European Journal of Operational Research 22 (1985) 329–337
North-Holland

The fleet size and mix problem for capacitated arc routing

Gündüz ULUSOY
Department of Industrial Engineering, Boğaziçi University, Bebek, Istanbul, Turkey

Abstract: There have been several attempts to solve the capacitated arc routing problem with m vehicles starting their tours from a central node. The objective has been to minimize the total distance travelled. In the problem treated here we also have the fixed costs of the vehicles included in the objective function. A set of vehicle capacities with their respective costs are used. Thus the objective function becomes a combination of fixed and variable costs. The solution procedure consists of four phases. In the first phase, a Chinese or rural postman problem is solved depending on whether all or some of the arcs in the network demand service with the objective of minimizing the total distance travelled. It results in a tour called the giant tour. In the second phase, the giant tour is partitioned into single vehicle subtours feasible with respect to the constraints. A new network is constructed with the node set corresponding to the arcs of the giant tour and with the arc set consisting of the subtours of the giant tour. The arc costs include both the fixed and variable costs of the subtours. The third phase consists of solving the shortest path problem on this new network to result in the least cost set of subtours represented on the new network. In the last phase a postprocessor is applied to the solution to improve it. The procedure is repeated for different giant tours to improve the final solution. The problem is extended to the case where there can be upper bounds on the number of vehicles with given capacities using a branch and bound method. Extension to directed networks is given. Some computational results are reported.

Keywords: Physical distribution, networks

Introduction

The capacitated arc routing problem (CARP) constitutes an important class of vehicle routing problems. It has applications in areas such as household refuse collection, street sweeping, inspection of electrical and pipeline networks for faults, mail delivery, etc.

A generic problem for CARP is the Chinese Postman Problem (CPP). CPP is stated as follows: Let N be the set of nodes and A_r the set of arcs. Given a network $G_1 = (N, A_r)$ with each arc $(i, j) \in A_r$ having a positive arc traversal cost c_{ij}, find the minimum total cost tour such that each arc of A_r is traversed at least once (Kwan, 1962).

Received February 1983; revised October 1984 and March 1985

CPP is solved by finding an Euler tour on G_1, if one exists and if not, then by finding the minimum total cost subset of arcs of A_r to be added to A_r to obtain A_r^* so that an Euler tour can now be constructed on the new graph $G_1^* = (N, A_r^*)$. Efficient algorithms have been constructed for both the directed and undirected CPP (Edmonds and Johnson, 1973). The mixed CPP has been discussed in great detail by Minieka (1979).

Another generic problem for CARP is the Rural Postman Problem (RPP) defined by Orloff (1974). RPP differs from CPP in that we are allowed to have arcs in the network which are not required to be traversed, i.e. their demand for service is zero. The set of arcs that are not required to be traversed is denoted by A_0. RPP is solved on $G_2 = (N, A)$ where $A = A_r \cup A_0$. Solution procedures for both undirected and directed cases are developed making use of 1-matching and transportation problems respectively (Orloff, 1974). An NP-completeness proof for RPP is provided by Lenstra and Rinnooy Kan (1981).

Both CPP and RPP can be considered as core problems for the vehicle routing problems where a single vehicle with infinite capacity case is far from representing real problems. There have been several attempts to extend CPP and RPP so as to arrive at more realistic representations of the vehicle routing problems serving arcs.

One such approach has been suggested by Christofides (1973). He imposes a finite capacity W to the vehicles. Each arc $(i, j) \in A_r$ has a positive demand for service. Side constraints on vehicle tours such as maximum length and/or maximum time might be imposed. This version of CPP is called the Capacitated Chinese Postman Problem (CCPP). CCPP has been addressed also by Golden and Wong (1981).

CARP has been defined by Golden and Wong (1981) and is essentially an extension of RPP to the case where the vehicles are restricted to have a finite capacity W. It is analogous to the extension of CPP to CCPP. CARP is defined as follows: Given a network $G_2 = (N, A)$, find a minimum total cost tour satisfying the demands for service q_{ij} for each arc $(i, j) \in A_r$ using vehicles each with a finite capacity W where the vehicle tours are possibly subject to side constraints such as maximum length and/or maximum time.

Golden and Wong have shown that CARP is NP-hard. They have proposed an algorithm based on the Clarke and Wright savings idea. The algorithm starts with all the demand arcs being served by a separate cycle. In the second phase of the algorithm, starting with the largest cycle available, possibilities are explored whether a demand arc on a smaller cycle can be serviced on the larger cycle The third phase consists of merging cycles subject to capacity and other constraints which yield the largest positive savings. The algorithm stops when the objective function value cannot be improved upon by any feasible merging of cycles. Two implementations on seven CCPP's have given promising results. The heuristic procedure proposed has never yielded a solution above 18% of the lower bound. It has been reported that the algorithm doesn't seem to perform well with large scale networks and when arcs with zero demand are present (Chapleau et al., 1982).

Another approach to solve CARP has been provided by Chapleau, et al. (1982). They have investigated the problem for specifically school bus routing but they report that the method is easily adaptable to other areas as well. The algorithm is based on the insertion method generating many routes in parallel. The idea has been applied previously to generate travelling salesman tours (Rosenkrantz et al., 1974). When constructing the routes, secondary constraints requiring the routes to be balanced with respect to distance and loading and progressively approaching the destination are taken into account. Two strategies are employed:

(1) Given an arc, determine the existing route into which it should be inserted to minimize the detour incurred.

(2) Given a route, determine which remaining arc should be inserted next. These two strategies are used alternatingly together with a pick-up distance concept to reach a final solution. The pick-up distance is defined to be the distance beyond which only insertions leading straight to the destination are allowed. Large networks are claimed to be handled in reasonable computation times.

Ulusoy (1981) has presented an algorithm for solving the fleet size and mix problem for CARP. According to the classification given by Bodin and Golden (1981) the solution approach can be classified as a route first – cluster second approach. A giant tour is obtained by solving the CPP on the network at hand. Then the giant tour is partitioned to obtain the vehicle tours. Partitioning approach is a rather promising one as has also been demonstrated by Frederickson et al. (1978) for the multipostman problem (k-CPP). The following is an extension of the study by Ulusoy (1981).

2. Statement of the problem

2.1. The objective function

In all the approaches above operation cost is taken to be a linear function of distance covered. In the approaches above the fixed cost component related to the vehicles is not taken into account. The fleet size is either fixed or is determined by the optimal routing with respect to the operation cost. The capacity of each vehicle is fixed and is the same for all vehicles, i.e. we have a homogeneous fleet. Obviously, these two dimensions can be parameterized in order to search for better solutions but the models do not include them as decision variables.

In this paper we extend the homogeneous fleet case to the heterogeneous fleet case. In other words, the fleet size and mix become decision variables. For determining the fleet size and mix we need to consider fixed cost for the vehicles simultaneously with the operation cost of the vehicles. The operation cost is taken as a linear function of distance covered.

The objective is to minimize the total cost which is the sum of fixed and variable costs.

2.2. The operating conditions

The operating conditions imposed here can be stated as follows:

(i) There is a single depot where the whole fleet is located initially.

(ii) Vehicles return to the depot after they are finished.

(iii) All service related activities take place on the arcs and at the depot. In other words, there are no dump sites as might be the case, e.g., in the garbage collection problem.

(iv) Each vehicle has a maximum operation time and/or a maximum range.

(v) All demands have to be satisfied.

(vi) No split service is allowed, i.e. all the service demand of an arc has to be satisfied by the same vehicle. In cases where the service demand is larger than the largest vehicle capacity, then the demand is met by as many fully loaded vehicles only serving this arc until the remaining service demand is less than the largest vehicle capacity. The fully loaded vehicle is selected to be the one with the lowest unit cost.

(vii) A vehicle with the smallest capacity meeting the demand of a vehicle tour will always be available to be assigned to that tour.

A mathematical formulation of the problem will not be given here.

3. Solution approach

3.1. Phases of the solution procedure

The solution procedure developed here for the fleet size and mix extension of CARP can be classified as a route first – cluster second approach. It will be stated for the undirected case. The extension to the directed case is given later in the paper.

The procedure consists of four phases:

1. Find a giant tour.

2. Partition the giant tour into subtours each corresponding to a feasible vehicle tour.

3. Select the least cost set of vehicle tours satisfying the demands from the set of vehicle tours generated.

4. Repeat the above steps until some criterion of stopping is satisfied.

3.2. Obtaining the giant tour

The giant tour on $G=(N, A)$ is found without taking into account the fixed cost component, the capacity constraints, and any other type of constraints which might be present. Note that if $q_{ij} > 0$ for all $(i, j) \in A$, then the giant tour is a solution to CPP on G. Otherwise, it is obtained by solving RPP on G. All the arcs on the giant tour which are not part of the required arc set A_r are called no-service arcs. If an arc $(i, j) \in A_r$ appears in the giant tour more than the required times, all its copies are called no-service arcs, too. The network on which the giant tour is constructed is denoted by G^E.

3.3. Partitioning the giant tour

Once the giant tour is obtained, it is partitioned into subtours each corresponding to a feasible vehicle tour. The partitioning of the giant tour takes place on a transformed network denoted by G^*. G^* is defined by its (i) node set, (ii) arc set, (iii) incidence relationships, and (iv) arc cost structure. The rules for constructing G^* are given below.

First we reduce the network G^E.

N1. If there is any chain of no-service arcs on the Euler tour, then it is reduced to an arc with initial and final nodes being those of the chain and its cost being equal to the total cost of the chain.

The following rule determines the node set of G^*.

N2. Each node on G corresponds to an arc on the Euler tour except the first node. The first node on

G^* corresponds to the depot node (node 1) of G. The arcs on the Euler tour are numbered in the same order as they appear in the tour and are mapped to the nodes of G^* in the same order with the first arc corresponding to node 2 on G^*. If the last node of G^* represents a no-service arc, then it is eliminated from the node set.

Each arc on G^* corresponds to a feasible vehicle tour on G. For example, an arc (k, h) on G^* corresponds to a vehicle tour on G which includes all the arcs represented by the nodes $(k+1), \ldots, h$. In Figure 1, a portion of an example network G^* is shown. The node number in G^* is given in the upper half of each node and the corresponding arc in the vehicle tour on G is given in the lower half. According to the convention adopted here, arc (7, 8) on G^* corresponds to a vehicle tour on G serving only arc (4, 6). Similarly, arc (7, 9) on G^* corresponds to a vehicle tour on G serving only arcs (4, 6) and (6, 9). How a vehicle tour serving those arcs is formed will be dealt with later in the paper.

The arcs present in the arc set of G^* are determined by the following rule.

N3. An arc is incident from each node h into each node k with $h < k$, given that the corresponding vehicle tour does not violate any of the side constraints that might be present.

The presence of no-service arcs on G is exploited to reduce the number of arcs on G^* as determined by N3. Let node i correspo[illegible] to a no-service arc on G.

N4. No arc is incident into node i except arc $(i-1, i)$.

N5. No arc is incident from node $(i-1)$ except arc $(i-1, i)$

These two rules are formulated so as to prevent no-service arcs to be present in the vehicle tours without any purpose and thus leading to inferior vehicle tours.

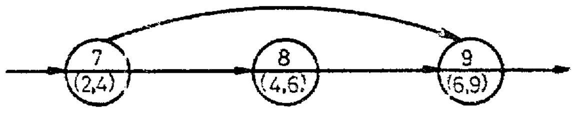

Figure 1. Portion of an example network

When partitioning the Euler tour we have obtained arc sequences from the Euler tour which are to be transformed into vehicle tours. For that purpose we distinguish between two cases: (i) Arcs on the sequence form a cycle, (ii) arcs on the sequence do not form a cycle.

If the arcs of the sequence form a cycle, then:

S1. If node 1 is a member of the sequence, then the sequence corresponds to a vehicle tour.

S2. If node 1 is not a member of the sequence, then connect node 1 to the node of the cycle with the least cost chain to node 1.

If the arcs of the sequence do not form a cycle, then:

S3. If node 1 is a member of the sequence, then connect the two odd degree nodes with the least cost chain.

S4. If node 1 is not a member of the sequence, then connect node 1 to the two odd degree nodes using least cost chains.

Let us see an example of the construction of G^* and see how an arc on G^* represents a vehicle tour on G. Consider the network in Figure 2 on which each arc is an arc with positive demand for service. Thus a CPP will be solved to obtain a giant tour. Such a giant tour is given by the following Euler tour: {1, 2, 3, 1, 4, 3, 1}. Note that arc (3, 1) is used twice. The second one is chosen as the no-service arc which reduces the size of G^* by one node according to rule N2. We have 6 nodes on G^* which is shown in Figure 3. All arcs resulting from N5 are included in G^* since no capacity or distance constraints are observed.

As an example, consider arc (3, 6) on G^*. The

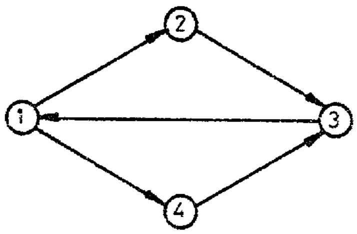

Figure 2. Original network G

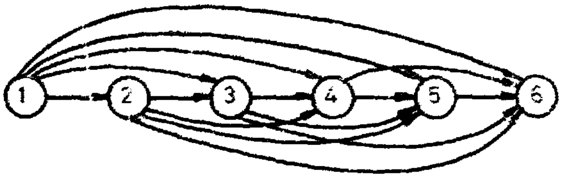

Figure 3. Transformed network G^*

corresponding vehicle tour includes arcs represented by the nodes 4, 5, and 6 on G^*. Using rule S1 we get the vehicle tour in Figure 4(a). The vehicle tour corresponding to arc (1, 5) includes arcs represented by nodes 2, 3, 4, and 5 on G^* and is given in Figure 4(b). It is obtained using rule S3 as a result of which the arc (4, 1) is included to complete the tour.

The construction of G^* will be complete once we assign the arc cost $\bar{c}_{ij}$ for each arc (i, j) on G^*. The arc costs are determined according to the following two rules:

C1. Given node $(k+1)$ on G^* is a no-service arc node (i.e. it represents a no-service arc on the giant tour), then $\bar{c}_{k,k+1} = 0$.

C2. $\bar{c}_{kh}$ is the sum of the arc costs c_{ij} included in the vehicle tour and the fixed cost of the least capacity vehicle meeting the demand of the corresponding vehicle tour.

3.4. Selecting the least cost set of vehicle tours on G^*

Having constructed G^* we are now ready to select the least cost set of vehicle tours represented on G^*. This is accomplished by solving for the shortest path from node 1 to node m where node m corresponds to the last arc with positive service demand on the giant tour. The shortest path is obtained by applying Dijkstra's algorithm (Dijkstra, 1959). The acyclic structure of G^* improves the efficiency of the algorithm. The vehicle tours corresponding to the arcs of the shortest path form the least cost set of vehicle tours on G^*.

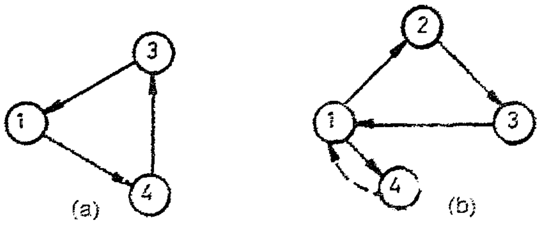

Figure 4. (a) Tour corresponding to arc (3, 6) of G^*. (b) Tour corresponding to arc (1, 5) of G^*.

3.5. Stopping criterion

Solving for the shortest route on G^* will result in the least cost set of vehicle tours for that particular G^* which is obtained for a given Euler tour on G^E. Different Euler tours constructed from the same G^E will result, in general, in different G^*'s. So far it seems to be impossible to devise a hill climbing type of heuristic algorithm where the generation of successive G^*'s will be stopped when the decrease in total cost drops below a preassigned value. Thus a reasonable approach seems to create as many different Euler tours only limited by G^E and computational resources available.

3.6. The algorithm

Two counters are introduced to control the number of distinct Euler tours. One of them is associated with the number of times different cycles are obtained, I, and the other, J, indicates the number of distinct Euler tours constructed from the same set of cycles. Note that $J \leqslant N_c$ where N_c is the number of cycles obtained.

Step 0. $i = 0$.
Step 1. Apply an appropriate algorithm to obtain G^E.
Step 2a. Construct cycles on G^E covering all arcs of it. $i \leftarrow i+1$, $j = 0$.
Step 2b. Construct a distinct Euler tour using the cycles of Step 2a. $j \leftarrow j+1$.
Step 3. Obtain the transformed graph G^* using the Euler tour of Step 2b.
Step 4. Obtain the shortest path on G^* from node 1 to node m.
Step 5. Determine the subtours corresponding to the arcs of the shortest path and the associated vehicle capacities. Eliminate subtours that consists of no-service arcs only. Compare the new solution with the current best solution and if it is better, declare it the current best solution.
Step 6. If $i = I$, go to Step 7. If $j = J$, go to Step 2a. Otherwise go to Step 2b.
Step 7. Stop. Declare the current best solution the final solution.

An example problem will be used to demonstrate the algorithm. But first let us see the extension of the above algorithm to the directed case.

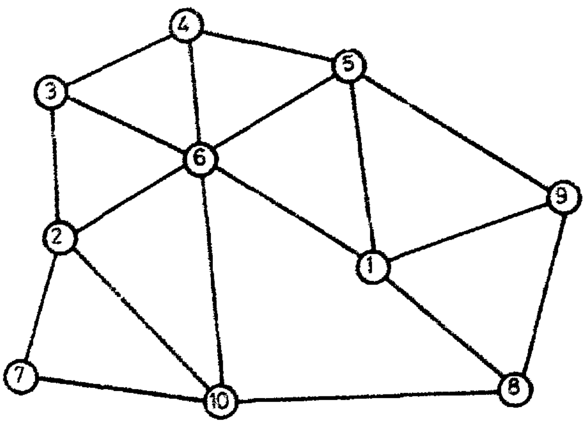

Figure 5. The original network

3.7. *Extension to the directed case*

When the cost matrix of G is asymmetric then, when obtaining the giant tour, a transportation problem is solved instead of the 1-matching problem. The rest of the algorithm applies as it is.

4. An example problem

Consider the network in Figure 5. The distance and demand matrices are combined together in the $Q \setminus C$ matrix in Figure 6 where the upper diagonal matrix represents the cost matrix and the lower diagonal matrix represents the demand matrix for arcs (i, j) of the original network G. We have three types of vehicl with capacities 50, 80, and 100. Their fixed costs are taken to be equal to their capacity.

We decide to apply Step 2a only once, i.e. $I = 1$. When applying Step 2a, we get two cycles:

$$\{1, 5, 6, 4, \underline{3, 4}, 5, 9, \underline{8, 9}, 1\}$$

and

$$\{1, 8, 10, 7, 2, 3, 6, 2, 10, 6, 1\}.$$

Thus there will be two distinct Euler tours for our purpose. The arcs (3, 4) and (8, 9) are no-service arcs and are underlined. The first Euler tour is obtained as:

$$\{1, 5, 6, 4, \underline{3, 4}, 5, 9, \underline{8, 9}, 1, 8, 10, 7, 2, 3, 6, 2, 10, 6, 1\}.$$

The transformed network G_1^* has 21 nodes the first one corresponding to node 1 of G and the remaining ones corresponding to the twenty arcs of the above Euler tour in the same order as they appear in that tour.

As an example of the arc costs $\bar{c}_{ij}$ let us calculate $\bar{c}_{14,17}$. The required arcs of this vehicle tour are (7, 2), (2, 3), and (3, 6) corresponding to nodes 15, 16, and 17 of G_1^*. The vehicle tour is constructed using rule S4. The vehicle tour is then $\{\underline{1, 6, 2, 7}, 2, 3, \underline{6, 1}\}$. The demands of the required arcs add up to 54. The smallest capacity vehicle that can handle this load is one with 80 units capacity with a fixed cost of 80. Then

$$\bar{c}_{14,17} = (23 + 38 + 33 + 31 + 53 + 23) + 80 = 314.$$

We observe that there is considerable deadheading and idle capacity costs and we can expect that this arc will not appear on the shortest path.

After calculating the cost for each feasible arc on G_1^* we apply Dijkstra's algorithm to get the shortest path on G_1^* as

$$\{1, 3, 8, 9, 10, 11, 16, 21\}.$$

The vehicle tours and the assigned vehicles' capacities are summarized in Table 1. Note that the

$$
Q \setminus C = \begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\left[\begin{array}{cccccccccc}
- & - & - & - & 26 & 23 & - & 25 & 32 & - \\
- & - & 31 & - & - & 38 & 33 & - & - & 40 \\
- & 12 & - & 42 & - & 53 & - & - & - & - \\
- & - & 42 & - & 30 & 40 & - & - & - & - \\
46 & - & - & 13 & - & 27 & - & - & 28 & - \\
38 & 17 & 19 & 3 & 14 & - & - & - & - & 27 \\
- & 23 & - & - & - & - & - & - & - & 45 \\
15 & - & - & - & - & - & - & - & 45 & 41 \\
65 & - & - & - & 36 & - & - & 41 & - & - \\
- & 6 & - & - & - & 16 & 19 & 26 & - & -
\end{array}\right]
\end{array}
$$

Figure 6. The distance and demand matrix

Table 1
The vehicle tours resulting from the first Euler tour

Vehicle tour	Vehicle capacity	Total demand	Vehicle tour cost
1, 5, 6, 1	80	60	76 + 80
1, 6, 4, 3, 4, 5, 9, 1	100	94	237 + 80
1, 9, 8, 1	50	41	102 + 50
1, 9, 1	80	65	64 + 80
1, 8, 10, 7, 2, 3, 6, 1	100	95	251 + 100
1, 6, 3, 6, 2, 10, 6, 1	100	96	257 + 100

vehicle tour $\{\underline{1, 8, 9, 1}\}$ is not included in Table 1 since it covers only the no-service are (8, 9).

The Euler tour has 20 arcs. This solution has 29 arcs and employs 6 vehicles. The total cost is 1497 units.

The second Euler tour is found to be:

$$\{1, 8, 10, 7, 2, 3, 6, 2, 10, 6, 1, 5, 6, 4, \underline{3, 4}, 5, 9, \underline{8, 9}, 1\}$$

The underlined arcs (3, 4) and (8, 9) are the no-service arcs. After transforming the second Euler tour to obtain G_2^*, the Dijkstra algorithm is applied to get the following shortest path on G_2^*:

$$\{1, 2, 7, 11, 12, 17, 19, 20, 21\}.$$

The vehicle tours and the vehicle capacities assigned are represented in Table 2 together with the demand and total cost for each vehicle tour. The total cost is 1477 units. The solution has 29 arcs compared to 20 arcs of the giant tour and employs 7 vehicles. The fourth and fifth vehicle tours of the first solution are kept in this solution, too. But the required arcs do not completely coincide except in the fourth vehicle tour.

Table 2
The vehicle tours resulting from the second Euler tour

Vehicle tour	Vehicle capacity	Total demand	Vehicle tour cost
1, 8, 1	50	15	50 + 50
1, 8, 1, 7, 2, 3, 6, 1	100	99	251 + 100
1, 6, 2, 10, 6, 1	80	77	151 + 80
1, 5, 1	50	46	52 + 50
1, 5, 6, 4, 3, 4, 5, 1	80	72	233 + 80
1, 5, 9, 8, 1	80	77	124 + 80
1, 9, 1	80	65	64 + 80

5. Postprocessor

As can be seen from the results of the example problem, the solution produced by the algorithm can still be improved upon. To test the solution for further improvements a postprocessor will be introduced. The postprocessor is based on the idea of exchanging no-service arcs with serviced arcs among the vehicle tours for more efficient utilization of vehicle capacities. The steps of the postprocessor are given as follows.

P1. List all the serviced arcs which are retraced with no service in decreasing order of service demand. Ignore those arcs for which the only retraced arcs occur in the same vehicle tour.

P2. Choose the arc with the largest service. Determine whether the deletion of this service demand from this vehicle tour will reduce the capacity requirement for that vehicle to lower levels. If so, go to P3. Otherwise check whether there exists a combination of arcs from the list in that vehicle tour which, when deleted, will reduce the capacity requirement for that vehicle tour to lower levels. If so, pick that combination which results in best capacity utilization for that vehicle and go to P3. Otherwise eliminate the arc under consideration from the list and repeat this step.

P3. Consider all vehicle tours in which this arc is retraced with no service. Determine the exchange causing the largest reduction in cost and make the necessary changes.

If there are more than one alternative causing the same reduction in cost, choose the one resulting in maximum capacity utilization. If no exchange results in a cost reduction, then don't make any changes. Eliminate the arc from the list and return to P2 if the list is not empty, otherwise stop. If we come from P2 with a combination of arcs, then this step is applied accordingly.

Additional constraints such as maximum operation time, maximum number of stops, etc. should be taken into account when making the exchanges.

Let us apply the postprocessor routine to the example problem above. Let us consider the vehicle tours resulting from the second Euler tour and apply P1 to get Table 3.

Table 3
Application of P1 to the second Euler tour

List of arcs	Service demand
(1, 5)	46
(1, 6)	38
(1, 8)	15

Table 4
Vehicle availabilities

Type	Capacity	Number available
1	50	3
2	80	3
3	100	2

For (1, 5), there are no vehicle tours that can accomodate its service demand by going to larger capacity levels. The same holds for (1, 6). The service demand for (1, 8) can be met by the 6th vehicle tour and this results in a reduction in cost. Thus that exchange is made which eliminates the first vehicle tour and increases the vehicle capacity of the 6th tour from 80 to 100. The total cost is reduced from the previous 1477 to 1397.

6. Extension to the limited availability case

Having solved the problem for the case of unlimited vehicle availability for each capacity represented we can now look to the case where there can be an upper bound on the number of each vehicle type. In that case, for the solution of the problem we apply a branch and bound procedure with the following characteristics.

The lower bound z_k at node k of the decision tree is provided by the solution of the shortest path problem on the modified network at node k, G_k^*. The branching is done according to the following rule: Given the infeasible solution to the problem on G_k^* at node k of the decision tree, the vehicle type with the least number of vehicle assignments, say h, where h is larger than the number of vehicles available of that capacity, h branches are created at node k, one for each vehicle assigned. At each branch the corresponding G^* is obtained by setting the cost of the arc on G_k^* corresponding to the vehicle under consideration to infinity. The type of decision tree search is depth-first search and the branching rule is to branch from the first available node. This rule is preferred due to its relatively less memory space requirement.

Initially the upper bound $\bar{z}$ is set equal to infinity and the lower bound z_0 is determined by solving the problem with infinite vehicle availability for all capacities.

The fathoming at any node can be due to at least one of the following: (i) A feasible solution is obtained. (ii) The lower bound obtained is not less than the current upper bound. (iii) The transformed network resulting is not connected.

The procedure stops when either all possibilities are exhausted or when none of the live nodes has a lower bound less than the current upper bound.

Let us solve the example problem of Section 4 using the vehicle availabilities given in Table 4.

Solving the initial problem on G_1^* obtained from the first Euler tour without any upper bounds imposed on the vehicle availability results in vehicle requirements (1, 2, 3) respectively. Thus, it is an infeasible solution. For the three vehicles required of type 3 causing infeasibility, we determine the arcs corresponding to their vehicle tours. They are arcs (3, 8), (11, 16), and (16, 21). We branch from the initial node into three nodes one for each vehicle and at each one of these nodes we set $\bar{c}_{38}$, $\bar{c}_{11,16}$, and $\bar{c}_{16,21}$ equal to infinity respectively (Figure 7). We continue the process from the node with $\bar{c}_{38} = \infty$ the selection being arbitrary.

During the course of the solution 43 nodes are investigated. 10 nodes are fathomed since they resulted in disconnected networks and 19 nodes are fathomed since their lower bound is not less than the current upper bound. At 2 nodes feasible solutions are obtained. The optimal solution for the problem on G^* is 1627 with the vehicle tours covering 31 arcs as shown in Table 5. The demand of arc (1, 8) is eliminated from vehicle tour 5 to reduce its capacity requirement from 100 to 80 and it is met by enlarging the vehicle tour {1, 9, 1}

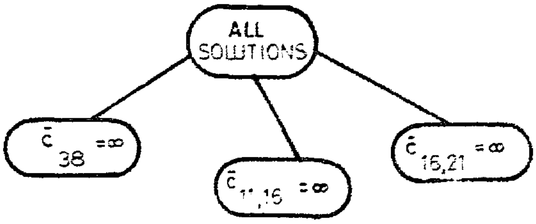

Figure 7. The first branching in the example problem

Table 5
The vehicle tours resulting from the first Euler tour with constrained vehicle availability

Vehicle tour	Vehicle capacity	Total demand	Vehicle tour cost
1, 5, 6, 1	80	60	76 + 80
1, 6, 4, 3, 4, 5, 9, 1	100	94	237 + 100
1, 9, 8, 1	50	41	102 + 50
1, 8, 10, 7, 2, 3, 6, 1	80	80	251 + 80
1, 9, 1, 8, 1	80	80	114 + 80
1, 6, 3, 6, 2, 10, 6, 1	100	96	257 + 100

which seems to be a persistent tour in the unconstrained case. All the other tours remain the same.

7. Summary and conclusions

In this paper, an algorithm for the fleet size and mix problem in capacitated arc routing has been presented for both undirected and directed networks. It has been extended to the case of limited vehicle availability which is solved for a branch and bound procedure.

The solution times for 10 undirected problems solved on a UNIVAC 1106 at Boğaziçi University are reported in Table 6. No bounds on vehicle availabilities are imposed in these problems. In all of them the giant tours are obtained by solving a CPP. The code for solving CPP is due to Burkard and Derigs (1980). It appears that the computation times obtained are reasonable and allow for extensive experimentation when the procedure is applied to a real life problem with additional considerations not included in the problem formulation.

Table 6
Computation times for some problems

Number	Number of nodes	Number of arcs	Comp. time (CPU sec)
1	10	18	1.389
2	12	24	1.635
3	13	28	1.820
4	15	36	2.960
5	18	54	3.521
6	20	64	5.632
7	24	86	7.958
8	26	96	8.941
9	28	114	8.199
10	30	128	11.308

References

Bodin, L., and Golden, B. (1981), "Classification in vehicle routing and scheduling", *Networks* 11, 97–108.

Burkard, R., and Derigs, U. (1980), *Assignment and Matching Problems: Solution Methods with FORTRAN Programs*, Lecture Notes in Economics and Mathematical Systems, No. 184, Springer, Berlin.

Chapleau, L., Ferland, J., Lapalme, G. and Rousseau, J.M. (1982), "A parallel insert method for the capacitated arc routing problem", Centre de Recherche sur les Transport, Publication No. 234, University of Montreal, Montreal, Canada.

Christofides, N. (1973), "The optimum traversal of a graph", *OMEGA* 1, 719–732.

Dijksta, E.W. (1959), "A note on two problems in connection with graphs", *Numerische Mathematik* 1, 269–271.

Edmonds, J., and Johnson, E.L. (1973), "Matching, Euler tours and the Chinese postman problem", *Mathematical Programming* 5, 88–124.

Frederickson, G.N., Hecht, M.S., and Kim, C.E. (1978), "Approximation algorithms for some routing problems", *SIAM Journal on Computing* 7, 178–193.

Golden, B., and Wong, R.T. (1981), "Capacitated arc routing problems, *Networks* 11, 305–315.

Kwan, Mei-Ko (1962), "Graphic programming using odd or even points", *Chinese Mathematics* 1, 273–277.

Lenstra, J., and Rinnooy Kan, A. (1981), "Complexity of vehicle routing and scheduling problems, *Networks* 11, 221–227.

Minieka, E. (1979), "The Chinese postman problem for mixed networks", *Management Science* 25, 643–648.

Orloff, C. (1974), "A fundamental problem in vehicle routing", *Networks* 4, 35–64.

Rosenkrantz, D., Stearns R., and Lewis, P. (1974), "Approximate algorithms for the travelling salesman problem", *Proceedings of the 15th Annual IEEE Symposium Switching and Automata Theory*, 33–42.

Ulusoy, G. (1981), "Vehicle routing problems and two algorithms", Report No. 8112, Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey.