\lfloor \frac{n}{10} \rfloor$, $\lfloor \frac{n}{5} \rfloor$ and $\lfloor \frac{n}{3} \rfloor$. The assignment of vehicles to depots (i.e. the vector $(nveic_v)_{v \in K}$) was determined by generating a random partition of the number of vehicles with as many components as the number of depots k. The lower bounds LB were obtained according to the method described in section 3, at the end of n iterations of the subgradient method [**4**]. The upper bounds UB are the costs of the solutions determined by the algorithm of section 4, where the algorithm used to solve the traveling salesman problem works as follows. First a tour is obtained by the so-called minimum spanning tree algorithm [**6**, **5**]. Next, the 3-optimal method appearing in [**7**, pg. 377–379] is used to improve that tour.

The main results are shown in figure 1. For the ten problems having the same node set, the same number of depots and the same number of vehicles, the average of the ratios $\frac{UB}{LB}$ is displayed in the 4-th and 9-th columns of the table. In the 5-th and 10-th columns we report the greatest value of these ratios among the ten problems.

n	k	NV	aver$\left(\frac{UB}{LB}\right)$	worst$\left(\frac{UB}{LB}\right)$	n	k	NV	aver$\left(\frac{UB}{LB}\right)$	worst$\left(\frac{UB}{LB}\right)$
50	2	5	1.102	1.170	80	4	8	1.123	1.188
		10	1.094	1.150			16	1.127	1.191
		16	1.078	1.127			26	1.078	1.132
	3	5	1.087	1.144		5	8	1.123	1.175
		10	1.099	1.125			16	1.107	1.160
		16	1.074	1.123			26	1.084	1.143
60	3	6	1.089	1.146	90	4	9	1.157	1.250
		12	1.087	1.146			18	1.147	1.240
		20	1.073	1.142			30	1.098	1.137
	4	6	1.094	1.164		6	9	1.164	1.252
		12	1.105	1.220			18	1.139	1.299
		20	1.079	1.141			30	1.119	1.207
70	3	7	1.098	1.174	100	5	10	1.151	1.284
		14	1.089	1.138			20	1.135	1.232
		23	1.057	1.087			33	1.081	1.147
	4	7	1.092	1.112		6	10	1.140	1.259
		14	1.091	1.135			20	1.120	1.226
		23	1.052	1.080			33	1.105	1.160

Fig.1

References

[1] S.M. BAAS and J.O. CERDEIRA, "Covering Nodes by k Node-Disjoint Trees with Additional Constraints", Memorandum nº 693, Faculteit der Toegepaste Wiskunde, Universiteit Twente, 1988.

[2] J.O. CERDEIRA, "Cobertura dos Vértices de um Grafo por k Árvores e Ciclos Disjuntos nos Vértices", Tese de Doutoramento, Faculdade de Ciências da Universidade de Lisboa, 1990.

[3] R.W. FLOYD, "Algorithm 97: Shortest Path", *Communications of the ACM*, 5, 1962, 345.

[4] M. HELD, P. WOLFE and H.P. CROWDER, "Validation of Subgradient Optimization, *Mathematical Programming*, 6, 1974, 62–88.

[5] D.S. JOHNSON and C.H. PAPADIMITRIOU, "Performance Guarantees for Heuristics", in *The Traveling Salesman Problem: A Guide Tour of Combinatorial Optimization*, Eds.: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, John Wiley & Sons, New York, 1985, 145–180.

[6] C.H. PAPADIMITRIOU and K. STIEGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, New York, 1982.

[7] M.M. SYSLO, N. DEO and J.S. KOWALIK, *Discrete Optimization Algorithms with Pascal Programs*, Prentice-Hall, Englewood Cliffs, New York, 1983.