A parallel insertion heuristic for vehicle routing with side constraints

M.W.P. Savelsbergh
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
and
Erasmus University Rotterdam
P.O. Box 1738
NL- 3000 DR Rotterdam
The Netherlands

In the early eighties, Fisher and Jaikumar developed a generalized assignment heuristic for vehicle routing problems. In this paper, we discuss some of the strong and weak points of this heuristic, and take its basic ideas to develop a new parallel insertion heuristic for the vehicle routing and scheduling problem that is better able to handle various side constraints.

Key Words & Phrases: generalized assignment, time windows, circle covering.

1. Introduction

In the early eighties, FISHER and JAIKUMAR (1981) developed a generalized assignment heuristic for the standard vehicle routing problem. It is a 'cluster first-route second' approach, which produces excellent results for most of the standard test problems. A number of papers have since been written that extend the ideas of this method, for instance by NYGARD et al. (1988) and SAVELSBERGH (1988).

One of the nice features of the generalized assignment heuristic, as originally proposed, is that decisions regarding the assignment of customers to vehicles and decisions regarding the determination of delivery sequences are separated. However, this also turns out to be one of its main drawbacks when additional constraints are incorporated in the model. Time window constraints at customers are a prime example. In that case, it is usually unwise to postpone all sequencing considerations until the routing phase.

This paper will describe a parallel insertion heuristic for the vehicle routing problem with time windows and mixed collections and deliveries that is based on the underlying ideas of the generalized assignment heuristic, but relaxes the separation of the assignment and sequencing decisions.

We would also like to point out that the selection of a good set of seed points is of crucial importance for the performance of the generalized assignment heuristic. This is often neglected, at least in the published literature on the

generalized assignment heuristic. Therefore, we will also present, in some detail, the circle covering method that we have designed to obtain seed points.
It is only for presentational convenience that we restrict ourselves to the vehicle routing problem with time windows and mixed collection and deliveries. It will become clear to the reader that the method can be easily extended to incorporate other side constraints. In addition to the general description of the method, we will also address some implementation issues.

2. The vehicle routing problem with time windows and mixed collection and deliveries

In the vehicle routing problem with time windows and mixed collections and deliveries, a number of vehicles is located at a single depot and must serve a number of geographically dispersed customers. Each vehicle has a given capacity. Each customer has a given demand, to be either collected or delivered, and must be served within a specified time window. All deliveries have to be collected at the depot and all collections have to be delivered at the depot. The objective is to minimize the total cost of travel.
Let $N=\{1,...,n\}$ be the set of customers and let $M=\{1,...,m\}$ be the set of vehicles. We will denote the time window of a customer $i \in N$ by $[e_i,l_i]$, the arrival time at i by A_i, and the departure time at i by D_i. It is assumed that service time at i is included in the travel time t_{ij} from customer i to customer j. Since service must start within the time windows, we require that $e_i \leqslant D_i \leqslant l_i$ for each $i \in N$. If $A_i < e_i$, then a waiting time $W_i = e_i - A_i$ occurs before the opening of the window at i. The demand of a customer $i \in N$ will be denoted by q_i, the set of customers where a vehicle has to make a collection by Γ, the set of customers where a vehicle has to make a delivery by Δ, and the vehicle capacity by Q.

3. The generalized assignment heuristic

The generalized assignment heuristic is usually presented as a two-phase method. In the first phase, an assignment of customers to vehicles is obtained by solving a generalized assignment problem with an objective function that linearizes and thereby approximates the cost of the traveling salesman tours of the vehicles through the customers. In the second phase, once the assignment decision has been made, a routing of each vehicle through its set of customers is obtained by solving a traveling salesman problem. The approximation of the delivery cost is obtained by constructing seed routes and considering the cost of inserting customers into these seed routes. A seed route is an artificial route consisting of the depot and a seed point, which is supposed to indicate an area that is expected to be visited by a single vehicle.
As mentioned above, the choice of a good set of seed points is of crucial importance for the performance of the method. In our opinion, it is therefore better to consider the generalized assignment heuristic as a three-phase method, to emphasize the importance of seed selection. In this perspective, a

set of seed points is chosen in the first phase, an assignment of customers to seed points is determined in the second phase, and routes are constructed for each of the obtained clusters in the third phase.

4. A PARALLEL INSERTION HEURISTIC

The parallel insertion heuristic described in this paper, is an example of a construction procedure: it tries to build a feasible solution starting from the raw data. The adjective 'parallel' indicates that the routes are not constructed one after the other, but simultaneously. To improve the quality of the resulting solution, one may apply iterative improvement methods afterwards. An extensive treatment of iterative improvement methods for vehicle routing problems with side constraints is given by SAVELSBERGH (1986, 1989).
In developing construction procedures for vehicle routing and scheduling problems based on insertion, the following three key questions serve as guidelines:

(a) How is the set of m initial routes, each visiting only a single customer, chosen?

(b) Which customer is selected next to be inserted into the current solution?

(c) Where will it be inserted?

The purpose of the set of initial routes is similar to the purpose of the set of seed points in the generalize assignment heuristic. However, in the generalized assignment heuristic all assignment decisions preceed all routing decisions, whereas in a construction procedure assignment and routing decisions are taken simultaneously.
The set of initial routes is chosen on the basis of customer demand and customer proximity, the choice of the customer to be inserted next, is derived from a heuristic for the generalized assignment problem proposed by MARTELLO and TOTH (1981), and the choice of the place of the place where to insert this customer is based on a heuristic for the traveling salesman problem with various side constraints proposed by SAVELSBERGH (1988).

4.1. Initial routes

In this subsection, we will describe a method to determine a good set of seed points, i.e., initial routes. As seed points are supposed to indicate areas that are expected to be visited by a single vehicle, it is clear that customer demand will play an important role in the determination of seed points. The method originally proposed by Fisher and Jaikumar divides the area around the depot in cones, each representing a total demand that is close to vehicle capacity, and locates the seed points on the rays that bisect the cones at such a distance from the depot that a fixed percentage of the total demand in the cone is closer to the depot. Note that it is not obvious how to extend this method to problems where vehicles do not have identical capacities.
The method we propose is not only based on customer demand but also on 'customer proximity'. This is a very important aspect in practical distribution

problems, for various reasons. Consider the two possible solutions for the same situation depicted in Figure 1.

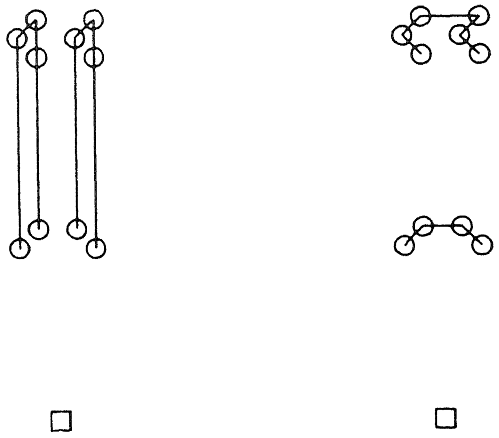

Figure 1. Difference between the cone covering and circle covering method.

The first solution might result if the *cone covering method* as proposed by Fisher and Jaikumar is applied. The second solution is the one that is usually preferred in practice.Note that this is not an artificially constructed example, but a frequently occurring phenomenon in practice.

Instead of the cone covering method we propose what might be called the *circle covering method.* The basic idea is the following. For each customer, determine the smallest circle with the center at this customer such that the total demand inside the circle is close to vehicle capacity. Next, order the customers by increasing radius of their associated circles. Finally, cover all customers by circles as follows. Iteratively take the first not yet covered customer on the list and add its associated circle to the covering until all customers are covered. Take the set of seed points equal to the set of centers of the circles that constitute the covering. Note that this approach is independent of the location of the depot, offers better opportunities to incorporate different vehicle capacities and possibly even opportunities to incorporate time related information.

Since we intend to incorporate these methods in the PC-version of CAR, an interactive planning system for Computer Aided Routing (SAVELSBERGH, 1988), internal memory is a scarce resource. In such a situation, the number of disk accesses has to be kept as small as possible. Therefore, we propose the following simplification. Instead of working with all n^2 distances throughout, we construct a minimum spanning tree and work only with distances corresponding to edges in the tree. This has the advantage that we can keep all the distances required during the determination of the circles in internal memory. In addition, there is a small improvement in running time as well. If we work with a complete graph, the running time of the circle covering method is

$O(n^2\log n)$, since there are n customers and it takes $O(n\log n)$ time to determine the circle associated with a customer. If we work only with a spanning tree, the running time reduces to $O(n^2)$, since it now takes only $O(n)$ time to determine the circle associated with a customer.

To facilitate access of the minimum spanning tree, we orient its edges and add a number of additional links. First, an arbitrary leaf of the tree is chosen to be the root and all edges are oriented towards this root. Secondly, a 'leftmost child-right sibling' structüre (see for instance AHO, HOPCROFT, and ULLMAN, 1983) is added to the tree. Figure 2 shows a spanning tree and the associated easily accessible augmented tree. It takes $O(n)$ time to augment a given minimum spanning tree. As an example of the use of the augmented tree, we present the procedure that gives all neighbors of a node, i.e., the nodes in the tree at distance one. This procedure is heavily used when the circle associated with a node is determined.

```
procedure neighbours (node)
begin

      neighbor ← parent(node)
      neighbor ← leftmostchild(node)
      while (rightsibling(neighbor) exists) do
      begin
                  neighbor ← rightsibling(neighbor)
      end
end
```

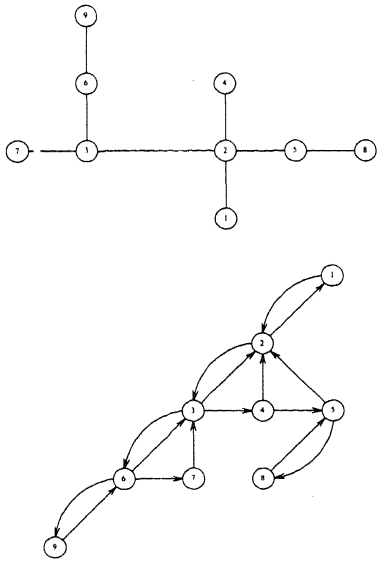

FIGURE 2. Spanning tree and its associated augmented tree.

4.2. Selection criterion

Current optimization algorithms for the generalized assignment problem are able to solve problems of upto 500 decision variables (GUIGNARD and ROSENWEIN, 1986). This prohibits the application of these algorithms in the context of the generalized assignment heuristic for vehicle routing, since practical routing problems are often much larger. Thus, we have to resort to approximation. MARTELLO and TOTH (1981) describe a simple but effective approximation algorithm for the generalized assignment problem. Assignments are made sequentially based on the following rule: always make the best feasible assignment for the column that has the largest difference between the best feasible assignment and the second best feasible assignment. We can interpret this as follows. An ideal objective value, initially equal to the sum of the costs of the best assignment for each column, is maintained, and the quality of an assignment is measured by the change in the ideal objective value if at a later stage this assignment would no longer be feasible. Thus it can be viewed as a simple look-ahead strategy, which tries to prevent bad assignments.

One of the drawbacks of keeping assignment decisions and routing decisions separated as is the case in the generalized assignment heuristic, is that the feasibility of an assignment is only determined on the basis of capacity considerations.

In the parallel insertion heuristic, the rule of Martello and Toth is applied in selecting the customer that is to be inserted into the current partial solution. However, the feasibility of an assignment is determined by the fact whether or not it is possible to actually insert a customer in a partial route. This means that we can incorporate all sorts of side constraints. In fact, the clustering phase and routing phase of the generalized assignment heuristic are merged. Instead of building clusters in parallel and creating routes for the resulting clusters, we are now constructing routes in parallel.

To implement the selection step efficiently, we maintain for each customer the best and the second best feasible assignment. Let us emphasize that best and second best refer to routes (in fact, different routes). Each time a customer is inserted into one of the routes, we must check all remaining free customers to see whether their best and second best feasible assignments are still correct. Let us analyze what may cause these values to become incorrect. There are two possible reasons: one of the assignments becomes infeasible, or a better feasible assignment becomes available. The first situation, which is a nasty one, occurs when the last insertion was in the route associated with the best or the second best assignment and causes an assignment to become infeasible. In that case, the best and second best feasible assignments have to be recalculated from scratch. However, as soon as it becomes impossible to assign a customer to a route, the route can be discarded for this customer for the remainder. The second situation, which can easily be handled, only arises at two specific places: immediately preceding or immediately succeeding the last inserted customer.

Note that it is possible that at some point a customer has no feasible assignment left. At that point the customer is marked as being *unvisited* and will not be included in the final set of routes. We envision that in future implementations iterative improvement methods will be applied to the partial solution at specific points in time to reduce the chance of ending up with unvisited customers.

4.3. *Insertion criterion*

As explained in Section 4.2, a customer is selected to be inserted on the basis of the difference between the cost associated with the best and second best feasible assignments. At this point, we clearly see the advantage of the parallel insertion heuristic over the generalized assignment heuristic: we are more accurate in evaluating the feasibility as well as in estimating the cost of an assignment. Especially with respect to the feasibility we are no longer restricted to simple capacity considerations, but can incorporate various other side constraints.

At this point, it is appropriate to analyze, in some detail, the insertion of a yet unrouted customer into a route. The analysis is in two parts: the first deals with the *feasibility* of an insertion, the second deals with its *profitability*. The presentation follows SAVELSBERGH (1988). For presentational convenience, we split the depot (vertex 0) in an 'origin' (vertex 0) and a 'destination' (vertex $n+1$), and, in the sequel, when we refer to a route, we assume that it is given by $(0,1,...,i\ ,...,n,n+1)$, where n is the number of customers in the route and i is the ith customer visited by the vehicle.

To establish the feasibility of an insertion, we have to test the side constraints. We will consider *time window* and *capacity* constraints. (Note that even if after the insertion both the total load to be delivered and the total load to be collected do not exceed the vehicle capacity, it is still possible that the ordering of the customers leads to a violation of the capacity constraints.)

Let u be the unrouted customer to be inserted, and i and $i+1$ the customers between which u is being inserted. The insertion of u between i and $i+1$ generally has two effects. First, it can affect all the arrival times at vertices $i+1$, $i+2$,..., $n+1$, which may result in an infeasible tour. Secondly, it affects either the vehicle load when visiting the vertices $0,1,...,i$, in case u is a delivery, or the vehicle load when visiting the vertices $i+1, i+2,...,n+1$, in case u is a collection.

To test the feasibility of an insertion with respect to the time window constraints efficiently, a *possible forward shift* S_i^+ is maintained which expresses the largest increase in the departure time D_i at i which causes no violation of the time windows along the path $(i,...,n+1)$:

$$S_i^+ := \min_{i \leqslant j \leqslant n+1}(l_j - (D_i + \Sigma_{i \leqslant k < j} t_{t_{k,k+1}})).$$

The feasibility test of an insertion then amounts to

$$\max(D_i + t_{i,u,}\ e_u) + t_{u,i+1} - D_{i+1} \leqslant S_{i+1}^+.$$

The following backward recursion will compute S_k^+ for all customers k in $O(n)$ time:

$$S_{n+1}^{+} \leftarrow l_{n+1} - D_{n+1};$$

$$S_k^{+} \leftarrow \min\{S_{k+1}^{+} + W_{k+1}, l_k - D_k\} \quad \text{for } k = n, \ldots, 1.$$

To test the feasibility of an insertion with respect to the capacity constraints efficiently, we introduce the following quantities. We define

$$C_k := Q - \Sigma_{j>k, j\in\Delta} q_j - \Sigma_{j\leqslant k, j\in\Gamma} q_j,$$

which is the *remaining capacity* in the vehicle at the departure at vertex k,

$$L_k^{-} := \min\nolimits_{j\leqslant k} C_j,$$

which is the *maximum delivery increase* the vehicle can accommodate on the path $(0,\ldots,k)$ and

$$L_k^{+} = \min\nolimits_{j\geqslant k} C_j,$$

which is the *maximum collection increase* the vehicle can accommodate on the path $(k,\ldots,n+1)$. The feasibility of an insertion can now be tested by

$$q_u \leqslant L_i^{-}$$

in case u is a a delivery and

$$q_u \leqslant L_{i+1}^{+}$$

in case u is a collection. The values of C_k, L_k^{-}, L_k^{+} can be computed for all customers k in $O(n)$ time as follows:

$$C_0 \leftarrow Q - \Sigma_{j\in\Delta} q_j,$$

$$C_k \leftarrow \left\{\begin{matrix} C_k - 1 - q_k & \text{if } k \in \Gamma \\ C_{k-1} + q_k & \text{if } k \in \Delta \end{matrix}\right\} \quad \text{for } k = 1, \ldots, n+1,$$

$$L_0^{-} \leftarrow C_0,$$

$$L_k^{-} \leftarrow \min\{C_k, L_{k-1}^{-}\} \quad \text{for } k = 1, \ldots, n+1,$$

$$L_{n+1}^{+} \leftarrow C_n,$$

$$L_k^{+} \leftarrow \min\{C_{k-1}, L_{k+1}^{+}\} \quad \text{for } k = n, \ldots, 0.$$

The profitability of an insertion is based on two observations. First, a wrong assignment for customers far away from the depot is probably more harmful than a wrong assignment for a customer close to the depot. Secondly, since we are minimizing total travel times, the additional travel time incurred should be minimal. Therefore we propose the following insertion criterion, where u is the customer to be inserted and i a customer in one of the routes:

$$2t_{0,u} + t_{i,i+1} - t_{i,u} - t_{u,i+1}$$

5. Computational results

We have compared the performance of the generalized assignment heuristic to the performance of the parallel insertion heuristic. Both methods use the circle covering method to create seed points and initial routes respectively. They have been applied to six different instances, all real-life problem situations obtained via Logion B.V., the consultancy firm that markets CAR. All instances have 100 addresses (both deliveries and collection) and are difficult due to time windows at customers and route duration restrictions. They differ in the location of the addresses and range from instances with a central depot and addresses located all over the Netherlands without easily identifiable clusters to instances with an outside depot and addresses located in a small densely populated area with several easily identifiable clusters. The results are given in Table 1.

As mentioned in the introduction, the main motivation for the development of the parallel insertion heuristic is the decrease in performance of the generalized assignment heuristic in the presence of various side constraints, such as time windows and mixed collections and deliveries. For none of the six instances the generalized assignment heuristic was able to construct a set with the specified number of routes visiting all addresses. As expected, the generalized assignment problems arising in the cluster phase were easily solved, but the construction of feasible routes for the resulting clusters failed due to the number and tightness of the time windows. The parallel insertion heuristic performed clearly better, in the sense that it succeeded in producing an initial feasible solution with the specified number of routes.

Another interesting characteristic of the parallel insertion heuristic, which was observed during the test runs, is that it seems less sensitive to a bad set of initial routes than the generalized assignment heuristic to a bad set of seed points, i.e., it is a more robust method.

As to computation times, the implementation of the generalized assignment heuristic was superior. It was about three to five times faster. Currently, the difference is partly caused by the fact that a lot of effort has been put in optimizing the code for the generalized assignment heuristic, whereas the code for the parallel insertion heuristic is still under development.

TABLE 1. Comparison of the performance of the generalized assignment heuristic and the parallel insertion heuristic.

		Generalized Assignment Heuristic			Parallel Insertion Heuristic		
problem instance	number of routes	number of unvisited addresses	total distance	total route duration	number of unvisited addresses	total distance	total route duration
1	4	6	1034	2804	0	966	2807
2	5	3	2453	5517	0	2251	5424
3	8	7	3124	8025	0	3534	8702
4	7	17	3514	6911	0	3767	7852
5	7	9	3481	7031	0	3521	7421
6	8	2	1294	5233	0	1001	5241

ACKNOWLEDGEMENT

The author would like to express his thanks to S. Martello and P. Toth for stimulating discussions on the integration of the clustering and routing phases in the generalized assignment heuristic.

REFERENCES

AHO, A.V. J.E. HOPCROFT and J.D. ULLMAN (1983), *Data Structures and Algorithms*, Addison-Wesley.

FISHER, M.L. and J. JAIKUMAR (1981), A generalized assignment heuristic for vehicle routing, *Networks 11*, 109-124.

GUIGNARD, M. and M.B. ROSENWEIN (1989), An improved dual-based algorithm for the generalized assignment problem, *Operations Research*, to appear.

MARTELLO, S. and P. TOTH (1981), An algorithm for the generalized assignment problem in: J.P. Brans (ed.). *Operations Research'81*, North-Holland, Amsterdam, 589-603.

NYGARD, K.E. P. GREENBERG, W.E. BOLKAN, and E.J. SWENSON (1988), Generalized assignment methods for the deadline vehicle routing problem in: B.L. Golden and A.A. Assad (eds.), *Vehicle routing: methods and studies*, North-Holland, Amsterdam, 107-126.

SAVELSBERGH M.W.P. (1986), Local search for routing problems with time windows, *Annuals Operations Research 4*, 285-305.

SAVELSBERGH, M.W.P. (1988), *Computer Aided Routing*, Doctoral thesis, Centre for Mathematics and Computer Science, Amsterdam.

SAVELSBERGH, M.W.P. (1989), An efficient implementation of local search for constrained routing problems. *European Journal of Operational Research*, to appear.

Received February 1989, Revised August 1989.