QÜESTIIÓ, Vol 14, nº 1,2,3 pp. 107-122, 1990

THE CAPACITATED ARC ROUTING PROBLEM. A HEURISTIC ALGORITHM

E. BENAVENT, V. CAMPOS
A. CORBERÁN, E. MOTA
Universidad de Valencia

In this paper we consider the Capacitated Arc Routing Problem, in which a fleet of K vehicles, all of them based on a specified vertex (the depot) and with a known capacity Q, must service a subset of the edges of a graph, with minimum total cost and such that the load assigned to each vehicle does not exceed its capacity.

A heuristic alogithm for this problem is proposed consisting of: the selection of K centers, the construction of K connected graphs with associated loads not exceeding the vehicle capacities, the resolution of a Generalized Assignment Problem, if necessary, to get a complete assignment of edges to vehicles and, finally, the construction of the routes, by solving, heuristically, a Rural Postman Problem for each vehicle.

Computational results on graphs up to 50 vertices and 97 edges are included. On average, the feasible solution is within 6.4% of the best known lower bound.

Keywords: Distribution, Heuristics, Routing.

1. INTRODUCTION

Routing Problems have been widely studied in the last years, mainly because of the great number of practical applications and the big increase of the costs associated with operating the vehicles. Basically, these problems can be divided into Node Routing Problems, if the demand occurs in the nodes or vertices of a graph, and Arc Routing Problems, in which the pickup or delivery activities

–E. Benavent, V. Campos, A. Corberán, E. Mota - Departamento de Estadística e Investigación Operativa. Facultad de Matemáticas. Universidad de Valencia.
–Article rebut l'abril de 1990.

occur along the arcs or edges of a graph. See Bodin and Golden (1981) and Lenstra and Rinnooy-Kan (1981) for a more detailed classification of these problems and their complexity.

However, the research work has been mainly focused on Node Routing Problems (see the excellent survey of Bodin *et al.*(1983)), while Arc Routing Problems have received comparatively little attention (see Benavent *et al.*(1983)), in spite of their applications in a great number of real problems, such as problems of refuse collection, street sweeping operations, delivery of milk or post, inspection of distributed systems (electric power, telephone or railway lines) and school bus routing.

In this paper we consider the Capactitated Arc Routing Problem (CARP), in which a fleet of K vehicles, all of them based on a specified vertex (the depot) and with a known capacity Q, must service the edges with positive load of a graph, with minimum total cost and such that the load assigned to each vehicle does not exceed its capacity.

Although Assad *et al.*(1987) have shown that certain classes of the CARP with special graph, demand or costs structures can be solved with a polynomial time algorithm, the general case is a NP–Hard problem. Even the problem of finding a solution with a cost less than 1.5 times the optimal solution cost is NP–Hard, as shown by Golden and Wong (1981).

Due to the complexity of the problem, several heuristic algorithms and lower bounding techniques have been developed for the CARP. Among the heuristics, it is possible to point out those of Cristofides (1973), Beltrami and Bodin (1974) for the routing of street sweepers, Male and Liebman (1978) and Bodin *et al.*(1989) for refuse collection problems, Stern and Dror (1979) for the routing of electric meters readers, Golden and Wong (1981), Golden *et al.*(1983) and Chapleau *et al.*(1984) for school bus routing.

Lower bounding techniques for the CARP are presented in Golden and Wong (1981), Assad *et al.*(1987), Benavent *et al.*(1987), Pearn (1988) and Zaw Win (1988).

In this paper we present a new heuristic algorithm for the CARP which is an extension of the one proposed by Benavent *et al.*(1985) and consists of: the selection of K centers, the construction of K connected graphs with associated loads not exceeding the vehicle capacities, the resolution of a Generalized Assignment Problem (GAP), if necessary, to get a complete assignment of edges to vehicles and, finally, the construction of the routes, by solving heuristically, a Rural Postman Problem for each vehicle.

2. PROBLEM DEFINITION AND NOTATION

Given a connected and non directed graph $G = (V, E)$, consider that to every edge $e = (i, j)$ of G there is associated:

i) a load $q_e \geq 0$,
ii) a cost $c_e \geq 0$ of just traversing edge e and
iii) a cost $c'_e \geq 0$ of servicing edge e (thus, traversing it), if $q_e > 0$.

Given vertex 1, representing the depot, and a number K of vehicles $(K \geq 2)$, each one with capacity Q $(Q \geq \max\{q_e, e \in E\})$, the problem considered in this paper consists of finding a set of K routes, each one containing the depot, such that, jointly, they service all the edges with positive load of G, at a minimum total cost and such that the sum of the loads corresponding to the edges serviced by each route does not exceed the vehicle capacity Q. Note that a route is determined by a set of traversed edges, with indication of the serviced ones.

Let s_{ij} denote the cost of the shortest path from vertex i to vertex j, computed using the costs c_e; costs s_{ij} will be also referred to as distances. Let $E^R = \{e \in E : q_e > 0\}$ be the set of required edges, V^R the set of vertices incident with at least one required edge and $G^R = (V^R, E^R)$.

For a given subset of edges $E', q(E')$ will denote the sum of the loads associated to the edges in E'. A graph is even iff the degree of all its vertices is even.

3. DESCRIPTION OF THE ALGORITHM

The heuristic algorithm proposed for the Capacitated Arc Routing Problem consists of several clearly differentiated stages. In the first stage, the set of edges to be serviced by each vehicle is defined, taking into account the capacity of the vehicles. In the second stage, the routes are constructed thus obtaining a feasible solution to the CARP. Finally, the solution is revised using an interchange procedure.

The first stage influences most decisively the quality of the feasible solution. Let $E_k \subseteq E^R$ be set of required edges assigned to vehicle k, and let G_k be the graph induced by E_k. In the first stage, the objective is to construct subgraphs G_k, $k = 1, \cdots, K$ such that:

a) they ought to have the minimum possible number of odd degree vertices,
b) they should be composed by a few number of connected components and, in any case, the distance among them should be small, and
c) they should be as close the depot as possible.

In order to meet, at least heuristically, these requirements, the procedure followed in this first stage is:

i) K vertices (called centers), uniformly distributed on the graph, are selected from V^R.
ii) Complete cycles in the graph are, successively, assigned to each center, in such a way that the set of cycles assigned to a center is connected and its total load does not exceed the capacity of the vehicles. Previously, artificial edges have been added to the graph G^R in order to obtain a connected and even graph.
iii) In case that in the previous step some required edges had been left unassigned, a GAP is solved by using assignment costs of edges to vehicles that take into account the partial assignment obtained in ii) as well as the objectives a) and c).

In what follows, a more detailed description of every stage of the proposed heuristic algorithm is given.

3.1 Selection of centers

For each vehicle k, a vertex $i_k \in V^R$ (center) other than the depot is selected, trying to obtain a uniform distribution of centers on graph G. Several methods, with different objectives were tried and computational results showed that the quality of the final solution did not depend too much on the selection method.

The finally adopted procedure consists of constructing a set of K centers, selecting for each $k = 1, \cdots, K$ a vertex $i_k \in V^R$, such that the product of the distances from this vertex to the centers $i_1, \cdots, i_{k-1}$ and the depot is maximum.

Afterwards, an interchange type procedure is applied, trying to generate new sets of centers in such a way that each interchange increases the product of distances among all centers and distances between the centers and the depot. The procedure ends when no further increase is obtained.

3.2 Assignment of Cycles

As a previous step, a set of edges with zero load (called artificial edges) is added to G^R to obtain a new even and connected graph G'. This is accomplished by solving, first, a Shortest Spanning Tree Problem (SST) to convert G^R into a connected graph, if it were not already connected and, second, a Minimum Cost Matching Problem on the odd degree vertices of the graph resulting from G^R by adding the edges in the above SST solution. Costs, in the resolution of both problems, are equal to those of the shortest paths, s_{ij} , between every pair of vertices.

The following procedure determines, for each vehicle k, a set of edges E'_k from G'. Let G'_k be the subgraph induced by the set of edges E'_k . Initially, G'_k contains only center i_k and no edges. Let $Q_k = Q - q(E_k)$ denote the residual (disposable) capacity of vehicle k.

Cycle assignment procedure:

Step 0: (Init)
Mark all vehicles as not complete.

Step 1: If all vehicles are complete, stop. **Partial assignment.**
Otherwise, let k^* be the vehicle with the biggest residual capacity among those not yet completed.

Step 2: Using a shortest path algorithm, find in G' the minimum load cycle containing any vertex of G'_{k^*}.
If such a cycle does not exist, or its associated load exceeds Q_{k^*} , mark vehicle k^* as complete and go to Step 1.

Step 3: Add the edges of the above cycle to E'_{k^*} , update Q_{k^*} and delete from G' the corresponding edges.
If the load associated to G' is zero, stop. **Complete assignment.**
Otherwise, go to Step 1.

If the above procedure produces a complete assignment we should go directly to build the routes, as explained in 3.4. However, the first application of the cycle assignment procedure results, generally, in a partial assignment, i.e. in G' there are required edges of the original graph G not assigned to any vehicle. Typically, the resulting graph G' consists of one or more cycles, each one with such a load that can not be assigned to any vehicle. In order to obtain a better assignment, two improvement procedures have been designed.

Note that in G' there are artificial edges with zero load, as well as required ones. Both procedures try to interchange edges of G' with edges of some graph G'_k , maintaining graph G'_k even and connected, provided that the residual capacity of vehicle k allows it and the load associated to the edges of G' decreases with such an interchange. In what follows, by interchanging a set of edges S of G' with a set of edges T of G'_k , it is meant to do $G' = G' - S + T$ and $G'_k = G'_k - T + S$, and update $Q_k = Q_k - q(S) + q(T)$.

The first improvement procedure examines the set of artificial edges assigned to a vehicle. Let i and j be the terminal vertices of such and edge and let k be the vehicle. If the required edge $e = (i, j)$ exists in G' and $q_e \leq Q_k$, the change of the artificial edge by the required one takes place. The procedure ends when such an interchange is no longer possible.

The second improvement procedure considers more general interchanges: Those of S, the set of edges of a path in G', with T, the edges of a path in some G'_k with the same terminal vertices. A detailed description of the procedure is given in what follows:

Improvement Procedure 2:

Step 0: (Init)
Let G'' be the graph with set of vertices V and no edges.
Do Flag = 0.

Step 1: While $q(G') \neq 0$, do:
Begin
- Find a simple cycle C in G'.
- For every pair of vertices i, j of C, find P_1 and P_2, the two sections of C determined by i and j, let q_1 and q_2 be the associated loads and do:
  - For every vehicle $k, k = 1, \cdots, K$, such that i and j belong to G'_k, compute P_k, the minimum load path in G'_k between i and j, and let q_3 be the associated load.
    If for some value of $r = 1, 2\, q_r - q_e \leq Q_k$ and $q_r > q_3$ (the interchange is both possible and convenient), interchange the set of edges of P_r in G' with that of P_k in G'_k, do Flag = 1.
  - If no interchange has been made, change the set of edges of C from G' to G''.
    End.

Step 2: If Flag = 0, **Partial assignment.** Stop
If $q(G'') = 0$, **Complete assignment.** Stop.
Otherwise, do $G' = G''$, $G'' = \emptyset$, Flag = 0. Go to Step 1.

Note that, in Step 1, all the paths in G'_k between i and j could have been considered instead of only the one with minimum load, but at a greater computational effort. For the same reason, only paths in G' that are sections of a simple cycle have been considered.

The three procedures: assignment of cycles, improvement 1 and improvement 2, are applied successively as long as interchanges are produced. If a complete assignment is obtained in any of them, the algorithm will proceed to construct the routes. Otherwise, if at the end of the procedures there is some required edge in G' not yet assigned,a GAP will be solved, as described in the next section. In any case, let $E_k = E'_k \cap E^R$, i.e. E_k is the set of required edges assigned to vehicle k by the above method.

3.3 A Generalized Assignment Problem

Consider the following GAP:

$$\text{Min} \sum_{e \in E^R} \sum_{k=1}^{K} a_{ek}\, y_{ek}$$

$$\sum_{k=1}^{K} y_{ek} = 1 \qquad \forall e \in E^R$$

$$\sum_{e \in E^R} q_e\, y_{ek} \leq Q \qquad \forall k = 1, \cdots, K$$

$$y_{ek} \in \{0,1\} \qquad \forall e \in E^R,\ \forall k = 1, \cdots, K$$

where $y_{ek} = 1$ iff the required edge e is assigned to vehicle k and 0 otherwise. The first set of constraints assures that each required edge is assigned to only one vehicle and the second set, that the capacity of the vehicles is not exceeded. Costs a_{ek} take into account the partial assignment obtained using the procedure described in the previous section and are set as follows:

$$a_{ek} = 0 \text{ iff } e \in E_k \quad \text{and}$$

$$a_{ek} = \text{Min}\,\{s_{1r} + c'_e + s_{ti_k} + s_{i_k 1},\ s_{1t} + c'_e + s_{ri_k} + s_{i_k 1}\} - 2s_{i_k 1} \qquad \forall e \notin E_k$$

where r and t are the terminal vertices of edge e and i_k is the center associated to vehicle k. Obviously, this cost represents the difference between the cost of just going and coming back from the depot to center i_k and the cost of doing it and furthermore servicing edge e; it approximates the cost of assigning edge e to vehicle k.

The solution to the above GAP produces an assignment of required edges to vehicles satisfying the capacity constraints provided that one exists (otherwise, the CARP would be infeasible). Usually, this assignment respects much of the partial assignment produced by the procedures of the previous section, therefore, the routes for the vehicles can be constructed at a small additional cost.

In order to solve the GAP, the "branch and bound" method of Ross and Soland (1975) has been implemented, using in the computation of lower bounds the algorithm of Fayard and Plateau (1982) for the resolution of the Knapsack Problem. This algorithm generates a sequence (usually a small number) of feasible solutions to the GAP with decreasing costs until the optimum is reached. Each of those feasible solutions is used to generate a solution for the CARP by using the method described in the following section. Since the assignment costs a_{ek} are computed heuristically, suboptimal solutions for the GAP may generate solutions for the CARP better than the one generated by the optimal GAP solution.

The idea of solving a GAP to obtain an assignment of customers to vehicles satisfying capacity constraints in order to construct the routes for the vehicles, was first proposed by Fisher and Jaikumar (1981) for the Vehicle Routing Problem (VRP).

3.4 Routes generation

Once the subsets E_k (with $q(E_k) \leq Q$), $k = 1, \cdots, K$, a partition of the original set of required edges E^R, have been determined, all that is needed in order to obtain a feasible solution to the CARP is to generate, for each vehicle k, a route servicing each edge of E_k, starting and finishing at the depot. Obviously, each one of these routes will consist of required edges, which must be serviced while being traversed by the vehicle, and, possibly, of artificial edges that will only be traversed.

For each $k = 1, \cdots, K$, let $E_k^+ = E_k$ if any edge of E_k is incident with the depot. Otherwise, in order to build E_k^+, add to E_k one artificial edge with zero load representing the depot; distances from its terminal vertices to the other vertices of the graph are equal to those from the depot. Let G_k^+ be the graph induced by E_k^+. If G_k^+ is connected, the problem of generating a route traversing all the edges at a minimum cost reduces to solving the Chinese Postman Problem (CPP) (Edmonds and Johnson (1973)). Otherwise, a Rural Postman Problem (RPP) has to be solved where the set of required edges is E_k^+. As this last problem is NP–Hard, it is solved heuristically using the algorithm proposed by Frederickson (1978) and Christofides *et al.*(1981), consisting of:

1) Adding a set of artificial edges to G_k^+, with minimum total cost, to obtain a connected graph. This is accomplished by solving a Shortest Spanning Tree Problem (SST).
2) Add to the resulting graph, that will be denoted too as G_k^+, a set of artificial edges, with minimum total cost, to obtain an even graph. This is accomplished by solving a Minimum Cost Matching Problem on the odd vertices of G_k^+.

In both problems the costs are set equal to the distances, s_{ij}, between every pair of vertices incident with edges in E_k^+.

For every $k = 1, \cdots, K$, the resulting graph is even and connected and can be traversed by vehicle k without repeating any edge. Therefore these graphs provide a feasible solution to the CARP. Its cost will be the sum of the costs of the artificial edges added to each vehicle plus the fixed cost given by the sum of the servicing costs of all the required edges in the original graph G.

As mentioned before, this procedure is applied to each solution of the GAP generated by the branch and bound algorithm, thus obtaining the corresponding CARP solution and, finally, the one with minimum cost is selected.

3.5 Improving the solution

Once a feasible solution to the CARP has been obtained, it is possible, in some cases, to improve the solution by applying a simple procedure which is described in what follows.

Let G_k^+ be the graph corresponding to each vehicle $k = 1, \cdots, K$ in the feasible solution. If there exists a pair of vertices i, j such that:

- for some vehicle t, G_t^+ contains the required edge $e = (i, j)$ and an artificial edge (i, j),
- the removal of these edges does not disconnect the graph G_t^+, and
- there exists another vehicle r, such that graph G_r^+ contains an artificial edge (i, j) and its residual load is not less than q_e,

then changing the required edge $e = (i, j)$ from G_t^+ to G_r^+ and removing the artificial edge (i, j) from both graphs produces a new feasible solution to the CARP with lower cost.

The procedure is applied until no further improvements are possible.

4. COMPUTATIONAL RESULTS

The heuristic algorithm described in Section 3 has been implemented in Fortran and applied to a set of test instances, using a UNIVAC 1100 / 60 computer. Table I describes the main characteristics of the 10 graphs used: number of vertices (N), number of edges (M) and total load (Q_T). In all the instances, every edge had a positive load. Loads and costs have been generated between 1 and 15, and the servicing cost was always greater than the traversing cost.

Table II shows the results obtained with the algorithm. For each instance, the graph used, the number of vehicles (K) and its capacity (Q), the best known lower bound (LB) (see Benavent *et al.*(1987)), the cost of the heuristic solution $(Z(H))$, its deviation from the lower bound $(D = (Z(H) - LB)/LB)$ and total CPU time in seconds (T) are given.

TABLE I

Characteristics of the graphs

	1	2	3	4	5	6	7	8	9	10
n	24	24	24	41	34	31	40	30	50	50
m	39	34	35	69	65	50	66	63	92	97
Q_T	358	310	137	627	614	457	652	566	654	704

TABLE II

Computational results

Graph	K	Q	LB	$Z(H)$	$D(\%)$	T
1	2	200	247	256	3.6	0.8
1	3	120	247	260	5.3	1.6
2	2	180	297	324	9.1	0.7
2	3	120	309	346	11.9	0.6
3	2	80	103	108	4.8	0.6
3	3	50	107	115	7.5	0.7
4	3	225	516	536	3.9	3.4
4	4	170	522	576	10.3	3.6
4	5	130	528	624	18.2	130.5*
5	3	220	562	594	5.7	1.8
5	4	165	580	620	8.3	10.3
5	5	130	598	646	8.0	16.1
6	3	170	330	337	2.1	1.1
6	4	120	336	351	4.5	14.6
7	3	200	382	382	0.0	1.5
7	4	150	382	414	8.4	2.0
8	3	200	522	556	6.5	20.9
8	4	150	531	566	6.6	138.9*
9	3	235	448	462	3.1	3.2
9	4	175	453	484	6.8	3.5
9	5	140	459	494	7.6	12.1
10	3	250	637	658	3.3	5.5
10	4	190	645	665	3.1	257.3*
10	5	150	653	684	4.7	4.7

In some instances (indicated by an asterisk) the branch and bound algorithm for the GAP was stopped before reaching optimality, after examining 20000 nodes in the tree search. Only the GAP solutions so far obtained were used to generate CARP solutions. Note that it occurs in only 3 out of the 24 tested instances.

On average, the heuristic solution has a cost 6.4% above the lower bound and has been obtained within a computational time that is less than 21 seconds, except for the 3 mentioned cases. Note that in graph 7 with $K = 3$, the feasible solution is optimal.

5. COMMENTS

The described heuristic algorithm has some characteristics that rend it more attractive than the existing ones for this problem:

- it always finds a feasible solution if one exists, because the GAP includes the constraints that assure the problem feasibility,
- the solutions consist of routes that, in most cases, are balanced and "compacted".
- the algorithm could be easily adapted to the case in which the vehicles capacities are different.
- by applying the algorithm to different fleet sizes it is possible to evaluate the convenience of acquiring or not more vehicles.

6. APPENDIX

Consider the instance of the CARP associated to graph 10 (depicted in Figure 1) with $K = 3$ and $Q = 250$. The depot is indicated by 1, numbers next to the edges correspond with the data in Table III, which contains, for each edge e, the load q_e the servicing cost c'_e and c_e, the cost of just traversing it.

Figure 2 shows the graph obtained at the end of the cycle assignment procedure (corresponding to section 3.2). The selected centers are indicated by an asterisk. The cycles assigned to each vehicle are indicated with dotted, dashed and continous lines respectively. Non–straight lines are artificial edges added to make the graph even. Note that there are three unassigned cycles, corresponding to edges 51, 59, 60, 67, 80, 81 and 89.

Finally, Figure 3 (in which non–straight lines correspond to edges traversed without been serviced) represents the feasible solution obtained after applying all the stages of the above described heuristic. Note that the final assignment of required edges to vehicles given by the solution to the GAP is very similar to the partial assignment given by the cycle assignment procedure.

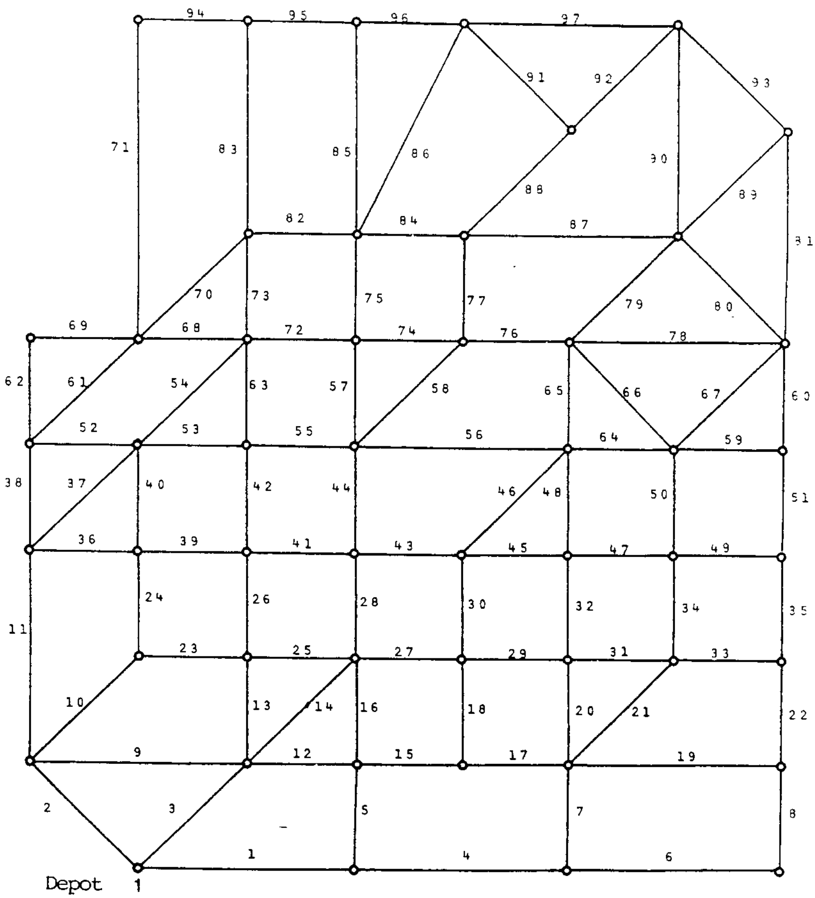

Figure 1: The Original Graph

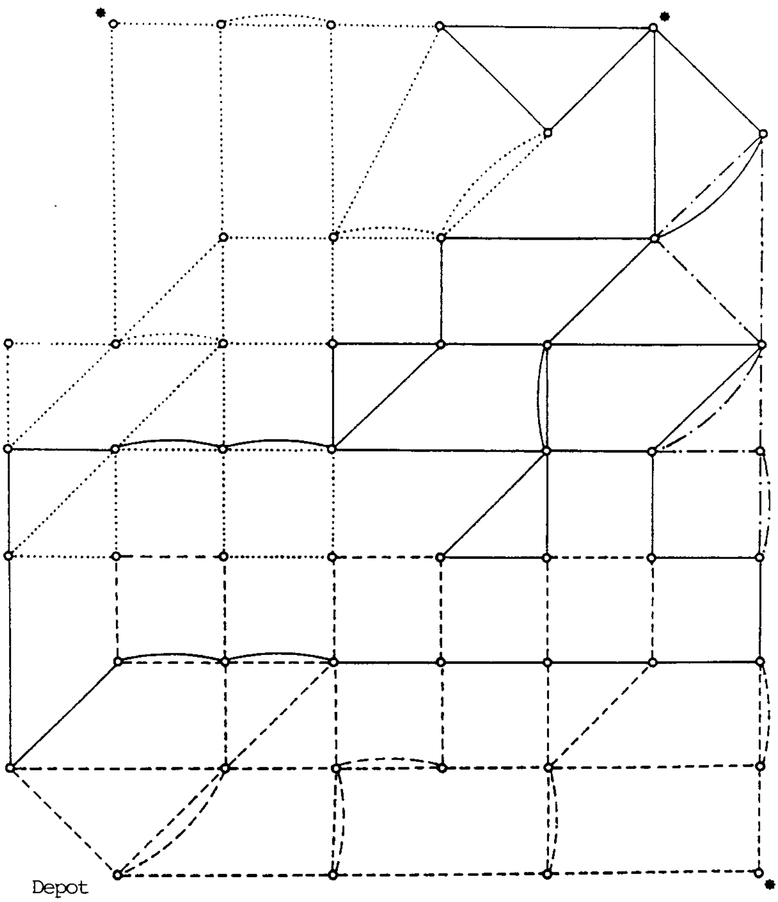

Figure 2: Cycles Assignment
Cycles .... Total load 175
Cycles – – – – – – Total load 238
Cycles ———— Total load 250
Unassigned edges – . – . – . – . – .

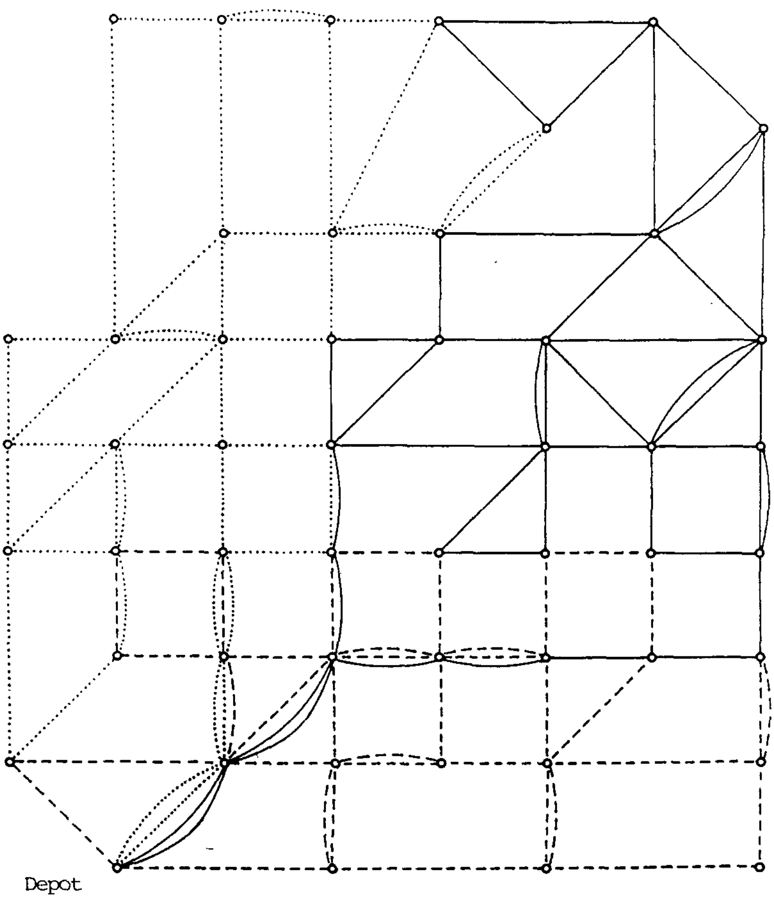

Figure 3: The feasible Solution

Route	....	Load 224	Cost 202
Route	– – – – – –	Load 235	Cost 217
Route	————	Load 245	Cost 239

TABLE III

(edge #) q_e, c'_e, c_e

(1)13,14,10	(2)8,7,7	(3)7,5,2	(4)10,9,6	(5)7,5,1	(6)9,8,6	(7)6,5,5
(8)6,5,3	(9)10,9,9	(10)9,5,3	(11)12,10,9	(12)7,5,5	(13)7,5,2	(14)8,7,3
(15)6,4,2	(16)6,5,4	(17)5,3,3	(18)7,5,3	(19),9,9,8	(20)8,4,2	(21)9,8,5
(22)11,10,7	(23)5,4,3	(24)8,5,1	(25)4,5,1	(26)7,5,2	(27)6,6,2	(28)6,5,3
(29)5,3,1	(30)9,8,5	(31)7,6,2	(32)6,5,1	(33)8,8,7	(34)7,5,3	(35)8,8,7
(36)4,2,1	(37)9,5,2	(38)10,9,5	(39)9,9,3	(40)6,5,1	(41)7,5,4	(42)12,12,9
(43)10,8,5	(44)7,5,4	(45)6,4,2	(46)9,8,6	(47)8,7,6	(48)10,7,3	(49)7,5,2
(50)11,10,5	(51)6,5,3	(52)4,2,2	(53)3,2,1	(54)10,12,5	(55)6,6,3	(56)12,10,7
(57)9,8,6	(58)12,11,8	(59)4,2,1	(60)10,10,6	(61)8,9,6	(62)7,5,5	(63)5,3,1
(64)5,5,3	(65)6,4,2	(66)7,5,3	(67)9,8,5	(68)6,5,4	(69)8,5,3	(70)7,4,3
(71)10,9,6	(72)5,4,2	(73)5,3,1	(74)6,5,3	(75)4,3,3	(76)4,2,2	(77)6,5,2
(78)9,10,7	(79)7,5,2	(80)8,6,3	(81)9,8,7	(82)4,3,3	(83)3,2,2	(84)6,74
(85)3,2,1	(86)4,3,2	(87)9,8,4	(88)3,3,1	(89)4,3,1	(90)6,6,6	(91)5,5,4
(92)7,6,5	(93)7,5,3	(94)7,6,4	(95)6,5,3	(96)10,9,7	(97)12,10,10	

7. REFERENCES

[1] **Assad, A., Pearn, W.** and **Golden, B.** (1987). "The Capacitated Chinese Postman Problem: Lower Bound and Solvable Cases". *American J. of Mathematical and Man. Sci.*, 7, 63-88.

[2] **Beltrami, E.** and **Bodin, L.** (1974). "Networks and Vehicle Routing for Municipal Waste Collection". *Networks 4 (1)*, 65-94.

[3] **Benavent, E., Campos, V., Corbern, A.** and **Mota, E.** (1983). "Problemas de Rutas por Arcos". *Qestii 7*, 479-490.

[4] **Benavent, E., Campos, V., Corbern, A.** and **Mota, E.** (1985). "A Heuristic Algorithm for the Capacitated Chinese Postman Problem". Paper presented at the 7th European OR Congress, Bologna, Italy.

[5] **Benavent, E., Campos, V., Corbern, A.** and **Mota, E.** (1987). "The Capacitated Chinese Postman Problem. Part II: Lower Bounds". Technical Report. Dpto. de Estadstica e Investigacin Operativa. Universidad de Valencia. A revised version of this paper will appear in Networks.

[6] **Bodin, L.** and **Golden, B.** (1981). "Classification in Vehicle Routing and Scheduling". *Networks 11 (2)*, 97-108.

[7] **Bodin, L., Golden, B., Assad, A.** and **Ball, M.** (1983). "Routing and Scheduling of Vehicles and Crews". *Computers & Operations Research 10*, 63-211.

[8] **Bodin, L., Fagin, G., Welebny, R.** and **Greenberg, J.** (1989). "The design of a computerized santitation vehicle routing and scheduling system for the town of Oyster Bay, New York". *Computers & Operations Research 16*,45-54.

[9] **Chapleau, L., Ferland, J., Lapalme, G.** and **Rousseau, J.M.** (1984). "A Parallel Insert Method for the Capacitated Arc Routing Problem". *Operations Research Letters 3*,95-99.

[10] **Christofides, N.** (1973). "The Optimum traversal of a Graph". *Omega 1 (6)*,719-732.

[11] **Christofides, N., Campos, V., Corbern, A.** and **Mota, E.** (1981). "An Algorithm for the Rural Postman Problem". Imperial College Technical Report ICOR 81-5. London.

[12] **Edmonds, J.** and **Johnson, E.L.** (1973). "Matching, Euler Tours and the Chinese Postman Problem". *Mathematical Programming 5*,88-124.

[13] **Fayard, D.** and **Plateau, G.** (1982). "Algorithm 47: An Algorithm for the Solution of the 0-1 Knapsack Problem". *Computing 28*,269-287.

[14] **Fisher, M.** and **Jaikumar, R.** (1981). "A Generalized Assignment Heuristic for Vehicle Routing". *Networks 11*,109-124.

[15] **Frederikson, G.** (1979). "Approximation Algorithms for Some Postman Problems". *Journal of the Association for Computing Machinery 26,(3)*,538-554.

[16] **Golden, B.** and **Wong, R.** (1981). "Capacitated Arc Routing Problems". *Networks 11*,305-315.

[17] **Golden, B., De Armon, J.S.** and **Baker, E.K.** (1983). "Computational Experiments with Algorithms for a Class of Routing Problems". *Computers & Operations Research 10, (1)*,47-59.

[18] **Lenstra, J.K.** and **Rinnooy-Kan, A.H.G.** (1981). "Complexity of Vehicle Routing and Scheduling Problems". *Networks 11*,221-227.

[19] **Male, J.W.** and **Liebman, J.C.** (1978). "Districting and Routing for Solid Waste Collection" *Journal of the Environmental Engineering Division* 104, EE1.

[20] **Pearn, W.L.** (1988). "New Lower Bounds for the Capacitated Arc Routing Problem". *Networks 18*,181-191.

[21] **Ross, G.** and **Soland, R.** (1975). "A Branch and Bound Algorithm for the Generalized Assignment Problem". *Mathematical Programming 8*, 91-103.

[22] **Stern. H.** and **Dror, M.** (1979). "Routing electric meter readers". *Computers & Operations Research 6*,209-223.

[23] **Zaw Win** (1988). "Contributions to Routing Problems". PhD Dissertation. Universitat-Augsburg.