Theory and Methodology

The vehicle routing problem with backhauls

Marc GOETSCHALCKX and Charlotte JACOBS-BLECHA
Material Handling Research Center, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

Abstract: The Vehicle Routing Problem with Backhauls is a pickup/delivery problem where on each route all deliveries must be made before any pickups. A two-phased solution methodology is proposed. In the first phase, a high quality initial feasible solution is generated based on spacefilling curves. In the second phase, this solution is improved based on optimization of the subproblems identified in a mathematical model of the problem.

An extensive computational analysis of several initial solution algorithms is presented, which identifies the tradeoffs between solution quality and computational requirements. The class of greedy algorithms is capacity oriented, while K-median algorithms focus on distance. It is concluded that the greedy and K-median algorithms generate equivalent tour lengths, but that the greedy procedure reduces the required number of trucks and increases the truck utilization. The effect of exchange improvement procedures as well as optimal procedures on solution quality and run time is demonstrated. Comparisons with the Clark–Wright method adapted to backhauls are also given.

Keywords: Distribution, routing, heurstics, computational analysis, mathematical programming

1. Introduction

1.1. Problem statement

The classical Vehicle Routing Problem (VRP) involves a set of delivery points to be serviced by a set of vehicles housed at a central depot or Distribution Center (DC). The objective of the problem is to develop a set of routes such that all delivery points are serviced, the demands of the points assigned to each route do not violate the capacity of the vehicle which services the route, and the total distance traveled by all vehicles is minimized.

The Vehicle Routing Problem with Backhauls (VRPB), also known as the linehaul-backhaul problem, is an extension of the VRP involving both delivery and pickup points. Linehaul (delivery) points are sites which are to receive a quanitity of goods from the single central DC. Backhaul (pickup) points are sites which send a quantity of goods back to the DC. The crucial assumption is that all deliveries must be made on each route before any pickups can be made. This is caused by the fact that the vehicles are rear-loaded and rearrangement of the loads on the trucks at the delivery points is not deemed feasible. The quantities to be delivered and picked up are fixed and known in advance. There exists a homogeneous fleet of vehicles each of which is assumed to have a fixed capacity of some weight or volume. Hence, a feasible solution to the problem consists of a set of routes where all deliveries for each route are completed before any pickups are made and the vehicle capacity is not violated by either the linehaul or backhaul points assigned

Received December 1986; revised October 1988

to the route. The objective is to find such a set of routes which minimizes the total distance traveled.

1.2. Importance of the problem

The importance of the vehicle routing problem and all of its variations and extensions is based on the very large cost of physical distribution. In a report prepared for the National Council of Physical Distribution Management (NCPDM), Kearney (1984) estimates annual distribution costs in the United States in 1983 at $650 billion, approximately 21% of GNP. In addition, Kearney reports that logistics costs account for 22.5% of the controllable costs in manufacturing. Consequently, there has been an abundance of related literature since Dantzig and Ramser (1959) first introduced the classical VRP.

The importance of the linehaul-backhaul problem is relative to the continuing effort to reduce distribution costs by taking advantage of the unused capacity of an empty vehicle traveling back to the DC. Golden et al. (1985) illustrate the potential for such savings: the Interstate Commerce Commission estimated that $165 million was saved nationally in 1982 by grocery stores who took advantage of backhauling. Kearney's (1984) report to the NCPDM includes a summary of programs implemented by companies in the period from 1978–1983 for improving productivity in logistics. The number one program, utilized by 83% of the survey respondents, was coordination of inbound with outbound freight to provide private fleet backhauls. Thus, the study of the linehaul-backhaul problem can provide an important contribution to improved productivity in industry.

The linehaul-backhaul problem arises naturally in many distribution settings. A large company has many retail outlets (linehaul points) to be supplied from a central distribution center. At the same time, the distribution center must be resupplied by various vendors (backhaul points) located in the same region as the delivery sites. Specific examples include grocery store chains, retail department store chains, and manufacturing concerns. In addition, the government deregulation of interstate commerce restrictions is making it possible for backhauling to become a profitable venture for any company with a large fleet of vehicles. Commodities can now be backhauled not only for the owning company, but also for other companies who are willing to pay for the backhauls as though for common carriage.

1.3. Literature review

Many approaches to solving the classical VRP have been developed. These include not only optimization procedures which are applicable only to problems of very small scale, but also many heuristic procedures for problems of more realistic size. This section describes some of these algorithms for the VRP and some extensions to the linehaul-backhaul problem.

Exact procedures

A procedure is called exact when it finds the optimal solution to the problem. Exact procedures for solving the VRP include branch and bound, dynamic programming, and cutting plane algorithms. Christofides et al. (1981) discuss exact algorithms for the VRP. The VRP is an example of an NP-complete problem (Lenstra and Rinnooy Kan, 1981). Exact algorithms for NP-complete problems require a number of computations that grows exponentially in the size of the problem. Computationally, exact algorithms for the VRP are restricted to solving problems of only up to about 25 customers. In fact, the use of any of these procedures is not generally practiced since the time required to solve a realistically sized VRP to optimality is prohibitive.

The standard VRP is a special case of the linehaul-backhaul problem with the number of backhaul points equal to one (the distribution center). Since the VRP is NP-complete and a special case of VRPB, the VRP with backhauls is also NP-complete. This justifies heuristic approaches to VRP as well as the VRPB problem.

There is at least one exact procedure for a special case of the linehaul-backhaul problem developed by Yano et al. (1987). This procedure uses set covering to find an optimal set of routes. Each route can have at most four points. All such routes are generated and the optimal route sequencing is found by complete enumeration. Here, the solution procedure capitalizes on the size of the actual problem (an average of 15 delivery points and apparantly no more than 15 pickup points) and the real-world constraints to get exact solutions for this particular instance. However, it is doubtful if such an approach could be generalized to a practical procedure for larger, more general problems.

Heuristic procedures

Bodin and Golden (1981) classify heuristic strategies for vehicle routing problems in the following manner:

(1) cluster-first/route-second,
(2) route-first/cluster-second,
(3) savings/insertion,
(4) improvement/exchange,
(5) mathematical programming based,
(6) interactive optimization.

The last two categories are relatively new and might be combined into the single category of optimization-based techniques.

The cluster-first/route-second strategy is illustrated by the sweep algorithm of Gillett and Miller (1974). This algorithm aims at forming radial sectors of demand points by having a ray emanating from the DC 'sweep' around the region, clustering points so that vehicle capacity is not violated. When a set of clusters has been formed, the points in each cluster are sequenced. The sweep approach can be easily extended to the VRPB where clusters are truncated when either the linehaul or backhaul capacity is exceeded.

The route-first/cluster-second method works by solving a traveling salesman problem (TSP) over all the demand points to form one large route. This route is then broken up into smaller routes to satisfy the capacity constraints for each vehicle. Applications of this approach are given by Newton and Thomas (1969) and Bodin and Berman (1979) for the routing of school buses. Extension of the route-first/cluster-second approach to the VRPB does not seem feasible because of violation of the precedence constraints.

The concept of savings/insertion is a constructive approach whereby a configuration of points is changed to an alternative configuration which yields the largest 'savings' in terms of a particular objective. Perhaps the most widely known and used savings algorithm was developed by Clarke and Wright (1964). Deif and Bodin (1984) have proposed an extension of this algorithm for VRPB. Their procedure is based on two modifications to the Clarke–Wright algorithm. The first modification adds the constraint that only one link from linehaul to backhaul (or vice versa) is allowed on any route. Second, the definition of savings is modified to include a penalty to reduce the size of savings for a changeover from linehaul to backhaul and thus delaying the connection. Further details are given in Section 4.5.

Golden et al. (1985) report on an insertion procedure for VRPB where any VRP algorithm is used to initially sequence the delivery customers. Once the linehaul customers are routed, the backhaul customers are then inserted onto routes according to an insertion criterion. This procedure relaxes the constraint of only one link from linehaul to backhaul by allowing the pickup points to be inserted anywhere on the route. A penalty factor is used in computing the insertion cost for each backhaul point. This procedure could be used to enforce a single link from linehaul to backhaul by allowing the penalty factor to become very large.

The improvement/exchange techniques take an existing solution and attempt to improve it, usually by an exchange of links. Perhaps the best known method is the *r*-opt algorithm of Lin and Kernighan (1973). Christofides and Eilon (1972) developed a VRP algorithm by transforming the problem into a traveling salesman problem and applying 2-opt and 3-opt.

Optimization-based techniques involve developing a solution algorithm based on a mathematical formulation of the problem. The design of an optimization-based heuristic includes developing a mathematical model of the problem, observing specialized structure in the model, employing an optimization strategy such as decomposition to exploit the structure, and perhaps enhancing the solution process by allowing input from the user.

Magnanti (1981) discusses three basic formulations of the classical VRP and their relationships to specialized problem decomposition that exploits embedded structure. In addition, a discussion of optimization-based heuristics is given. He ascribes the two most successful approaches to VRP to Fisher and Jaikumar (1978, 1981), Cullen (1984), and Cullen et al. (1981). Both of these optimzation-based techniques have proven to be very successful in solving a wide range of test problems with as many as 200 demand points. In addition, the execution times are competitive with the more standard heuristics that ignore optimization procedures, for example the Clarke–Wright method, and yet provide better cost solutions in almost every instance.

1.4. Organization of the paper

This paper is organized as follows. In Section 2, the structure of the probem is discussed and a solution method based on a mathematical model of the problem is presented. In Section 3, algorithms are developed which very quickly generate an initial feasible solution based on spacefilling curves. Section 4 presents the results of a computational study of the algorithms. Finally, in Section 5, conclusions and a discussion on future research are given.

2. Problem structure

2.1. Mathematical model

In order to provide a more exact statement of the problem, notation for an integer programming formulation of the linehaul-backhaul problem is introduced below. The formulation presented here is modeled after Fisher and Jaikumar's (1978) formulation for the VRP, and the structure found here is similar to that observed in their model.

Constants

K	=	number of vehicles,
N	=	number of linehaul customers, indexed $1, 2, \ldots, N$,
M	=	number of backhaul customers, indexed $N+1,\ N+2, \ldots, N+M$ (index 0 indicates the distribution center),
a_i	=	demand of linehaul customer i, $i = 1, 2, \ldots, N$,
b_i	=	amount supplied by backhaul customer i, $i = N+1,\ N+2, \ldots, N+M$,
C	=	fixed capacity of the delivery vehicles,
c_{ij}	=	cost of direct travel from customer i to customer j, $i,\ j = 0,\ 1, \ldots, N,\ N+1, \ldots, N+M$.

Variables

$$u_{ik} = \begin{cases} 1 & \text{if linehaul customer } i \text{ is serviced by vehicle } k,\ i = 0, 1, \ldots, N, \\ 0 & \text{otherwise,} \end{cases}$$

$$v_{jk} = \begin{cases} 1 & \text{if backhaul customer } j \text{ is serviced by vehicle } k,\ j = N+1,\ N+2, \ldots, N+M \text{ and } j = 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels directly from customer } i \text{ to } j,\ i,\ j = 0,\ 1, \ldots, N,\ N+1, \ldots, N+M, \\ 0 & \text{otherwise.} \end{cases}$$

Formulation

$$\text{Minimize} \sum_{k=1}^{K} \sum_{i=0}^{N+M} \sum_{j=0}^{N+M} c_{ij} x_{ijk}$$

Subject to:

$$\sum_{k=1}^{N} a_i u_{ik} \leqslant C, \quad k = 1, 2, \ldots, K, \tag{1}$$

$$\sum_{k=1}^{K} u_{ik} = 1, \quad i = 1, 2, \ldots, n, \tag{2}$$

$$u_{0k} = 1, \quad k = 1, 2, \ldots, K,$$
$$u_{ik} = 0 \text{ or } 1, \quad i = 1, 2, \ldots, N,\ k = 1, 2, \ldots, K, \tag{3}$$

$$\sum_{i=N+1}^{N+M} b_i v_{ik} \leqslant C, \quad k = 1, 2, \ldots, K, \tag{4}$$

$$\sum_{k=1}^{K} v_{ik} = 1, \quad i = N+1,\ N+2, \ldots, N+M, \tag{5}$$

$$v_{0k} = 1, \quad k = 1, 2, \ldots, K,$$
$$v_{ik} = 0 \text{ or } 1,\ i = N+1,\ N+2, \ldots, N+M,\ k = 1, 2, \ldots, K, \tag{6}$$

$$\sum_{i=0}^{N+M} x_{ijk} = \begin{cases} u_{jk} & \text{if } j = 1, \ldots, N, \\ v_{jk} & \text{if } j = N+1,\ N+2, \ldots, N+M \text{ and } j = 0, \end{cases} \quad k = 1, 2, \ldots, K, \tag{7}$$

$$\sum_{j=0}^{N+M} x_{ijk} = \begin{cases} u_{ik} & \text{if } i = 0, 1, \ldots, N, \\ v_{ik} & \text{if } i = N+1,\ N+2, \ldots, N+M, \end{cases} \quad k = 1, 2, \ldots, K, \tag{8}$$

$$\sum_{i=0}^{N} \sum_{\substack{j=N+1 \\ \text{and } j=0}}^{N+M} x_{ijk} = 1, \quad k = 1, 2, \ldots, K, \tag{9}$$

$$x_{ijk} \in S, \tag{10}$$

$$x_{ijk} = 0 \text{ or } 1,$$
$$i, j = 0, 1, \ldots, N, N+1, \ldots, N+M, \quad k = 1, 2, \ldots, K, \tag{11}$$

where the set S can be expressed as

$$S = \left\{ \text{all } x_{ijk} \text{ such that } \sum_{i \in Q} \sum_{j \in Q} \leqslant |Q| - 1 \right\}$$

for every nonempty subset Q of $\{1, 2, \ldots, N, N+1, \ldots, N+M\}$.

Constraints (1) and (4) represent the condition that the trucks cannot be overloaded on either linehaul or backhaul, respectively, on any route. Constraints (2) and (5) indicate that only one vehicle can be assigned to each linehaul and backhaul route, respectively, and that no more than K vehicles can leave or return to the distribution center. Constraint set (7) says that exactly one vehicle must enter each customer site and the DC once, while (8) says that exactly one vehicle must leave each site once. Constraint (9) says that there must be exactly one link traveled by each vehicle from linehaul to backhaul on each route. Constraint (10) is the usual subtour elimination constraint for a traveling salesman problem.

2.2. Problem structure and decomposition methods

The model naturally decomposes into three subproblems, corresponding to the clustering decisions for the linehaul and the backhaul, and K independent, single route sequencing problems with one precendence constraint, respectively.

The feasibility of a solution for the delivery points with respect to the vehicle capacity constraints depends only on variables u_{ik}. These variables appear only in constraints (1), (2), and (3), the constraint set of a Generalized Assignment Problem (GAP) (see, e.g., Ross and Soland, 1975). Solving the GAP yields the clustering decision for the linehaul points. Similarly, the variables v_{jk} determine the feasibility of a solution for the backhaul points with respect to vehicle capacity. These variables appear in the constraint set (4), (5), and (6), forming a second GAP for the backhaul points. The remaining variables x_{ijk} represent the sequencing decisions for each of the K independent routes. Given a feasible solution for u_{ik} and v_{jk}, there will exist a feasible solution of x_{ijk} satisfying constraints (7)–(11) which is the constraint set of a Traveling Salesman Problem with a single side constraint for each vehicle. Constraint (9) represents the restriction that there should only be one link connecting linehaul with backhaul on each route.

Fisher and Jaikumar (1978) have considered a similar model for the VRP and proposed a solution algorithm based on Bender's decomposition. A similar approach could be taken here by considering the generalized assignment problems as the master problem $[\boldsymbol{Y} \equiv (u_{ik}, v_{jk})]$ and the TSPs with side constraints as the subproblem $[\boldsymbol{X} \equiv x_{ijk}]$. Let $f(\boldsymbol{u}_k, \boldsymbol{v}_k)$ represent the cost of an optimal tour of the linehaul and backhaul points assigned to route k. Also let $g(u_{ik} \mid \boldsymbol{u}_k, \boldsymbol{v}_k)$ represent the cost of assigning linehaul point i to route k, and $h(v_{jk} \mid \boldsymbol{u}_i, \boldsymbol{v}_k)$ the cost of assigning backhaul point j to route k. Note that g and h are both functions of all the other points assigned to route k, since they depend on the optimal linehaul-backhaul tour through these points. Thus, the function

$$f(\boldsymbol{u}_k, \boldsymbol{v}_k) = g(u_{ik} \mid \boldsymbol{u}_k, \boldsymbol{v}_k)\boldsymbol{u}_k + h(v_{jk} \mid \boldsymbol{u}_k, \boldsymbol{v}_k)\boldsymbol{v}_k$$

is extremely complicated and highly nonlinear. The Benders' decomposition algorithm constructs a piecewise linear approximation of the function $f(\boldsymbol{u}_k, \boldsymbol{v}_k)$.

In order to actually implement this approach, one must be able to solve the subproblems to optimality, which means that K special TSPs must be solved to optimality at each Benders' iteration in order to generate a cut to be passed to the master problem. Since the TSP itself is not easily solved to optimality, a TSP with the above side constraint will certainly not be any easier to solve. Perhaps the best approach for obtaining this solution would be to solve iteratively the linear programming relaxation and add constraints until the optimal integer solution is found. Once this problem has been solved, the Bender's cut can be formed and added to the GAP master problem, another hard problem. Considering the amount of work which will have to be done at each iteration of the decomposition procedure, this approach does not appear to provide a viable technique for real-time implementation for realistic problem sizes.

However, an heuristic procedure could be developed based on the structure suggested by the above decomposition. This procedure will require an anlysis of the problem structure to gain insight into an intelligent way to approximate the functions $g(u_{ik} \mid \boldsymbol{u}_k, \boldsymbol{v}_k)$ and $h(v_{jk} \mid \boldsymbol{u}_k, \boldsymbol{v}_k)$, which are essentially the information link between the clustering problem (the GAPs) and the routing problem (the TSPs with side constraints). An iterative procedure could then be developed as follows: find an initial feasible solution, estimate the cost of assigning all points to all routes (i.e., estimate values for the functions g and h), solve the clustering problem, solve the routing problem to obtain a new feasible solution, and repeat until an acceptable solution has been found.

2.3. An heuristic approach based on optimization methods

Any heuristic should be both efficient and effective. The structure in the model (the GAPs and the special TSPs) suggest that an effective algorithm should take these problems into account. Efficiency can be attained in more than one way: first the algorithms themselves must be of a polynomial nature, and second by finding a good initial solution, the heuristic itself has a better chance of terminating early at a near-optimal solution.

This provides the basis for a two-phased approach to solving the linehaul-backhaul problem. In the first phase, a good initial solution is generated. In the second phase, information is derived from the initial solution to be used for initiating an improvement algorithm. This paper will deal with a study which has been conducted for obtaining an initial feasible solution from heuristics based on spacefilling curves.

3. Generating an initial solution

Bartholdi and Platzman (1984) have developed fast algorithms based on spacefilling curves which can be used not only for sequencing points, but also for clustering. Following is a description of the notion of spacefilling curves, two spacefilling curve heuristics, and adaptations of the heuristics for VRPB.

3.1. Spacefilling curves

A spacefilling curve is a continuous mapping of the unit circle onto the unit square. Effectively one may think of the unit circle as then being straightened to form the unit interval. Figure 1 illustrates the idea of a spacefilling curve.

A critical property of spacefilling curves is that they tend to preserve 'nearness' among points. That is, if two points are close to each other on the unit interval, then they will be close together in the plane. On the other hand, two points which are near to each other in the plane will be *likely* to fall close together on the interval. This nearness preservation property occurs because in traversing a spacefilling curve one tends to visit all the points in one region of the plane before going on to a different region.

3.2. Spacefilling curve heuristics

Bartholdi and Platzman (1984) and Bartholdi et al. (1983) introduced the idea of using spacefilling curves to develop heuristics for routing problems. The advantages of these heuristics are their ease of execution and their speed, in addition to a fairly good solution quality. These heuristics have been investigated and adaptations made for the linehaul-backhaul problem.

The spacefilling curve transformation can be used to heuristically solve the clustering problem. One clustering problem on the line is the K-median problem, which can be stated as: given n points, choose K of these points as 'medians' so that the total distance from each point to its closest median is as small as possible. Once the points are transformed to the line, the unit interval can be divided into K identical subintervals. The medians are chosen to be those points closest to the midpoints of the subintervals. It is a simple computation to assign each of the points to its closest median.

The routing problem can also be solve heuristically via the spacefilling curve. By applying the spacefilling curve transformation, the points can be visited in the same order as they appear on the unit interval, giving an heuristic solution to the

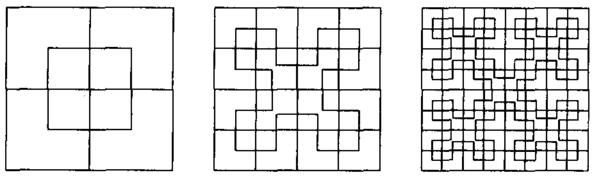
Figure 1. An example of a spacefilling curve

traveling salesman problem. Platzman and Bartholdi (1984) show that the computational effort of this algorithm is O(n log n) and that this tour may be fairly long; the worst case bound is $2\sqrt{n}$. However, the ratio of heuristic to optimal tour lengths is shown to be O(log n) and conjectured to be a constant.

3.3. Adaptations of spacefilling curve heuristics to the linehaul-backhaul problem

By employing adaptations of spacefilling curve heuristics it is possible to obtain a good initial feasible solution for VRPB *very* quickly. Note also that both phases, clustering and routing, can be accomplished almost simultaneously. Two spacefilling curve heuristics have been developed for generating an initial solution. One is a 'greedy' method; the second is based on a solution to the K-median problem. These procedures are described below.

The only difference in the two algorithms is the way in which the points are clustered. There are two sets of points (pickup and delivery) to consider. Each set is clustered independently in order to quickly generate a solution. The essential components of each algorithm are as follows.

VRPB spacefilling curve heuristic

(1) Transform the set of delivery points to the unit interval via the generic spacefilling curve heuristic, forming a sorted list of points on the unit interval.
(2) Cluster the points.
(3) Route the points by visiting the points in each cluster in the order they appear in the sorted list, smallest to largest.
(4) Repeat steps (1)–(3) for the backhaul points.
(5) Link each linehaul route to its corresponding backhaul route and to the distribution center.

The clustering step for the greedy algorithm is as follows.

Greedy clustering step

Assume without loss of generality that the total linehaul demand is larger than the total backhaul supply. Starting with the smallest delivery position not yet routed, add points to the route until the next point will overload the truck. Begin the next route with this point. Repeat until all points are routed.

This step is modified slightly for the backhaul points by adding the condition that a point will begin the next route if its position is greater than the position of the last point on the corresponding linehaul route (see, for example, point B_8 in Figure 2). This condition provides some insurance that the corresponding linehaul and backhaul routes are close together.

The K-median algorithm first requires an input of the number of routes, K, to be formed. The clustering step is then performed as follows.

K-median clustering step

Solve the K-median problem as per the spacefilling curve clustering heuristic. Begin with the point of smallest position that has not yet been routed. Assign the point to route T, where T is the

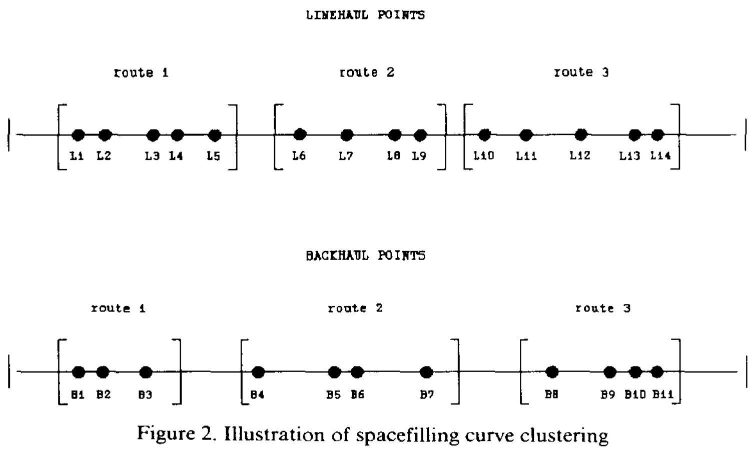

Figure 2. Illustration of spacefilling curve clustering

median point closest to the point, if the truck is not overloaded. If truck capacity is violated, assign the point to route $T+1$. Repeat until all points are routed. If all points cannot be routed, increment K by 1 and repeat the clustering step.

The linehaul and backhaul portions of the route are joined to each other and to the DC by choosing the best combination of links based on actual Euclidean distance in the two-dimensional plane.

4. Computational study

Empirical evaluation of the performance of these heuristics involves computing the total distance traveled by all the vehicles. Since the solutions obtained by the two heuristics do not consider distance per se, the total cost must be obtained after the solutions have been obtained. Euclidean distance is used as the measure of cost. In reality, this measure may prove to be too inaccurate, in which case a matrix of actual measured distances may be required. Also, acceptable accuracy may be attained by using a regression model to approximate actual distances from Euclidean distances.

Since the spacefilling curve (SFC) heuristic provides only approximate solutions to the routing problem, it is likely that improvement in the total cost can be found. In order to better analyze the initial solutions obtained from the heuristics described above, two additional procedures were implemented. First, keeping the initial clusters and the initial interface points fixed, the tours were improved heuristically be invoking both 2-opt and 3-opt on the initial sequence. Second, an optimal route was obtained for the fixed clusters and interface points by employing a branch and bound routine taken from Sysło et al. (1983). Next, the effect of freezing the interface points while keeping the clusters constant was investigated. Finally, in an effort to understand the overall quality of the SFC heuristics, comparisons were made with solutions obtained from the modified Clarke–Wright algorithm developed by Deif and Bodin (1984).

These procedures were coded in Microsoft Pascal and implemented on an IBM-XT with an 8087 numerical coprocessor. Routines were included for graphical as well as alpha-numeric output. All test problems were randomly generated with pickup and delivery points being uniformly distributed and the demands normally distributed with a mean of 500 units and a standard deviation of 200 units. Fifteen test problems were generated for analysis. Linehaul problem size were generated as 20, 30, 45, 75, and 100 points. The number of pickup points corresponded to 25, 50, and 100% of the number of delivery points.

For each problem generated, the capacity of the vehicle is a design parameter. This was varied for each of the problems based on an average number of points desired for each route. The variation of vehicle capacities resulted in a total of 57 test problems.

For each of these problems, several solution values were computed. An important consideration in selecting the best solution between alternatives is the number of routes produced. A simple lower bound on the number of routes required is

$$R_{\min} = \left\lceil \frac{\max\{\text{linehaul demand, backhaul demand}\}}{\text{truck capacity}} \right\rceil,$$

where $\lceil x \rceil$ is the smallest integer larger than x and $R_{\min}$ represents the minimum number of routes.

The greedy algorithm allows no user control over the number of routes produced. Thus, there are exactly 57 greedy solution values. The number of routes produced by the K-median procedure depends on the value of K input to the algorithm. Therefore, various values of the K-median solution were found for each of the 57 problems. The total number of K-median solutions was 132.

Next, an analysis of the results of the three solution values for each of the 189 problems runs (57 greedy solutions and 132 K-median solutions) is given.

4.1. *K-median vs. greedy solutions*

The 57 greedy solutions are compared to the corresponding best K-median solutions (best meaning smallest total distance) according to the following criteria:

(1) number of routes,
(2) vehicle capacity utilization,
(3) paired t-test.

A discussion of the results is given below.

Number of routes

The greedy algorithm produced 54 out of 57 solutions that were either equal to or one more than the minimum number of routes. From the K-median algorithm, only 25 of 27 solutions met this criterion. The K-median problem is solved ignoring the capacity constraints. Thus, it is possible to form clusters of points which are in extreme violation of the vehicle capacity. The algorithm then sometimes proceeds to 'paint itself into a corner", by not having sufficient capacity for the remaining points on the last route. Remedies for this problem will be discussed in the conclusions to this paper.

Vehicle capacity utilization

As a measure of the utilization of the vehicle capacity, a percent slack was computed as

$$\frac{\text{total truck capacity-max total demand}}{\text{total truck capacity}} \times 100\%,$$

where total truck capacity is the number of vehicles times the capacity per vehicle. This slack represents the unused truck capacity that is available. For the Greedy algorithm, the solutions had a range of percent slack of [4.5, 35.64] with an average of 15.8%. The solutions from the K-median algorithm resulted in a percent slack ranging over [6.9, 59.6] with a mean of 32.2%. These results indicate that the greedy method provides better solutions with respect to capacity constraints.

Paired t-test

A paired t-test was run with the first half of each pair represented by a solution (i.e., total distance traveled) obtained from the greedy algorithm, and the second by the corresponding best solution from the K-median algorithm. The results indicated that, at a level of significance of 95%, the hypothesis that the average difference is 0 cannot be rejected. In other words, there is essentially no difference in the two algorithms in terms of the total number of miles traveled.

Conclusions for greedy vs. K-median

In general it appears that the greedy algorithm plus 2-opt/2-opt is better suited for generating an initial solution to VRPB. However, the original rationale behind developing the K-median algorithm was to find a 'smarter' way of clustering the points in the unit interval, i.e., by grouping the points based on their nearness to each other. The greedy algorithm virtually ignores distance and location of points within a cluster and concentrates on the truck capacity constraints. The solutions obtained from this algorithm tend to have fewer routes and better vehicle utilization. On the other hand, the K-median algorithm is location/distance oriented while ignoring truck capacity constraints. The problems displayed by this algorithm are due to forming clusters without consideration for the demand data for the points in the clusters. The total length of a set of routes is not appreciably improved by using one algorithm over the other. Thus, the issures discussed previously must be considered in making a final judgement concerning which of the two algorithms is better for a given problem instance. In addition, there are some simple improvements to the algorithms which could be implemented. These will be discussed in Section 5.

4.2. *Initial solutions vs. two-opt and three-opt solutions*

In this part of the experiment, the clusters and interface points were kept fixed. For all 189 test problems, the improvement procedure always (with the exception of one problem) gave some improvement over the initial solution. The range in the percentage decrease was [0.0, 31.2] with a mean of 8.2%.

The execution times, however, did show considerable increase. The range of execution times for the initial solutions was [1.26, 9.12] seconds. For initial solutions plus 2-opt and 3-opt, the range increased to [4.95, 79.9]. Although the time required to implement this improvement is substantial compared to the time to generate the initial solution, the marked improvement in the solution values justifies the computational expense.

4.3. *Two-opt and three-opt solutions vs. optimal branch and bound solutions*

Again the clusters and interface points were kept fixed in this part of the experiment. Of the 189 solutions examined, 152, or 80% of the 2-opt and 3-opt solutions were identical to the branch and bound solutions. Of the remaining 37 solutions which were suboptimal, all were within 1% of optimality. The execution times for the branch and bound procedure ranged from 4.56 to 2019.0 seconds. Even though for some problems branch and bound was faster, it should be noted that the 2-opt/3-opt solution was passed to the branch and bound procedure as the initial upper bound. Such a small increase in the overall solution quality is an indication that the 2-opt and 3-opt solutions are good enough for practical purposes.

4.4. *Optimal routes for the fixed clusters*

The influence of freezing the interface points was investigated next. As a further tool in evaluating the initial solution procedures, the algorithm described below was used to generate the optimal routes for the clusters determined by the initial solutions. The test problems resulted in a wide range of TSP problem sizes: from a minimum of 3 points (2 linehaul and 1 backhaul) to a maximum of 27 (15 linehaul and 12 backhaul).

An optimal solution to the special TSP can be found as follows. Given a cluster of linehaul and backhaul points to be serviced by a single vehicle, suppose that the backhaul points in the cluster are indexed $1, 2, \ldots, J$. For each backhaul point j, adjoin the point to the linehaul cluster, setting the distance from j to the DC = 0. The optimal solution for the TSP on this extended cluster gives the best linehaul interface point q for the given backhaul point j with a cost of LTSP. The optimal solution for the backhaul cluster with fixed interface point j (with distance from j to DC = 0) has a cost equal to BTSP. The route cost is then LTSP + BTSP. The minimum route cost over all backhaul points j is then the optimal TSP cost for the route with corresponding optimal interface points q (linehaul) and j (backhaul).

Statistics were computed based on the percent error of the heuristic over the optimal solution. The average error of the spacefilling curve heuristic plus 2-opt/2-opt improvement was 2.3% and in 58% of the cases the heuristic found the optimal solution for the routing problem. The average error of the heuristic among the suboptimal solutions was only 5.5%.

These statistics indicate that the spacefilling curve heuristics followed by 2-opt/3-opt provide an excellent method for solving the special TSP over a fixed cluster of points.

4.5. *Modified Clarke–Wright algorithm vs. spacefilling curve heuristics*

The previous analyses are only useful in understanding how the spacefilling curve heuristics perform with respect to each other, and on a route-by-route bases. There is no information to indicate how these algorithms perform with regard to the global solution. That is, how do solution quality and computational efficiency compare with the

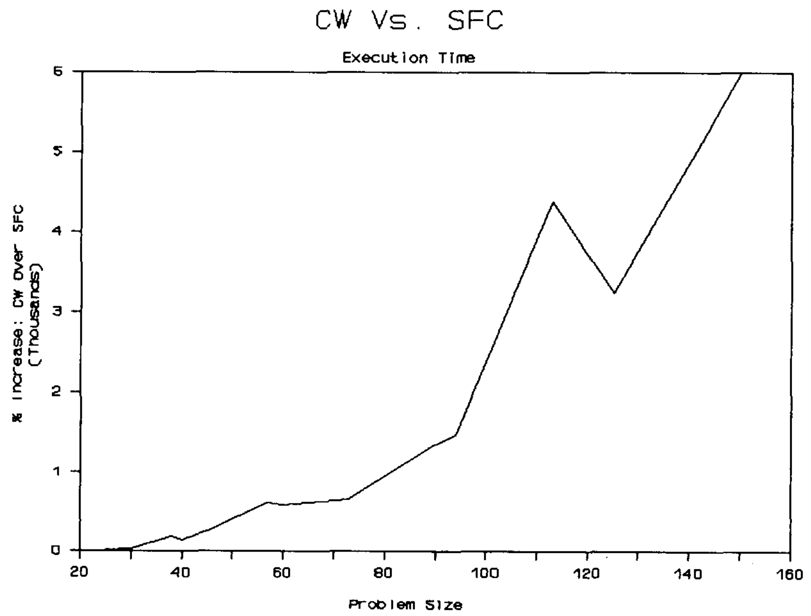

Figure 3. Comparison of execution time for CW and SFC

best solutions which can be obtained? Optimal solutions for the set of test problems can be obtained, but only with an inordinate amount of computational time and expense. In addition, the literature does not provide a set of benchmark problems with which to make comparisons. Thus, the only approach that can be taken at this time is to compare the SFC heuristics with the results from other heuristics for VRPB.

The only previously developed heuristic algorithm for generating solutions to VRPB is the modified Clarke–Wright procedure discussed by Deif and Bodin (1984). The Clarke–Wright algorithm is based on a 'savings' computation, s_{ij}, determined by the distance saved from combining two points, i and j, into a single route as opposed to assigning them to separate single-point routes. For the modified version for backhauls the adjusted savings values are computed as, $s_{ij} = s_{ij} - \alpha s_{\max}$, where $s_{\max}$ is an estimate of the maximum savings value and α is a penalty multiplier, $\alpha \in [0.1, 0.3]$.

Since the Clarke–Wright procedure is perhaps the most widely used method in commercial vehicle routing packages, the modified algorithm is expected to give reasonable solutions to VRPB. This algorithm, which will be referred to as CW, was coded (also in Microsoft Pascal) and executed on the same set of test problems as before. The spacefilling curve heuristics will be referred to as SFC and all data is averaged over both the greedy and K-median output. For both CW and SFC, 2-opt and 3-opt were executed on all solutions. The results are given below.

Execution times

The range of execution times for CW was [6.15, 2445.61] seconds, with an average of 441.36 seconds. In order to compare these times with the execution times for SFC plus 2-opt/3-opt, comparisons were made based on problem size. An average time was computed for all the problems of a particular size and the percent increase was computed as

$$[(\text{avg. CW} - \text{avg. SFC})/\text{avg. SFC}].$$

The results are illustrated by the graph shown in Figure 3. This graph illustrates that the modified Clarke–Wright algorithm decreases in efficiency as the problem size increases. Recall that the SFC plus 2-opt/3-opt times ranged from approximately 5 seconds to 80 seconds. Thus, the spacefilling curve times are relatively constant compared to the modified Clarke–Wright execution times.

Solution quality

The solution quality of the CW solutions was also compared to SFC in terms of problem size.

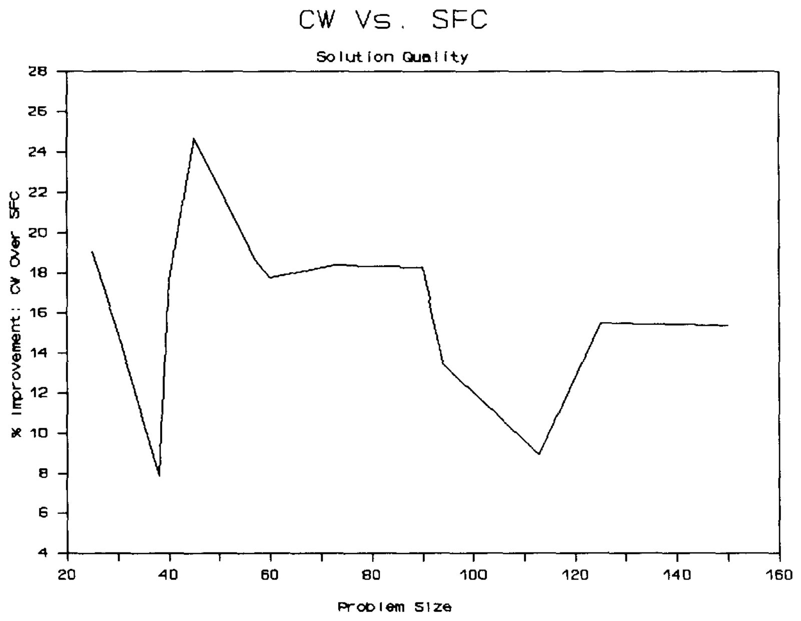

Figure 4. Comparison of solution quality for CW and SFC

An average solution value for each problem size considered was computed for both CW and SFC. The percent improvement of CW over SFC (again with 2-opt/3-opt included for both) was calculated as

$$[(\text{avg. SFC} - \text{avg. CW})/\text{avg. CW}].$$

The range of improvement was [6.2%, 28.9%] with an average of 16.3%. The overall results are shown in Figure 4.

From the graph of Figure 4 it is difficult to draw any general conclusions about the performance of the SFC heuristics in terms of solution quality. If we assume that the CW algorithm provides a good benchmark against which to measure solution quality, it appears that some SFC solutions are considerably better than others, but the size of the problem is not the determining factor. While some of the SFC solutions were fairly good compared to CW (as close as 8%) others were quite bad (up to about 25% worse). This means that initial solutions generated with SFC will have an inconsistent quality with respect to the optimal solution. However, the comparison of execution times indicates that the SFC heuristics are a viable method for the initial solution procedure.

5. Conclusions and future research

The primary objective of this work was to study the feasibility of using spacefilling curve heuristics to derive an initial solution to the linehaul-backhaul problem. The computational study leads to three major conclusions. First, in terms of the efficiency of an overall algorithm for the problem, the SFC method is appropriate for producing an initial solution. With average run times under one minute on an IBM-XT, this approach quickly finds a solution from which an improvement algorithm can begin. Also, execution times are shown to be relatively constant when compared with a modified version of the Clarke–Wright heuristic. The basic Clarke–Wright algorithm has been widely accepted in commercial routing packages for the classical VRP. Second, the solution of the routing subproblem from the SFC heuristics plus 2-opt and 3-opt is so good (all within 1% of optimality) that this method could also be incorporated into the improvement algorithm. Finally, the solution quality of the SFC methods is still somewhat in question based on the results of the comparison with the modified Clarke–Wright algorithm. While efficiency is a plus, there must be some trade-off between computational time and quality of solution. Potential solutions for this problem will disscussed below, as well as other improvements to the algorithms.

The SFC methods are currently formulated so that once the spacefilling curve transformation has been made, the clustering process always begins at the point 0 on the unit interval. Many alternative solutions could be produced by changing this starting point. It is likely that many of the 'bad' solutions found in the tests described above have a much better companion solution that could be easily found with a simple grid search for the best starting point.

Other improvements in these algorithms could be implemented as follows. The first improvement is for the K-median algorithm. The major problem is that capacity constraints are completely ignored in the clustering step. This problem can be resolved by first computing the distance on the line from each median point to each demand point, yielding an $m \times n$ cost matrix. Since the clustering problem is a generalized assignment problem (GAP), this cost matrix can be used to heuristically solve the GAP. One procedure for doing this is the savings regret heuristic of Martello and Toth (1981). This change in the clustering step will prevent the algorithm from making myopic assignments of points to clusters which at later stages can results in insufficient capacity to assign some points at all. It must be observed that solving the GAP even heuristicallyrequires a much larger computational effort than the simple K-median problem.

The second improvements is for the greedy algorithm. Since the clustering step in this method greedily puts the points onto successive routes, it is often the case that the final route has only one or two points left to be assigned to it, resulting in poor vehicle utilization for this route. In addition, a shortcoming of the algorithm is that there is no control over the number of routes generated. A remedy for both problems is to compute an *average load* for the routes, computed as the total demand of the points divided by the number of routes desired (this number of routes to be input to the algorithm). Then in the clustering step, a route will be terminated when its load just exceeds the average load.

In conclusion, the spacefilling curve heuristics are suited to the purpose of producing initial clusters very quickly due to their extreme speed. Even with the addition of 2-opt and 3-opt, the run times are moderate. Thus, the spacefilling curve heuristics developed here will provide initial clusters very cheaply in terms of computational time, but with trade-offs in solution quality.

The remainder of the research for the linehaul-backhaul problem involves further analysis of the problem structure in order to appropriately design an improvement algorithm. Investigations of the described enhancements to the SFC heuristics will also be made in order to get the best performance from the improvement algorithm. This study will involve theoretical analysis and experimental examination of the procedures in order to find the best solution methods for the vehicle routing problem with backhauls. The results of this study will be reported in a later paper.

Acknowledgement

This research was supported in part by the Member Companies of the Material Handling Research Center of the Georgia Institute of Technology.

References

Bartholdi, J.J. and Platzman, L.K. (1988), "Heuristics based on spacefilling curves for combinatorial problems in Euclidean space", *Management Science* 34 (3), 291–305.

Bartholdi, J.J., Platzman, L.K., Collins, R.L., and Warden, W.H. (1983), "A minimal technology routing system for meals-on-wheels", *Interfaces* 13 (3), 1–8.

Bodin, L.D., and Berman, L. (1979), "Routing and scheduling of school buses by computer", *Transportation Science* 13 (2) 113–129.

Bodin, L.D., and Golden, B.L. (1981), "Classification in vehicle routing and scheduling", *Networks* 11 (2) 97–108.

Christofides, N., Mingozzi, A., and Toth, P. (1981), "Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations", *Mathematical Programming* 20, 255–282.

Christofides, N., and Eilon, S. (1972), "Algorithms for large-scale traveling salesman problems", *Operations Research Quarterly* 23, 511–518.

Clarke, G., and Wright, J. (1964), "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research* 12, 568–581.

Cullen, F.H. (1984), "Set partitioning based heuristics for interactive routing", Unpublished Doctoral Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

Cullen, F.H., Jarvis, J.J., and Ratliff, H.D. (1981), "Set partitioning based heuristics for interactive routing", *Networks* 11 (2) 125–143.

Dantzig, G.B., and Ranser, J.H. (1959), "The truck dispatching problem", *Management Science* 6, 80–91.

Deif, I., and Bodin, L.D. (1984), "Extension of the Clarke–Wright algorithm for solving the vehicle routing problem with backhauling", in: A.E. Kidder (Ed.) *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, MA, 75–96.

Fisher, M.L., and Jaikumar, R. (1978), "A decomposition algorithm for large-scale vehicle routing", Working paper 78-11-05, Wharton Department of Decision Sciences, University of Pennsylvania.

Fisher, M.L., and Jaikumar, R. (1981), "A generalized assignment heuristic for vehicle routing," *Networks* 11 (2) 109–124.

Gillet, B., and Miller, L. (1974), "A heuristic algorithm for the vehicle dispatch problem", *Operations Research* 22, 340–349.

Golden, B., and Miller, L. (1974), "A heuristic algorithm for the vehicle dispatch problem", *Operations Research* 22, 340–349.

Golden, B.L., Baker, E.K., Alfaro, J.L., and Schaffer, J.R. (1985), "The vehicle routing problem with backhauling: Two approaches", Working Paper Series M/S 85-037, College of Business and Management, University of Maryland.

Kearney, A.T., Inc. (1984), *Measuring and Improving Productivity in Physical Distribution*, A report prepared for the National Council of Physical Distribution Management, Oak Brook, IL.

Lenstra, J.K., and Rinnooy Kan, A.H.G. (1981), "Complexity of vehicle routing and scheduling problems", *Networks* 11 (2), 221–227.

Lin S., and Kernighan, B. (1973), "An effective heuristic algorithm for the traveling salesman problem", *Operations Research* 21, 498–516.

Magnanti, T.L. (1981), "Combinatorial optimization and vehicle fleet planning: Perspectives and prospects", *Networks* 11 (2), 179–213.

Martello, S., and Toth, P. (1981), "An algorithm for the generalized assignment problem", in: J.P. Brans, (Ed.), *Operational Research '81*, North-Holland, Amsterdam.

Newton, R., and Thomas, W. (1969), "Design of school bus routes by computer", *Socio-Economic Planning Sciences* 3, 75–85.

Platzman, L.K., and Bartholdi, J.J. (1984), "Spacefilling curves and the planar traveling salesman problem", PDRC Report Series 83-02, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

Ross, G.T., and Soland, R.M. (1975), "A branch and bound algorithm for the generalized assignment problem", *Mathematical Programming* 8, 91–103.

Sysło, M.jM., Deo, N., and Kowalik, J.S. (1983), *Discrete Optimization Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.

Yano, C.A., Chan, T.J., Richter, L., Culter, T., Murty, K.G., and McGettigan, D. (1987), "Vehicle routing at quality stores", *Interfaces* 17 (2), 52–63.