Route First—Cluster Second Methods for Vehicle Routing

JE BEASLEY

Imperial College of Science and Technology, London, UK

(Received November 1982; in revised form January 1983)

In this paper we consider route first—cluster second methods for the vehicle routing problem. Extensions to the basic method both to improve its effectiveness and to enable it to cope with practical constraints are described. Computational results are given for the method applied to standard vehicle routing problems drawn from the literature.

INTRODUCTION

The vehicle routing problem can be defined as the problem of designing routes for delivery vehicles of known capacities, operating from a single depot, to supply a set of customers with known locations and known demand for a certain commodity. Routes for the vehicles are designed to minimise some objective such as the total distance travelled.

Recent surveys [5,17,20] list many approaches (both heuristic and optimal) to the problem. In this paper we evaluate one approach to the problem based upon a route first—cluster second heuristic. A similar approach has been successfully applied to bus routing problems [3,18], the routing of electric meter readers [19], the routing of street sweepers [2,4] and vehicle fleet size and mix problems [16]. However, as far as we are aware, no evaluation of the approach on standard vehicle routing problems (which would enable it to be compared with other methods) has been published. This paper attempts to remedy this.

We also give a number of extensions to the basic method that illustrate that it can be easily adapted to deal with many of the practical constraints encountered in vehicle routing. We first describe the basic method.

BASIC METHOD

The basic route first—cluster second method is best illustrated by a diagram. Consider Fig. 1 where we have a central depot surrounded by a number of customers. We first form a 'giant tour' from the depot around all the customers and back to the depot (i.e. a travelling salesman tour around all the customers including the depot). This tour can be formed in a number of different ways, as discussed later.

Fig. 1. Giant tour.

The key to the approach is that it is very easy to optimally partition such a tour into a set of feasible vehicle routes. Arbitrarily assign a direction to the giant tour and (without loss of generality) let 1 be the first customer on the directed tour after the depot (which we denote
by 0), 2 be the second customer on the tour after the depot, \ldots, n be the last customer on the tour after the depot. Let \((d_{ij})\) be the inter-
customer distance matrix and define a matrix
\(c_{ij}\) by

\[
c_{ij} = \begin{cases} 
\text{the distance travelled by a vehicle in} \\
\text{supplying the customers } (i+1, \\
i+2, \ldots, j) \text{ in that order if the vehicle} \\
\text{route } (0, i+1, i+2, \ldots, j, 0) \text{ is feasible} \\
(i < j) \\
\infty \text{ otherwise;}
\end{cases}
\]

i.e.

\[
c_{ij} = d_{0(i+1)} + \sum_{k=i+1}^{j-1} d_{i(k+1)} + d_{j0} \quad \text{or} \quad c_{ij} = \infty.
\]

Note that we assume here that all vehicles are identical. If we then find the least cost path from
0 to \(n\) in the directed graph with arc costs \((c_{ij})\) we will have an optimal partition of the (di-
rected) giant tour into feasible vehicle routes. (Note that if no path from 0 to \(n\) exists then the
problem is infeasible.)

For example, suppose that the least cost path
from 0 to \(n\) is \(0-s-t-n\) of total cost \(c_0 + c_s + c_n\) then from the way that \(c_{ij}\) is defined we must
have \(s < t < n\). The first part 0–s of this least cost path involves a vehicle supplying customers
1, 2, \ldots, s in that order (from the definition of
\(c_0\)). The second part s–t of this least cost path
involves a vehicle supplying customers \(s+1, \\
s+2, \ldots, t\) in that order (from the definition of
\(c_s\)). The final part \(t–n\) of this least cost path
involves a vehicle supplying customers \(t+1, \\
t+2, \ldots, n\) (from the definition of \(c_n\)).

We know that each of these three vehicle routes is feasible (from the definition of the \((c_{ij})\)) and
together they supply all the customers. Hence we have a solution to the vehicle routing
problem. Note that this partition of the giant tour into three feasible vehicle routes is optimal
since we found the least cost path from 0 to \(n\) in the directed graph with arc costs \((c_{ij})\). (Any
path from 0 to \(n\) corresponds to a partition of the giant tour into feasible vehicle routes and
the least cost path from 0 to \(n\) corresponds to an optimal partition.)

Note that, in general, if the least cost path
from 0 to \(n\) involves \(m\) arcs then \(m\) vehicles are
used.

**OVERVIEW**

We can see from the above description why
the method is called route first—cluster second. We first decide the order in which the customers
are to be visited (the routing part of the process) and then partition the customers (cluster the
customers) into sets that constitute feasible vehicle routes.

On paper this route first—cluster second heu-
ristic appears to be attractive for a number of
reasons:

1. The use of a giant tour ensures that custom-
ers who are near to each other are close
together on the giant tour and hence likely to
be together on the vehicle routes consid-
ered in the formation of the matrix \((c_{ij})\).

2. Via the partitioning approach we are able
implicitly to consider a large number of
feasible vehicle routes and from them pick
an optimal set of routes.

3. The partitioning of the giant tour is rela-
tively fast computationally (e.g. using the
algorithm of Dijkstra [13] involves only
\(O(n^2)\) operations).

4. Because the partitioning procedure is fast
(and the other parts of the method are also
not particularly time consuming) one can start
from a number of different giant tours
and produce a feasible set of vehicle routes
from each tour. This overcomes the problem
that any single giant tour might lead to a bad
set of vehicle routes.

Note here that since it is easily shown that an
optimal travelling salesman (giant) tour fol-
lowed by an optimal partitioning does not nec-
essarily lead to an optimal set of vehicle routes,
one would expect that a heuristic, rather than
optimal, approach to the formation of the giant
tour would be sufficient (e.g. an initial random
tour followed by a 2–optimal [7] procedure or
see [9] for other heuristic approaches to the
travelling salesman problem). Levy et al. [16]
make a number of similar points in their dis-
cussion of the use of the approach for vehicle
fleet size and mix problems.

Finally we note that the fact that other au-
thors have reported success with the approach
applied to problems similar to the vehicle rout-
ing problem [2, 3, 4, 16, 18, 19] would also lead
one to suppose that it would be an effective
method for the vehicle routing problem.
EXTENSIONS

There are a number of extensions that we can make to the basic method described above both to improve its effectiveness and to enable it to cope with the practical constraints associated with vehicle routing problems. We first extend the definition of the \( c_{ij} \) to be

\[
c_{ij} = \begin{cases} 
\text{the total cost of supplying the customers} & \text{for each } c_{ij} \text{ as a simple task to check whether the route is feasible with respect to} \\
\text{of supplying them is feasible} & \text{the time windows (see Christofides \cite{8}). If we} \\
(i < j) & \text{are interested in reordering the customers} \\
\infty & \text{then we have a travelling salesman problem} \\
& \text{with additional constraints which can be} \\
\end{cases} \\
tackled heuristically (e.g. by a modified 2-optimal procedure) or optimally (see \cite{10, 11}).
\]

(5) If we have different types of vehicle (and a constraint on the number of each type) then there may well be more than one type of vehicle that can supply a set of customers—in this case we must expand \( c_{ij} \) to have a third subscript dealing with the type of vehicle used (the definition being the obvious one) and the least cost path problem now becomes a constrained least cost path problem—the constraint on the least cost path relating to the number of vehicles of each type that can be used. Large problems of this type can be solved optimally relatively quickly (see \cite{1}).

(6) Customers who object to certain types of vehicles can also be incorporated into the method (as in (5) above).

(7) The multi-depot problem (where the customers are to be supplied from one of a number of depots) can also be incorporated into the method by associating different vehicle types with each depot (as in (5) above).

Note that many of the extensions discussed above relate to the use of the method for dealing with problems involving practical constraints (such as customer time windows, limited numbers of vehicles of different types etc.).

COMPUTATIONAL RESULTS

As mentioned previously, although the route first—cluster second approach has been fairly widely discussed and used in the literature we know of no evaluation of the method on standard vehicle routing problems (such as the problems of Eilon et al. \cite{14}). Accordingly we programmed the method and solved some test problems from the literature. The details of our implementation of the method were as follows:

(1) We generated an initial giant tour (excluding the depot) randomly and then used a
TABLE 1. COMPUTATIONAL RESULTS

<table>
<thead>
<tr>
<th>Problem number</th>
<th>Number of customers</th>
<th>Savings solution(^1)</th>
<th>2-optimal solution(^2)</th>
<th>Route first—cluster solution 1 5</th>
<th>Route second solution 10 25</th>
<th>Best solution at iteration</th>
<th>Time per iteration CDC 7600 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>119/2</td>
<td>114/2</td>
<td>114/2</td>
<td>114/2</td>
<td>114/2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>290/4</td>
<td>290/4</td>
<td>290/4</td>
<td>290/4</td>
<td>290/4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>598/4</td>
<td>585/4</td>
<td>608/4</td>
<td>585/4</td>
<td>585/4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>955/5</td>
<td>949/5</td>
<td>1017/6</td>
<td>994/6</td>
<td>968/5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>29</td>
<td>963/5</td>
<td>875/4</td>
<td>879/4</td>
<td>876/4</td>
<td>875/4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>1427/8</td>
<td>1414/8</td>
<td>1524/9</td>
<td>1462/8</td>
<td>1444/8</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>839/5</td>
<td>810/4</td>
<td>848/5</td>
<td>815/5</td>
<td>814/5</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>585/6</td>
<td>556/5</td>
<td>564/5</td>
<td>564/5</td>
<td>565/5</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>75</td>
<td>900/10</td>
<td>876/10</td>
<td>906/11</td>
<td>895/11</td>
<td>884/11</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>887/8</td>
<td>863/8</td>
<td>902/8</td>
<td>880/8</td>
<td>878/8</td>
<td>20</td>
</tr>
</tbody>
</table>

\(^1\)Solution value given as distance travelled/vehicles used, distance travelled is sum of true route distances rounded to nearest integer.

\(^2\)Best of three trials (problems 3, 5 best of ten trials).

2-optimal interchange procedure to improve the tour until no further improvements could be made.

(2) The matrix \((c_{ij})\) was then calculated with the 2-optimal interchange procedure being used to reorder the customers \(i + 1, i + 2, \ldots, j\) for the purposes of calculating a value for \(c_{ij}\). We also added a large positive constant to each \(c_{ij}\) so that the set of routes produced by partitioning the giant tour involved as few vehicles as possible.

(3) We used Floyd's [15] algorithm to calculate the least cost paths in the directed graph with cost matrix \((c_{ij})\) and thereby obtained an optimal partitioning of the giant tour.

(4) The routes in the optimal partition of the giant tour were individually 3-optimised [7]—with a small number of customers on the route this means we are almost sure of having an optimal travelling salesman tour of the customers.

We programmed the algorithm in FORTRAN and ran it on a CDC 7600 using the FTN compiler with maximum optimisation. Table 1 gives details of the problems solved, all of which were taken from Eilon et al. [14]. For each problem we generated 25 giant tours and the table shows the best result obtained after 1/5/10/25 giant tours had been generated and partitioned. We also give in that table the number of giant tours (iterations) needed to produce the best result. The computation time given is the average time to generate and partition one giant tour in CDC 7600 sec. Note here that the computationally most expensive part of our implementation of the route first—cluster second method is the use of Floyd's [15] algorithm to calculate the least cost paths in the directed graph with cost matrix \((c_{ij})\) which enable us to optimally partition the giant tour. This implies that computation times are proportional to the cube of problem size (number of customers).

The author is grateful to one of the referees for pointing out that for problem 6 the original Clarke and Wright [12] data for the problem involves an out and back trip for a fully loaded vehicle for customer number 19. This fact is not made clear in the data given for that problem by Eilon et al. [14].

Also in Table 1 we give the savings [12] and 3-optimal [6] solution values taken from Eilon et al. [14].

DISCUSSION

Examining the results obtained after 25 giant tours had been generated and partitioned, we see that in five of the ten problems the route first—cluster second approach gives a result at least as good as the 3-optimal solution, in two of the ten problems it gives a result between the savings and 3-optimal values and in three of the ten problems it gives a result worse than the savings solution.

These results are fairly encouraging—the method appears to give results that are, on balance, at least as good as the savings method and often as good as the 3-optimal method for a reasonable computation time per iteration.
In the four problems with route distance constraints (problems 3, 4, 5 and 7) the method does not appear to be less effective than in problems with no route distance constraints and certainly the efficiency of the method is not adversely affected by the presence of such constraints.

Since the number of vehicles used is often a more important criterion of solution quality the distance travelled it is interesting to consider the results from this viewpoint. Overall the total number of vehicles used for the route first—cluster second method is 58 after only one iteration, 57 after five iterations, 56 after ten iterations and 55 after 25 iterations. This compares with 57 for the savings method and 54 for the 3-optimal method.

An interesting question is whether a better (lower cost) giant tour leads to a better partitioning and hence to a better set of vehicle routes. To get some insight into this question we compare in Table 2 the solution for the best giant tour (giant tour length and corresponding routing solution) and the solution for the best vehicle routes (corresponding giant tour length and routing solution).

The results of this comparison are very interesting. For most problems the routes for the best giant tour are close (if not equal) to the best vehicle routes. In a few problems there are significant differences (problems 4, 6 and 7) and these seem to be associated with problems where a slightly longer giant tour can be partitioned into fewer vehicle routes.

Table 2 would seem to indicate that fewer iterations than we have used, with more computational effort put into the construction of the giant tour, would lead to better quality results.

## CONCLUSIONS

In this paper we have examined the route first—cluster second approach to vehicle routing and have discussed the basic method and the extensions to the method that are possible to deal with practical constraints. Computational results for one implementation of the method on standard vehicle routing problems drawn from the literature have been given. We feel that these results, together with the extensions to the method that are possible, lead one to conclude that the basic method has promise as a foundation for an effective procedure for practical vehicle routing problems.

## ACKNOWLEDGEMENT

The author is grateful to both referees for their helpful comments and suggestions.

## REFERENCES


**ADDRESS FOR CORRESPONDENCE:** Dr JE Beasley, Department of Management Science, Imperial College of Science and Technology, Exhibition Road, London SW7 2BX, UK.

**APPENDIX**

In this appendix we give, for each of the problems in Table 1, the routes corresponding to the best solution found. Further details of the problems are in Eilon et al. [14]. In all cases the depot is denoted by 0.

**Problem 6**
Routes: 0—19—0 (see note about problem 6 in main text)
0—2—1—20—12—17—0
0—21—30—0
0—18—8—25—0
0—19—10—26—0
0—3—4—6—5—11—16—15—27—23—0
0—22—28—24—0
0—29—13—7—9—14—0

**Problem 7**
Routes: 0—14—13—10—9—8—32—11—12—2—1—0
0—6—7—5—4—3—30—31—0
0—29—28—16—27—26—0
0—18—19—21—20—22—23—24—25—17
15—0

**Problem 8**
Routes: 0—5—49—10—45—33—39—30—34—50—9
0
0—12—17—37—15—44—42—19—40—41—4
47—0
0—18—13—25—14—24—43—6—0
0—27—48—23—7—26—8—31—28—22—1
32—46—0
0—11—2—3—36—35—20—29—21—16—38
15—0

**Problem 9**
Routes: 0—45—29—27—13—54—52—34—0
0—46—8—19—59—14—35—7—0
0—53—11—66—65—38—0
0—58—10—31—55—25—9—39—72—0
0—30—18—24—49—16—33—0
0—63—23—56—41—64—42—43—1—73—0
0—6—22—62—2—68—75—0
0—51—3—44—32—40—12—17—0
0—26—67—4—0
0—30—48—21—69—61—28—74—0
0—5—47—36—71—60—70—20—37—15—57
15—0

**Problem 10**
Routes: 0—2—57—42—100—85—91—44—38—14
0—43—15—41—22—73—21—40—0
0—89—18—60—83—8—45—17—86—16—61
0—84—5—99—96—6—0
0—94—95—59—93—98—37—92—97—87
13—58—0
0—53—28—26—12—76—50—1—69—27—0
0—52—7—82—48—47—46—36—49—64—11
19—62—88—0
0—31—10—63—90—32—66—65—71—20
30—70—0
0—77—3—79—33—81—51—9—35—34—78
29—24—80—68—0
0—54—4—55—25—39—67—23—56—75—74
72—0