OMEGA Int. J. of Mgmt Sci., Vol. 12, No. 5, pp. 497–504, 1984
Printed in Great Britain.

0305-0483/84 $3.00 + 0.00

A Heuristic Algorithm for the Period Vehicle Routing Problem

CCR TAN
JE BEASLEY

Imperial College of Science and Technology, London, UK

(*Received February 1984; in revised form April 1984*)

In this paper we consider the period vehicle routing problem, which is the problem of designing routes for delivery vehicles to meet customer service level requirements (not all customers require delivery on every day in the period). A heuristic algorithm, based upon the daily vehicle routing algorithm of Fisher and Jaikumar, is presented and computational results are given for test problems drawn from the literature.

INTRODUCTION

THE PERIOD vehicle routing problem (PVRP) is the problem of designing routes for delivery vehicles for all the days of a given T-day period where not all customers require delivery on every day in the period. Typically, if a customer requires k ($\leq T$) visits during the period then these visits may only occur in one of a given number of allowable k-day delivery combinations. For example, if a customer requires two deliveries in a 5-day week then the allowable delivery combinations might be Monday/Wednesday, Tuesday/Thursday or Wednesday/Friday with no other combinations of delivery days being acceptable. Note here that the PVRP is a generalisation of the single-day (daily) vehicle routing problem (VRP) which has been extensively discussed in the literature (see [8, 11]).

In this paper we present a heuristic algorithm for the PVRP based upon the successful single-day vehicle routing algorithm of Fisher *et al.* [5, 6], but first we review the work in the literature dealing with the PVRP.

PREVIOUS WORK

Early work on the PVRP was carried out by Beltrami and Bodin [1] in their study of hoist compactor routing. They considered two approaches to the problem:

(1) developing routes which were then assigned to delivery days; and
(2) assigning customers to delivery days and then routing each day separately.

Russell and Igo [10] proposed three approaches to the problem:

(1) assigning customers to delivery days by a clustering algorithm where the clusters for each day were formed around customers with a single allowable delivery combination,
(2) an adaptation of the single-day vehicle routing heuristic MTOUR of Russell [9] which is based on the travelling salesman heuristic of Lin and Kernighan [7],
(3) an adaptation of the single-day vehicle routing heuristic of Clarke and Wright [3].

Christofides and Beasley [2] presented heuristic algorithms for the PVRP based upon an initial choice of customer delivery days to meet service level requirements followed by an interchange procedure to improve upon the choice of delivery days. Their algorithms replaced the underlying daily VRP by:

(1) a median problem; and
(2) a travelling salesman problem

in order to computationally evaluate interchanges in customer delivery combinations.

In the next section we develop a formulation of the PVRP based upon the formulation of the single-day VRP given by Fisher and Jaikumar [6]. Note here that although Fisher *et al.* [5] have discussed an application of their approach [6] to a practical problem involving choice of customer delivery days, few details of the algorithm used were given.

PROBLEM FORMULATION

In this section we first review the formulation of the single-day VRP given by Fisher and Jaikumar [6] and then extend that formulation to the PVRP. This leads to a large, complex, zero-one integer program and so we develop a simpler formulation of the PVRP as a smaller zero-one integer program and discuss how this program can be solved computationally.

Fisher and Jaikumar formulation

Fisher and Jaikumar [6] formulated the single-day VRP in the following way:

Let

q_i be the demand of customer i $(i = 1, \ldots, n)$

Q_k be the capacity of vehicle k $(k = 1, \ldots, K)$

d_{ik} be a measure (derived in some fashion) of the distance contribution of customer i $(i = 1, \ldots, n)$ to the route followed by vehicle k $(k = 1, \ldots, K)$ if customer i were to be delivered to by vehicle k.

Define

$x_{ik} = 1$ if customer i $(i = 1, \ldots, n)$ is delivered to by vehicle k $(k = 1, \ldots, K)$

$= 0$ otherwise,

then the single-day VRP can be formulated as:

$$\min \sum_{i=1}^{n} \sum_{k=1}^{K} d_{ik} x_{ik} \tag{1}$$

$$\text{s.t.} \sum_{k=1}^{K} x_{ik} = 1 \quad i = 1, \ldots, n \tag{2}$$

$$\sum_{i=1}^{n} q_i x_{ik} \le Q_k \quad k = 1, \ldots, K \tag{3}$$

$$x_{ik} \in (0, 1) \quad i = 1, \ldots, n \quad k = 1, \ldots, K \tag{4}$$

Equation (1) represents the total distance travelled, equation (2) ensures that each customer is assigned to a vehicle, equation (3) ensures that the vehicle capacity is not exceeded and equation (4) is the integrality constraint.

As noted in [6] the program [equations (1) to (4)], is a linear (n by K) generalised assignment problem (for which computationally effective optimal algorithms exist), the solution to which defines a (capacity) feasible assignment of customers to vehicles. The delivery sequence for each vehicle can be determined by applying any (heuristic or optimal) travelling salesman algorithm to the customers assigned to the vehicle.

Fisher and Jaikumar [6] define the (d_{ik}) matrix by generating K 'seed points' (one for each vehicle) and letting d_{ik} equal the extra distance travelled when customer i is inserted into the route in which vehicle k travels out from the depot to its seed point and back again. Computational results for the algorithm on a number of test problems drawn from the literature indicate that the method is, on average, currently the best known method for the single-day VRP.

At first sight it might appear simple to extend the formulation given above from the single-day (n customers, K vehicles) VRP to the T-day [(n customers, KT vehicles (K available each day)] PVRP and still retain the linear generalised assignment structure. In the next section we show that this is not so and that the extended formulation is a large, complex, zero-one integer program.

Extension to the PVRP

In the PVRP we cannot assign customers directly to vehicles but must first assign a customer to an allowable delivery combination, thereby defining the delivery days for the customer, and then assign the customer to a delivery vehicle on each of the chosen delivery days. This means that the PVRP is not an n by KT linear generalised assignment problem, as might be supposed, but a much larger and more complex problem. Formally:

Let

R be the number of distinct delivery combinations

S_i be the set of allowable delivery combinations for customer i, $S_i \subseteq [1, 2, \ldots, R]$ $(i = 1, \ldots, n)$

q_i be the demand of customer i $(i = 1, \ldots, n)$ for each delivery

Q_k be the capacity of vehicle k $(k = 1, \ldots, K)$

d_{ikt} be a measure (derived in some fashion) of the distance contribution of customer

i ($i = 1, \ldots, n$) to the route followed by vehicle k ($k = 1, \ldots, K$) on day t ($t = 1, \ldots, T$) if customer i were to be delivered to by vehicle k on day t (note here that, for simplicity, we assume all K vehicles are available on each day of the period).

Let

$a_{rt} = 1$ if delivery combination r ($r = 1, \ldots, R$) involves delivery on day t ($t = 1, \ldots, T$)
$= 0$ otherwise.

Define

$x_{ir} = 1$ if customer i ($i = 1, \ldots, n$) is assigned to delivery combination $r (\in S_i)$
$= 0$ otherwise.

$y_{ikt} = 1$ if customer i ($i = 1, \ldots, n$) is delivered to by vehicle k ($k = 1, \ldots, K$) on day t ($t = 1, \ldots, T$)
$= 0$ otherwise.

Then the PVRP can be formulated as:

$$\min \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{t=1}^{T} d_{ikt} y_{ikt} \tag{5}$$

$$\text{s.t.} \sum_{r \in S_i} x_{ir} = 1 \quad i = 1, \ldots, n \tag{6}$$

$$\sum_{k=1}^{K} y_{ikt} = \sum_{r \in S_i} a_{rt} x_{ir} \quad i = 1, \ldots, n \quad t = 1, \ldots, T \tag{7}$$

$$\sum_{i=1}^{n} q_i y_{ikt} \leq Q_k \quad k = 1, \ldots, K \quad t = 1, \ldots, T \tag{8}$$

$$x_{ir} \in (0, 1) \quad \forall r \in S_i \quad i = 1, \ldots, n \tag{9}$$

$$y_{ikt} \in (0, 1) \quad i = 1, \ldots, n \quad k = 1, \ldots, K \quad t = 1, \ldots, T \tag{10}$$

Equation (5) represents the total distance travelled and equation (6) ensures that an acceptable delivery combination is chosen for each customer. In equation (7) the term on the right-hand side is one if a combination is chosen for customer i that involves a delivery on day t and zero otherwise. Hence equation (7) ensures that a vehicle is used for a delivery to a customer i on some day t if the delivery combination chosen for i requires it. Equation (8) ensures that the vehicle capacity is not exceeded and equations (9) and (10) are the integrality constraints.

It is clear that the program [equations (5) to (10)] is a large, complex, zero-one integer program involving $0(n + nT + KT)$ constraints and $0(nKT)$ variables which would be (computationally) difficult to solve optimally (unlike the Fisher and Jaikumar [6] formulation of the single-day VRP which being only a small linear generalised assignment problem is relatively easy to solve optimally).

We see then that the extension of the Fisher and Jaikumar formulation to the PVRP leads to a program that is difficult to solve optimally. However, by regarding the PVRP as the problem of assigning customers to delivery combinations (and neglecting the assignment of customers to vehicles) we can represent the PVRP by a smaller zero-one integer program. We outline this approach in the next section.

Formulation of the PVRP

We can regard the PVRP as the problem of assigning customers to delivery combinations within an overall constraint upon the total demand on any day of the period. Formally:

Let

D_{it} be a measure (derived in some fashion) of the distance contribution of customer i ($i = 1, \ldots, n$) to any route involving customer i on day t ($t = 1, \ldots, T$).

We shall call (D_{it}) the *contribution matrix*. The PVRP can now be formulated as:

$$\min \sum_{i=1}^{n} \sum_{t=1}^{T} \sum_{r \in S_i} D_{it} a_{rt} x_{ir} \tag{11}$$

$$\text{s.t.} \sum_{r \in S_i} x_{ir} = 1 \quad i = 1, \ldots, n \tag{12}$$

$$\sum_{i=1}^{n} \sum_{r \in S_i} q_i a_{rt} x_{ir} \leq \sum_{k=1}^{K} Q_k \quad t = 1, \ldots, T \tag{13}$$

$$x_{ir} \in (0, 1) \quad \forall r \in S_i \quad i = 1, \ldots, n \tag{14}$$

Equation (11) represents the total distance travelled, equation (12) ensures that an acceptable delivery combination is chosen for each customer, equation (13) ensures that the total delivered on any day does not exceed the total vehicle capacity and equation (14) is the integrality constraint. We shall call the above program [equations (11) to (14)] the PVRP program.

Note here that the PVRP program only assigns customers to delivery days and we still have to solve the T (independent) single-day VRP's that result. This could conveniently be done using the Fisher and Jaikumar [6] algorithm for the VRP (or any other algorithm for the VRP).

Note also that, so far, we have only considered the PVRP in terms of a constant amount (q_i for each customer i) being supplied at each delivery. It is clear from the nature of equation (13) that we could easily modify the PVRP program to cope with the situations where:

(a) the amount supplied at each delivery depends upon the day of delivery (define q_{it} $i = 1, \ldots, n$ $t = 1, \ldots, T$ in the obvious way)
(b) the amount supplied at each delivery depends upon the day of delivery and the customer combination chosen (define q_{irt} $i = 1, \ldots, n$ $\forall r \in S_i$ $t = 1, \ldots, T$ in the obvious way).

Both of these situations require only minor changes to the algorithm presented below.

In the next section we consider how, computationally, we might solve the PVRP program.

Solution of the PVRP program

Considering the PVRP program we see that it involves $(n + T)$ constraints and

$$\left(\sum_{i=1}^{n} |S_i|\right)$$

variables. Hence for a problem involving 100 customers for delivery over a five-day week with an average of three allowable delivery combinations per customer we would have a problem involving 105 constraints and 300 variables. As in linear programming (LP) terms this is not a particularly large problem it raises the possibility of solving the PVRP program via LP.

The LP relaxation of the PVRP program involves replacing equation (14) by the equation

$$x_{ir} \geq 0 \quad \forall r \in S_i \quad i = 1, \ldots, n \tag{15}$$

Note here that the upper limit of one on each variable from equation (14) is automatically enforced by equation (12). Since, from equation (12), we need at least n variables non-zero and there are only $(n + T)$ constraints then we claim that, *at most*, $2T$ variables can be non-integer (fractional) in the optimal solution to the LP relaxation of the PVRP program. This can be seen as follows: from equation (12), for each customer i, we have that in the optimal solution to the LP relaxation of the PVRP program either

(a) exactly one x_{ir} $(r \in S_i)$ value is non-zero (and hence equal to one so integer); or
(b) two (or more) x_{ir} $(r \in S_i)$ values are non-zero (fractional, but summing to one).

Suppose n_1 customers fall into category (a). Since the LP relaxation of the PVRP program has $(n + T)$ constraints then, *at most*, $(n + T)$ x_{ir} variables can be non-zero. Of these $(n + T)$ non-zero values n_1 values are integer and hence at most $(n + T) - n_1$ values are fractional. Now $(n - n_1)$ customers fall into category (b) above and as each customer in category (b) requires, *at least*, two fractional variable values we must have $2(n - n_1) \leq (n + T) - n_1$. Rearranging we get $n_1 \geq (n - T)$ and hence the maximum number of fractional variable values $(n + T) - n_1$ satisfies $(n + T) - n_1 \leq 2T$, i.e. *at most* $2T$ variables can be fractional in the optimal solution to the LP relaxation of the PVRP program. We would expect that, in practice, far fewer than $2T$ variables would be non-integer. Any non-integer variables could be dealt with

(a) heuristically (e.g. by rounding); or
(b) optimally, by the use of cutting planes/tree search.

Since the PVRP program cannot really be regarded as an exact formulation of the PVRP problem (depending, as it does, on actually knowing the (D_{it}) values) we decided to solve the PVRP program heuristically by solving the LP relaxation of the PVRP program exactly and then rounding any non-integer variable values in the following manner:

(1) Let (X_{ir}) represent the values of (x_{ir}) in the solution to the LP relaxation of the PVRP program.
(2) Consider the (X_{ir}) in descending order and for each X_{js} $(s \in S_j)$ assign combination s to customer j if
    (a) j has not previously been assigned a delivery combination; and
    (b) combination s together with the previously assigned delivery combinations do not produce any capacity infeasible days.
(3) At the end of (2) any customer j not assigned a delivery combination is assigned the combination $s (\in S_j)$ with the largest X_{js} value (in which case the (rounded) integer solution will be infeasible).

It is clear that for the PVRP program to give a reasonable assignment of customers to delivery days care must be taken in the definition of the contribution matrix (D_{it}). In the next section we outline our approach to defining this matrix.

CONTRIBUTION MATRIX DEFINITION

In their algorithm for the single-day VRP Fisher and Jaikumar [6] defined their measure

[d_{ik}—equation (1)] of the distance contribution of a customer to a vehicle route by first generating a seed point for each vehicle route and then calculating the extra distance involved in inserting customers into the routes from the depot to the seed points. Since their approach was so successful computationally we will follow a similar approach in defining our measure (D_{it}) of the contribution of a customer to the routes on any particular day. We shall assume here that the reader is familiar with the seed point generation procedure of Fisher and Jaikumar [6].

Initially we want to choose KT seed points (K for each day) but, in general, have little idea of the customers that will be scheduled for delivery on day t (except for any customers i for which there is only one allowable delivery combination, i.e. $|S_i| = 1$). Hence we will choose KT points to act as seed points and associate a day with each one. Formally this can be accomplished as follows.

Step 1. Define

$$Q^* = \left(\sum_{k=1}^{K} Q_k\right) \Big/ K \qquad \text{(the average vehicle capacity)}$$

and use the Fisher and Jaikumar seed generation procedure on the single-day VRP consisting of:

(a) KT vehicles of capacity Q^*
(b) n customers where customer i has demand given by

$$\min\left(q_i \sum_{t=1}^{T} a_{rt} \mid \forall r \in S_i\right) \tag{16}$$

Essentially this equation represents the total amount delivered to customer i over the T-day period.

Step 2. Let P be the entire set of KT seed points as decided at step 1 above. Then to associate a day t with each seed point p we form the matrix (v_{pt}) where v_{pt} is interpreted as the value of having seed point $p(\in P)$ assigned to day t $(t = 1, \ldots, T)$. This matrix is formed by:

(i) let B be the set of customers such that
  (a) the total demand of the customers in B does not exceed the vehicle capacity Q^* $\left(\text{i.e. } \sum_{i \in B} q_i \leq Q^*\right)$
  (b) each customer $i \in B$ could be assigned a combination which would result in a delivery on day t $\left(\text{i.e. } \sum_{r \in S_i} a_{rt} \geq 1\right)$
  (c) the customers in B are the $|B|$ nearest customers to the seed point p with regard to all the customers for whom it is possible to make a delivery on day t
  (d) $|B|$ is a maximum
(ii) then we define v_{pt} by

$$v_{pt} = \sum_{i \in B}\left[\sum_{r \in S_i} a_{rt} / |S_i|\right] \tag{17}$$

where the value in square brackets in equation (17) above is the fraction of those combinations for i which include t as a delivery day. Intuitively the larger the value v_{pt} is the more attractive it is to associate day t with seed point p

(iii) let $y_{pt} = 1$ if seed point $p \in P$ is assigned to day t $(t = 1, \ldots, T)$
$= 0$ otherwise,

then consider the program

$$\max \sum_{p \in P} \sum_{t=1}^{T} v_{pt} y_{pt} \tag{18}$$

$$\text{s.t. } \sum_{t=1}^{T} y_{pt} = 1 \quad \forall p \in P \tag{19}$$

$$\sum_{p \in P} y_{pt} = K \quad t = 1, \ldots, T \tag{20}$$

$$y_{pt} \in (0, 1) \quad \forall p \in P \quad t = 1, \ldots, T. \tag{21}$$

Equation (19) ensures that each seed point is allocated to just one day and equation (20) ensures that K seed points are chosen for each day. This program [equations (18) to (21)] can be viewed as a linear assignment problem and is easily solved to give K seed points for each day. Note here that this approach to assigning seed points to days is based on a similar approach given in [2].

Step 3. Let (Y_{pt}) represent the values of (y_{pt}) in the optimal solution of the above program [equations (18) to (21)]. Then we can define the contribution matrix (D_{it}) by $D_{it} = \min$(extra distance travelled in inserting customer i in the route from the depot to seed point p and back again $| Y_{pt} = 1 \; \forall p \in P$).

Once the contribution matrix has been decided as above then we can solve the PVRP program, as discussed previously, to assign customers to delivery combinations. We felt that (since the choice of seed points for each day was fairly arbitrary) a better solution would be obtained by repeating the process, but with the seed points adjusted to cope with the customer delivery combinations chosen. Hence we have

Step 4. Let V_t $(t = 1, \ldots, T)$ be the set of customers who have a delivery on day t in the solution to the PVRP program obtained as described above. Then use the Fisher and Jaikumar seed generation procedure, for each day t, where the VRP for day t consists of

(a) K vehicles of capacity Q^*
(b) the customer set V_t where customer $i \in V_t$ has demand q_i.

We can use the seed points associated with each day to again define the contribution matrix (D_{it}) and hence to resolve the PVRP program. Whilst step 4 could be repeated, examination of a limited number of computational experiments appeared to show no significant advantage in doing so, and so in the computational results reported later we only carried out step 4 once.

In the next section we discuss how we can evaluate the final set of customer delivery combinations produced by the above algorithm in order to ascertain whether they are a 'good' set of customer delivery combinations or not.

COMBINATION EVALUATION

There is a computational difficulty with evaluating any set of customer delivery combinations and this can be illustrated by the following question: "suppose I give you two sets of customer delivery combinations and ask you to choose the better set—how can you choose?"

The better set of delivery combinations is the set that has the lowest total routing distance across the period (all days having a feasible routing). To decide the total distance for any set of customer delivery combinations we need to solve the VRP for each of the T days in the period *optimally*. This is computationally very difficult to do for even a small (say 20–30) number of customers per day. Accordingly all we can do is use the same (single-day VRP) *heuristic* algorithm on each set of delivery combinations and choose the combination set that gives the lowest overall routing distance (all days feasible). Note that this choice might differ from the choice that we would have made if we could have solved the daily VRP's optimally.

We mentioned before that one way of solving the T daily VRP's associated with each set of customer delivery combinations would be to use the Fisher and Jaikumar [6] algorithm. Since, in this paper, we were primarily interested in comparing the quality (in routing terms) of the customer delivery combinations chosen by our algorithm with those chosen by the algorithms given in [2] we decided to adopt the same approach as used in [2] to solving daily VRP's. In that paper we generated many solutions for each daily VRP, and improved these solutions using a sophisticated interchange algorithm. For tightly constrained VRP's routes were also constructed manually.

In the next section we report on the computational performance of our algorithm for the PVRP on test problems drawn from the literature.

COMPUTATIONAL RESULTS

The algorithm for the PVRP presented in this paper was programmed in FORTRAN and run on a CDC 7600 using the FTN compiler with maximum optimisation. The LP relaxation of the PVRP program was solved using the APEX-III linear programming package.

Table 1 (from [2]) gives details of the problems solved, all of which were taken from Christofides and Beasley [2] being derived from problems in Eilon *et al.* [4] and Russell and Igo [10]. Note here that to solve the problem from Russell and Igo [10] we had to change our algorithm to cope with the situation where the amount delivered to a customer depends upon the day delivery is made (a simple modification).

Table 2 shows the results of the heuristic algorithm. In that table we give, for each problem, the total number of variables

$$\left(\sum_{i=1}^{n} |S_i|\right)$$

in the PVRP program, the number of fractional variable values in the solution to the LP relaxation of the PVRP program (final iteration) and the total time in CDC 7600 seconds. We also give, for each problem, the total routing distance across the period (derived as described above) for the final set of customer delivery combinations as decided by our algorithm and the best total routing distance across the period as reported in [2] for the Christofides and

Table 1. Problem details

Problem number	Number of customers n	Eilon *et al.* [4] problem number	Length of period T (days)	Vehicles each day (K)	Combination details
50b	50	8	5	3	demand $\leq$ 10 one delivery in period 11 $\leq$ demand $\leq$ 25 (10100) *or* (01010) *or* (00101) demand $\geq$ 26 delivery every day
75b	75	9	5	6	demand $\leq$ 15 one delivery in period 16 $\leq$ demand $\leq$ 27 (10100) *or* (01010) *or* (00101) demand $\geq$ 28 delivery every day
100b	100	10	5	5	demand $\leq$ 10 one delivery in period 11 $\leq$ demand $\leq$ 25 (10100) *or* (01010) *or* (00101) demand $\geq$ 26 delivery every day
100d	100	10	5	4	demand $\leq$ 10 one delivery in period 11 $\leq$ demand $\leq$ 25 (10100) *or* (01010) *or* (00101) demand $\geq$ 26 (10101) *or* (01011) *or* (11010)
Russell	126	—	5	4	See Russell and Igo [10] and section 7 of [2]

Note: (01010) means no delivery on the first day of the period, delivery on the second day, no delivery on the third day etc.

Beasley algorithms where the underlying daily VRP's are replaced by

(1) a median problem; and
(2) a travelling salesman problem (TSP)

together with the total time in CDC 7600 seconds for the Christofides and Beasley algorithms. Note here that all the times given exclude the time to generate the routes for the daily VRP's (in order to evaluate the chosen delivery combinations).

Examining Table 2 we see that the optimal solution to the LP relaxation of the PVRP program is often integer or nearly all-integer (as expected) and also that the computation times are reasonable. Turning to the total routing distance the computational results show that, over all five problems attempted, the algorithm presented in this paper is roughly competitive with the algorithm of Christofides and Beasley based on a median problem. Note however that the result for the Russell problem indicates that as both the number of customers and, more critically, the amount of customer combination choice

$$\left(\sum_{i=1}^{n} |S_i|\right)$$

in the problem increase, the algorithm presented in this paper becomes very competitive with the algorithm of Christofides and Beasley based on a median problem.

It is also clear from Table 2 that the algorithm of Christofides and Beasley based on a travelling salesman problem produces a lower total routing distance, but at the expense of a much higher computation time. This tradeoff between quality of result and computer time incurred is common with heuristic algorithms.

Overall then we conclude that the heuristic algorithm presented in this paper for the PVRP is a computationally inexpensive way of solving the problem which leads (in terms of underlying total routing distance) to a reasonable set of customer delivery combinations.

CONCLUSIONS

In this paper we considered the period vehicle routing problem (PVRP) and extended the

Table 2. Computational results

Christofides and Beasley problem number	n	Number of variables	Number of fractional values		Routing distance (Time CDC 7600 seconds)	Christofides and Beasley algorithm Best routing distance (Time CDC 7600 seconds): Median problem	TSP problem
50b	50	170	—		1481.3 (2.2)	1487.6 (1.0)	1443.1 (2.7)
75b	75	263	2		2192.5 (3.6)	2207.9 (1.5)	2187.3 (25.2)
100b	100	352	—		2281.8 (4.1)	2294.2 (4.8)	2153.3 (63.3)
100d	100	380	2		1833.7 (4.2)	1819.2 (3.9)	1674.0 (26.7)
Russell	126	665	2		878.5 (6.0)	883.4 (16.8)	847.3 (73.4)
				Total	8667.8 (20.1)	8692.3 (28.0)	8305.0 (191.3)

heuristic algorithm of Fisher and Jaikumar, designed for the daily vehicle routing problem, to the PVRP. Computational results for test problems drawn from the literature indicated that the algorithm developed is a computationally inexpensive way of obtaining reasonable quality results.

REFERENCES

1. Beltrami E and Bodin L (1974) Networks and vehicle routing for municipal waste collection. *Networks* **4,** 65–94.
2. Christofides N and Beasley JE (1984) The period routing problem. *Networks* **14,** 237–256.
3. Clarke G and Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Ops Res.* **12,** 568–581.
4. Eilon S, Watson–Gandy CDT and Christofides N (1971) *Distribution Management: Mathematical Modelling and Practical Analysis.* Griffin, London.
5. Fisher ML, Greenfield AJ, Jaikumar R and Lester JT (1982) A computerised vehicle routing application. *Interfaces* **12**(4), 42–52.
6. Fisher ML and Jaikumar R (1981) A generalised assignment heuristic for vehicle routing. *Networks* **11,** 109–124.
7. Lin S and Kernighan BW (1973) An effective heuristic algorithm for the travelling salesman problem. *Ops Res.* **21,** 498–516.
8. Mole RH (1979) A survey of local delivery vehicle routing methodology. *J. Opl Res. Soc.* **30,** 245–252.
9. Russell RA (1977) An effective heuristic for the m-tour travelling salesman problem with some side conditions. *Ops Res.* **25,** 517–524.
10. Russell RA and Igo W (1979) An assignment routing problem. *Networks* **9,** 1–17.
11. Watson–Gandy CDT and Foulds LR (1981) The vehicle scheduling problem: a survey. *NZOR* **9,** 73–92.

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