Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Waste Management 24 (2004) 711–721
waste management
www.elsevier.com/locate/wasman

The investigation of a class of capacitated arc routing problems: the collection of garbage in developing countries

S.K. Amponsah, S. Salhi *

Management Mathematics Group, School of Mathematics and Statistics, The University of Birmingham, Edgbaston, Biriminghan B15 2TT, UK

Accepted 5 January 2004

Abstract

The collection, transport and disposal of solid waste, which is a highly visible and important municipal service, involves a large expenditure but receives, scant attention. This problem is even more crucial for large cities in developing countries due to the hot weather. A constructive heuristic which takes into account the environmental aspect as well as the cost is proposed to solve the routing aspect of garbage collection. This is based on a look-ahead strategy which is enhanced by two additional mechanisms. Interesting results were obtained when tested on instances with and without the presence of the effect of the environment.

1. The problem and its impact on the environment

Collection of household refuse/industrial waste is one of the most difficult operational problems faced by local authorities in any large city. The collection problem is especially crucial for cities in developing countries. This research stems from the need to address this problem for cities in Ghana where on one hand the weather is hot and on the other hand there is unfortunately not enough resources available. These two factors when combined together makes the problem more significant from an environmental and health viewpoints and also more challenging for research.

Refuse characteristics depend on a number of factors such as food habits, cultural traditions, socio-economic and climatic conditions. The organic matter in solid waste in developing countries is much higher than that in the waste from developed countries (Bhide and Sundaresan, 1983). The organic matter is found to be higher due to the use of fresh unprocessed vegetables and has a high moisture content. This large fraction tends to decompose at a faster rate and at the higher ambient temperatures encountered. Table 1 shows the constituents of the refuse in Accra, the capital city of Ghana (Holmes, 1984).

Solid wastes generated from urban and industrial sources also contain a large number of ingredients, some of which are toxic. Various tests and criteria have been devised by different agencies to determine whether a given substance is toxic or hazardous. The possible toxic effect depends upon the quantity of the waste (Bhide and Sundaresan, 1983). The collection of the waste generated in individual premises and that generated on the streets is the responsibility of the local authorities. Collection of streets is done by manual labour and is deposited in storage bins where vehicles collect them for disposal. One of the most important aspects of solid waste management in developing countries is related to the problem of effective storage in generating premises (Oluwande, 1984). The appropriate method for waste collection will depend on the properties and composition of the waste.

The problem of the optimization of the refuse collection vehicle tours for developing countries can be described as follows. The refuse is located in containers along the streets and they must be all collected by a fleet of vehicles whose capacity cannot be exceeded. However, large institutional sites such as schools, hospitals, and

* Corresponding author. +Tel.: +44-121-414-6602; fax.: +44-121-414-3389.
E-mail address: s.salhi@bham.ac.uk (S. Salhi).

0956-053X/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.wasman.2004.01.008

Table 1
Refuse analysis – city of Accra, Ghana

	Cantonment areas house-to-house collection (%)	Market refuse – new town market (%)	Public dumping area – Keneshie (%)	Accra composite (%)
Vegetable and putrescible	69.0	96.4	90.9	87.1
Paper	15.5	2.1	3.0	5.7
Metal	7.0	0.01	2.2	2.6
Textile	1.9	0.59	1.1	1.2
Glass	1.5	0.01	0.7	0.7
Plastic	3.7	0.04	0.5	1.3
Miscellaneous	1.4	0.93	1.6	1.4
Total	100	100	100	100

apartment complexes have the refuse stored in large metal/rubber containers. Each vehicle can service several such sites before going to the dump site to unload. The vehicle leaves the depot at the start of the day and must return there empty at the end of the day. The first trip begins where the vehicle collects refuse along the streets. When the vehicle is full it goes directly to the dump site located at some distance from the city to unload the refuse. The vehicle then goes back to the city to begin its second trip. After completing its final trip, the vehicle proceeds to the dump site to unload for the last time, and finally return to the depot. Each tour is a sequence of two or more trips. One feature of the situation in the urban areas is that there are two or more dump sites from which to choose to empty the vehicle, which are distributed round the city. The depot is not generally a dump site.

The remainder of this paper is organised as follows. Section 2 gives a brief review of previous work. The proposed heuristic is presented in Section 3 and our computational results are provided in Section 4. Our findings are summarized in Section 5.

2. Related work

The waste collection problem can be modelled as the Capacitated Arc Routing Problem (CARP) (Gelders and Cattrysse, 1991). The mathematical formulation of the CARP is given in Appendix A. As this problem can not be solved by optimal (exact) methods in practice, heuristics are used for this purpose.

One possible approach is first to find a giant tour and then decompose it into a set of routes which are feasible with regard to the vehicle capacity. This route first-cluster second approach was used by Beltrami and Bodin (1974) and also Ulusoy (1985). Mourão (2000) developed a similar approach for solving a refuse collection problem except that the transformation of a solution (usually infeasible) is obtained with a lower-bounding method.

Another commonly used approach is the path-scanning procedure, which is based on the construction of one cycle at a time using a certain myopic optimization criterion. In forming a cycle, an edge that looks most promising is added until vehicle capacity is exhausted; then the least cost path to the depot is followed to form a complete cycle. This procedure, is developed by Golden et al. (1983). Pearn (1991) proposed a random path-scanning algorithm which is a variation of the path-scanning algorithm whereas Gelders and Cattrysse (1991) used an interactive decision tool based on this algorithm.

A two-phase approach which considers a set of points for the collection of routes rather than the arcs making up the streets is given by Kulcar (1996).

Christofides (1973) developed construct-strike algorithm for the solution of the undirected Capacitated Chinese Postman Problem (CCPP). The basic idea of this algorithm is to construct feasible vehicle routes which, when removed, do not separate the graph in disconnected components (not counting isolated vertices). When such a route C is constructed, all the required edges on C are removed from the original graph, and this procedure is repeated until no feasible vehicle routes can be determined. Pearn (1989) has proposed a modified version of this algorithm where feasible cycles are constructed and removed repeatedly until no more cycles can be found.

Golden et al. (1983) also developed Augment-Merge algorithm whose idea is to first construct shorter cycles which are then merged into larger ones based on the savings.

Insertion procedures for the Travelling Salesman Problem (TSP) are used by Chapleau et al. (1984) to develop their heuristic known as parallel-insert. Pearn (1991) has also proposed two versions of this augment-insert procedure which were found to be particularly efficient especially on sparse graphs with low density.

Eglese and Murdock (1991), in their solution to 'Routeing Road sweepers in a Rural Area', designed a heuristic to produce feasible routes, with little dead distance as possible in using some priority order for selecting the next road to be swept. Alvarez-Valdes et al. (1993) modified the problem by allowing some containers to be moved by the workers to the nearest junction without the need for the vehicle to traverse the corresponding street segments.

An efficient implementation of tabu search, called CARPET, for the undirected CARP is given by Hertz et al. (2000). It was found that CARPET outperforms all known heuristics often producing a proven optimum. Greistorfer (2002) used tabu scatter search to solve the Capacitated Chinese Postman Problem. Its algorithmic backbone is the tabu search procedure, which is extended by a commonly shared pool of elite solutions.

3. Methodology

In this section, we present an efficient constructive heuristic for the routing problem of the collection of garbage for developing countries with hot weather.

We consider that the sectors to be covered by each vehicle in the working period are already determined by the Local Authorities concerned. This problem of sectorization has obviously an effect on the collection and will be investigated as a future work. The problem now is not only to find the minimum deadheading cycles through all the required edges but also to consider the effect of the bulk uncollected waste on the community at large. There is the need to construct routes along street segments with maximum load at each junction of the collection process. For instance if large quantities of waste are allowed to stay on the street for quite a long time, it will have negative impact on the community. The algorithm works with the combination of two different objective functions: the minimization of the total cost (time or distance) of the operation while taking into consideration the effect of the bulk uncollected waste on the community. The latter is time-dependent, and hence the longer larger quantities are left the worse the situation becomes due to the hot weather. In other words, the objective is to minimize the total cost and the inconvenience due to smell. In our situation we define inconvenience due to smell by the total energy over all arcs where quantities exist.

Let $E_{ij}(t) = q_{ij}(t_{ij} - t_0)$ be the energy on edge/arc (i,j) at time t_{ij}, where t_0 is the starting time for the collection of the garbage, t_{ij} is the time when the street segment (i,j) is serviced, and q_{ij} is the quantity of garbage to be collected along the edge (i,j).

We use two costs namely a collection cost and empty movement cost (traversing a street without collection of garbage), and information as to whether edge (i,j) is a directed or undirected edge.

3.1. The proposed look-ahead strategy

3.1.1. Basic idea

Servicing a street segment depends on the quantity q_{ij} along that street and the cost c_{ij} of servicing it. The larger the quantity on a given edge, the greater its environmental impact is. However, such an edge may be too costly to revisit again empty. One way is to combine both elements into the formulation. A weight factor, say α, is assigned to the quantities and $(1-\alpha)$ to the cost where $0 \leqslant \alpha \leqslant 1$. The following weighted function (Eq. (1)) is proposed. In other words, we want to minimize both the environmental effect (quantity to be removed as early as possible) and the total cost while emphasizing their individual impacts

$$\delta_{ij} = \frac{(q_{ij})^{\alpha}}{(c_{ij})^{1-\alpha}}, \quad 0 \leqslant \alpha \leqslant 1. \tag{1}$$

In our first experiments we shall analyze the effect of α by setting $\alpha = 0.0, 0.25, 0.50, 0.75$ and 1.00. Let i be the first junction from the depot where the collection of garbage begins: The step by step description of our first implementation is given in Fig. 1. This variant is also used as a basis for the next procedure.

This approach is myopic in that an edge may have a maximum demand/cost ratio along it but the subsequent edges may not be promising (smaller ratios may be encountered). One way to improve this greedy method is to introduce a look-ahead strategy by taking into account present as well as later choices in the algorithm.

3.1.2. A look-ahead strategy

The idea of the look-ahead strategy is to examine the total demand/cost ratio on all possible temporary edges with respect to their likelihood to yield future advantage to prune away unpromising edges in the collection process and to choose edges that are most promising.

The algorithm proceeds from one junction (node) to one of its adjacent nodes at each stage.

Step 1 For each edge or arc emmanating from junction i, compute

$$\delta_{ij} = \frac{(q_{ij})^{\alpha}}{(c_{ij})^{1-\alpha}}, \qquad 0 \le \alpha \le 1. \tag{1}$$

Step 2 Find (i, j_m) where $j_m = arg[\max_j\{\delta_{ij}\}]$, service edge (i, j_m) and set $i = j_m$.

Step 3 Repeat steps 1 and 2 until all required edges are serviced.

Step 4 If there exists an edge (i, j_d) $\quad (d = 1, 2 \ldots, k)$ which is a dead end and $q_{ij_d} \ge 0$, service edge (i, j_d) and then return to i to continue service.

Fig. 1. Initial implementation of quantity and cost-dependent routing.

Let j_c^n be the current node (junction) under consideration, and $j_1^{n+1}, j_2^{n+1}, \ldots, j_r^{n+1}$ be the (r) nodes adjacent to j_c^n.

Using our first implementation, for every $m \leqslant r$,

$$j_m^{n+2} = arg[\max_b \{\delta_{j_m^{n+1} j_{b=1,2,\ldots,u_m}}\}],$$

where u_m is the number of nodes adjacent to j_m^{n+1}. We consider r temporary chains, each of length two (i.e., $(j_c^n, j_1^{n+1}, j_1^{n+2})$, $(j_c^n, j_2^{n+1}, j_2^{n+2}) \ldots$, $(j_c^n, j_k^{n+1}, j_k^{n+2}) \ldots$, $(j_c^n, j_r^{n+1}, j_r^{n+2})$) emanating from j_c^n. The temporary chain to be considered is the one with maximum total demand/cost ratio, say $(j_c^n, j_1^{n+1}, j_1^{n+2})$, then the edge (j_c^n, j_1^{n+1}) is the next edge to be serviced. Although the temporary chains help us to select the 'best' chain, in the collection of refuse, we do not follow the chain to the end for the collection. We then replace j_c^n by j_1^{n+1} and the process is repeated until all required edges are serviced. The step by step procedure is given in Fig. 2.

3.1.3. Illustrative example for the look-ahead strategy

We illustrate this technique through a numerical example as shown in Fig. 3. Let the capacity of the vehicle $W = 50$ and node 1 denote the starting node (depot node). We compute the total demand/cost ratio for all the temporary chains of length two emanating from node 1. These are $(1, 6, 5)$, $(1, 5, 8)$ and $(1, 2, 3)$ with corresponding total demand/cost ratio of 4.5, 4.6 and 5.0, respectively. The temporary chain $(1, 2, 3)$ has the maximum total demand/cost ratio and so the edge $(1, 2)$ of this chain is serviced. At node 2, all the temporary chains of length two emanating from it are considered. These are $(2, 4, 5)$ and $(2, 3, 8)$ with corresponding total demand/cost ratio of 3.0 and 5.8, respectively. In this case the edge $(2, 3)$ is chosen. This process is continued until all the demand edges are serviced.

3.1.4. Time-dependent model

In this section we extend Eq. (1) to cater for practical events that exist especially in developing countries due to the hot weather. In other words we make α time-dependent as the smell gets worse and worse with time. The emphasis on the environmental factors is given in Section 1.

In this new development α is a monotonic non-decreasing function of time. In other words α of Eq. (1) is replaced by $\alpha(t)$ in Eq. (2). We define

$$\delta_{ij} = \frac{(q_{ij})^{\alpha(t)}}{(c_{ij})^{1-\alpha(t)}}, \quad 0 \leqslant \alpha(t) \leqslant 1. \tag{2}$$

At the start of the collection process, early in the morning, $\alpha(t)$ is given a low value, and this is increased

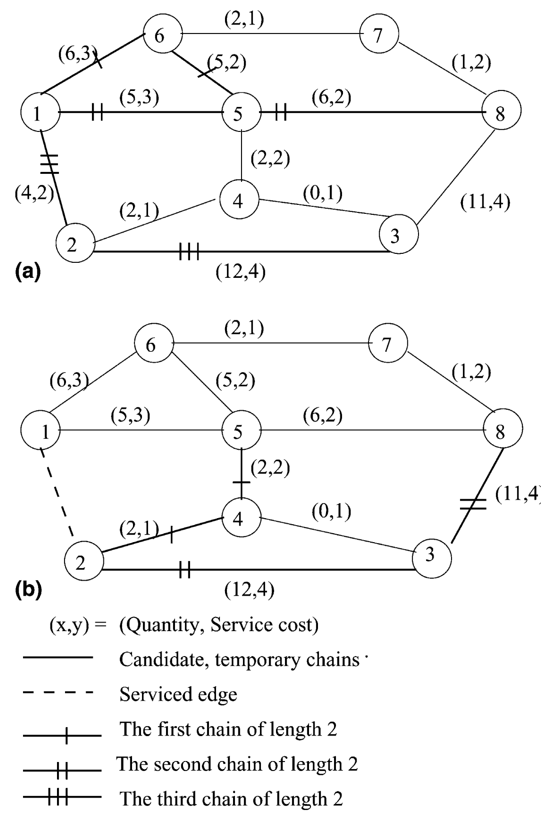

Fig. 3. Collection of garbage using look-ahead strategy.

Step 1 At each stage, consider all the number (r) of edges $(j_c^n, j_{m=1,2,\ldots,r}^{n+1})$ emanating from the current node j_c^n.

Step 2 If there exists an edge (j_c^n, j_d^{n+1}), where $d \leq r$, which is a dead end and $q_{j_c^n j_d^{n+1}} > 0$, service edge (j_c^n, j_d^{n+1}) and then return to j_c^n to continue service.

Step 3 For each j_m^{n+1}, find $j_m^{n+2} = arg[\max_b\{\delta_{j_m^{n+1} j_{b=1,2,\ldots,u_m}}\}]$ where u_m is the number of nodes adjacent to j_m^{n+1}; $m = 1, 2, \ldots, r$.

Step 4 Find $\max_{m=1,2,\ldots,r} \{\delta_{j_c^n j_m^{n+1}} + \delta_{j_m^{n+1} j_m^{n+2}}\}$, say $\{\delta_{j_c^n j_1^{n+1}} + \delta_{j_1^{n+1} j_1^{n+2}}\}$, and collect garbage along the edge (j_c^n, j_1^{n+1}).

Step 5 Set $j_c^n = j_1^{n+1}$.

Step 6 Go to step 2 until all required edges are serviced.

Fig. 2. A look-ahead strategy.

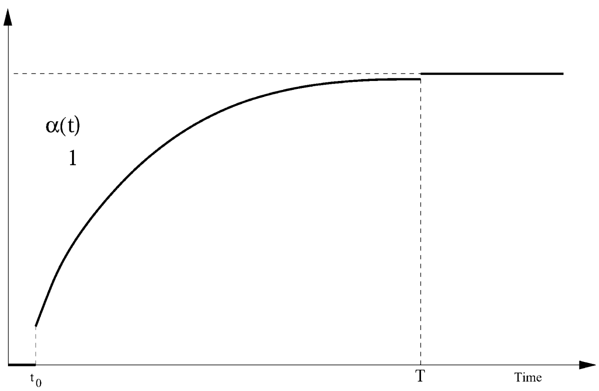

Fig. 4. $\alpha(t)$ vs. time.

with respect to time according to (3) until it reaches the upper limit of 1. This brings to light the idea of servicing the street segment with the maximum quantity at each junction. $\alpha(t)$ can be expressed analytically as in Eq. (3) and shown graphically as in Fig. 4.

$$\alpha(t) = \begin{cases} 0, & t < t_0, \\ 1 - e^{-t}, & t_0 \leqslant t \leqslant T, \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

The implementation of $\alpha(t)$ as given in (3) can be computationally time consuming and as such an expression is just an approximation, one way forward is to propose a simpler form of $\alpha(t)$, which can be easier to implement. One way is to discretize $\alpha(t)$ as shown in the following equation:

$$\alpha(t) = \begin{cases} 0, & t < t_0, \\ 0 + \epsilon, & t_0 \leqslant t < t_1, \\ 0 + 2\epsilon, & t_1 \leqslant t < t_2, \\ 0 + 3\epsilon, & t_2 \leqslant t < t_3, \\ \ldots, & \ldots, \\ \ldots, & \ldots, \\ \ldots, & \ldots, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

Note that ϵ is a non-negative real constant which can be given.

3.1.5. Weakness of the look-ahead strategy

The adopted strategy does not take into account the current load of a vehicle. Note that when the vehicle is nearly full more care is needed. If we apply the look-ahead strategy without guidance, the vehicle may not get to the end of the chain as planned since at this stage, there is the need to minimize the shortest distance (cost) from the last junction to the dump site. This weakness can be remedied by applying the least insertion rule at this stage. This problem can be illustrated diagrammatically as shown in Fig. 5.

Let junction 11 be the current junction under consideration in the collection process. At this point chains (11,18,19,20), (11,17,16,15) and (11,12,13,14) will be considered when using the look-ahead strategy with chains of length three and the chains (11,18,19), (11,17,16) and (11,12,13) will be considered when using the look-ahead strategy with chains of length two. In both cases the vehicle will move away from the dump site thus moving along edge (11,12) or (11,17) even though the vehicle is nearly full. At this stage in the collection process it is appropriate to force the vehicle to move towards the dump site by using the least insertion cost criterion; thus we choose the edge (11,18) instead.

The idea of using two criteria instead of one is not new. For instance, Gelders and Cattrysse (1991) suggested that the following combination of two criteria leads to good results. Given that a provisional path (start node = 1) arrives in node i, the authors added arc (i, j) satisfying one of the following criteria:

- If the truck load is less than 50%, maximize the shortest distance from j to the end node including the distance of arc (i, j).
- Else minimize the shortest distance from j to the end node including the distance of arc (i, j).

In this study we relax the 50% cut off point by identifying the switching point between the two criteria in a more guided way. This enhancement is described below.

3.1.6. A switching mechanism

If the quantity of garbage collected so far in a particular trip is more than or equal to a critical value $\ddot{Q}$, which is based not only on the vehicle capacity but on the average load on the routes, we shift from the look-ahead strategy to the least insertion cost rule. In this situation, at the current junction i, we compute $\min(c_{ij} + c_{jd})$, the minimum sum distance (cost) value of moving along the edge (i, j) and (j, d), where d is the dump site. If $\min_j(c_{ij} + c_{jd}) = (c_{ij_*}, c_{j_*d})$, the edge (i, j_*) is serviced and then we set $i = j_*$ to continue the process using the least distance rule. The main steps of the switching mechanism are given in Fig. 6.

3.1.7. A refinement scheme

This section provides a constructive heuristic that aims at improving the solutions obtained by the use of the look-ahead strategy when $\hat{q} = \bar{q}$ (i.e., $\lambda = 0$) in step 3 of Fig. (6). A single tour covering all the required edges is created by merging all the n trips obtained by use of the look-ahead strategy. All the deadheading edges in this tour are then removed leaving a set of disconnected chains. New trips (routes) are then created by modifying the order in which the required edges are visited subject to the capacity constraint of the vehicle with the object of cutting down the total travelling cost and the effect of the environment. In this case servicing a required edge in an opposite direction to its original direction of service is allowed. For computational efficiency, a matrix of the least cost elements, which connects nodes of deadheading edges, is stored.

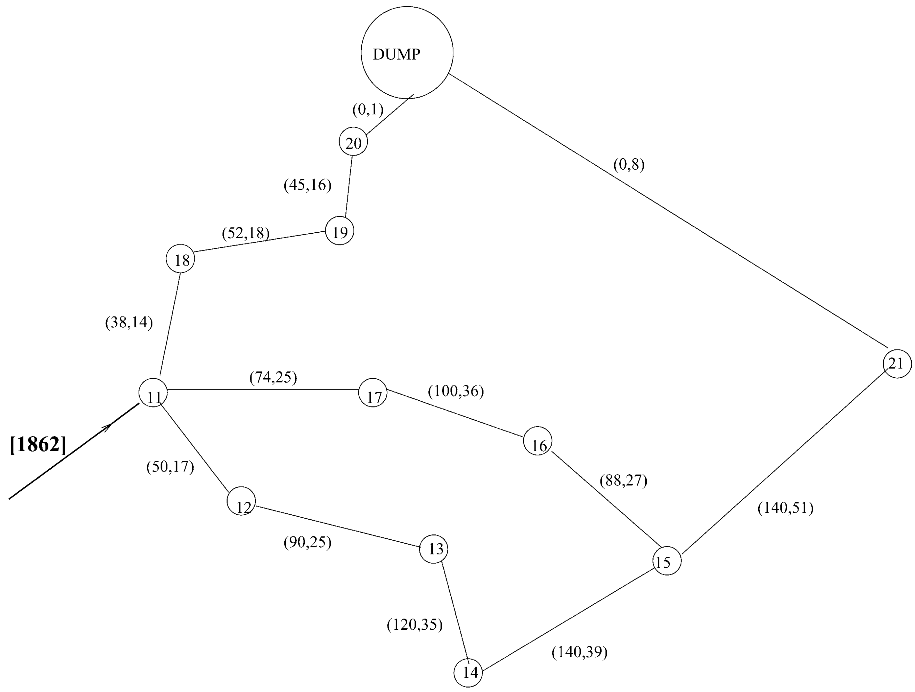

Fig. 5. Need for insertion at a late stage of the trip.

Step 1 Compute the average quantity of garbage along the edges/arcs as $\bar{q} = \dfrac{\sum q_{ij}}{|E|}$.

Step 2 Determine the variance of garbage along each edge/arc of the network as $S^2 = \frac{1}{|E|-1}[\sum {q_{ij}}^2 - |E|\bar{q}^2]$.

Step 3 Evaluate $\hat{q} = \bar{q} + \lambda S$, where $\lambda \in \{0, 1, 2\}$.

Step 4 Compute the correction factor as $\rho = 1 - \dfrac{\hat{q}}{W}$ (W is the vehicle capacity).

Step 5 Find the critical value for switching as $\ddot{Q} = \rho.W \qquad 0 \leq \rho \leq 1$.

Fig. 6. A switching mechanism.

The step by step procedure of the refinement scheme is given in Fig. 7.

3.1.8. Illustrative example

Consider the network depicted in Fig. 8(a) with depot node 1. The numbers indicated along the edges are the quantity of garbage. Servicing and deadheading costs of each edge of the network are 2 units and 1 unit, respectively. Let the vehicle's capacity $W = 8$. Using the look-ahead strategy with the insertion mechanism, we obtain an initial solution as given in Fig. 8(b) with total travelling cost of 22 units. $T1, T2$ and $T3$ are the first, second and third trips, respectively. Applying the refinement scheme, the edges can be visited as shown in Fig. 8(c) with total travelling cost of 20 units, which happen to be optimal as this can be shown using complete enumeration. The edges in Fig. 8(b) and (c) are broken down into two classes. A deadheading edge is indicated by a dotted line and a smooth line indicates an edge needing servicing.

Table 2 displays the tours, their corresponding cost and the amount of smell constructed by the refinement mechanism. $a \rightarrow b$ indicates that edge (a, b) needs servicing and $c \dashrightarrow d$ indicates that (c, d) is a deadheading edge.

Tour number 6 is one of the optimal solutions obtained by use of the refinement mechanism as indicated in Fig. 8(c) and tour 2 is also optimal in terms of cost and has a smaller smell value as compared to tour 6. Optimality can be achieved by complete enumeration. In this case, tour 2 can be considered to be the 'best' with respect to both the cost and the effect of the environment. Again tour 3 is not an optimal solution in terms of cost, but has a smaller smell value as compared to the

Step 1: An initial schedule is constructed and all disconnected chains recorded.

Step 2: The positions of the depot node in the set of disconnected chains are identified. Set the total number of the depot node found to k.

Step 3: Set $p = 1$, i.e., the first edge containing the depot node and let $Y(p)$ denote the position of the p^{th} depot node in the disconnected set of chains.

Step 4: Denote by $S(p)$, the $(Y(p)+1)^{th}$ node. Scan through the set of disconnected chains and find the positions of all nodes equal to $S(p)$. Denote by $f(p)$, the frequency of $S(p)$.

Step 5: Set $r = 1$. Starting from the p^{th} depot node, a connected route of required and non-required edges is built by adding an edge/arc at a time, until the vehicle capacity is exhausted.

Step 6: Set $r = r + 1$. If $r \leq f(p)$ go to step 5 else set $p = p + 1$ and go to step 4.

Step 7: If $p > k$ Stop.

Fig. 7. A chain-based insertion.

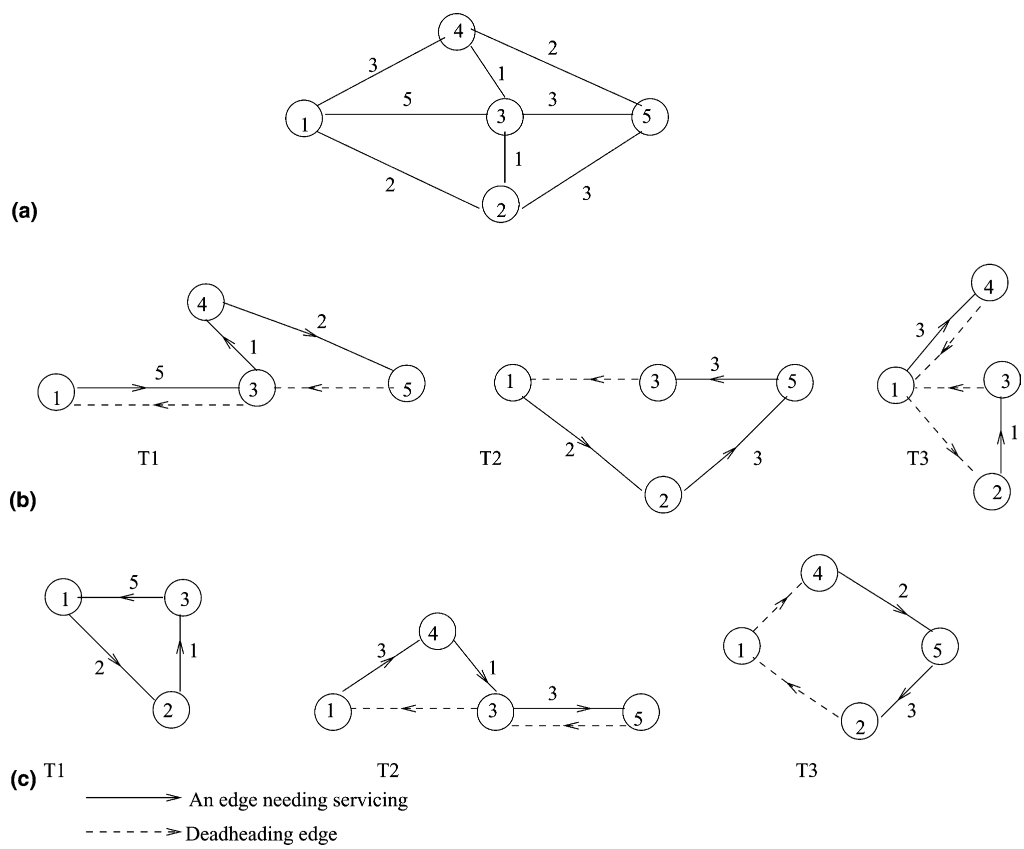

Fig. 8. Illustration of the refinement scheme.

corresponding smell value of tour 2. The question is which tour shall we use in order to minimize cost as well as the effect on the environment? There is no 'right' answer but we hope the different scenarios (possible solutions) will assist the decision-maker in selecting the answer that suits his/her problem best.

4. Computational results

The proposed heuristic is assessed using two experiments, both tested on the 23 instances from DeArmon (1981). There were initially 25 such instances, but data sets 8 and 9 were removed because they contain inconsistencies. The size of these problems range from 7 to 27 vertices and from 11 to 55 edges. In the first experiment, we test our heuristic using both cost and the effect of the smell. We provide the results of three different scenarios using $\beta = 0.2, 0.5$ and 0.8. For completeness we also carried out a second experiment by ignoring the effect of the environment. In this experiment the proposed heuristic is compared against existing methods. The proposed heuristic is coded in Fortran programming language and compiled using $f90 - cg92 - O1$ optimizer.

Table 2
Tour, cost and smell

No.	Trip 1	Trip 2	Trip 3	Cost	Smell
1	1 → 3 → 4 → 5--→3--→1	1 → 2 → 5 → 3--→1	1--→2 → 3--→1 → 4--→1	22	24.75
2	1 → 3 → 2 → 1	1 → 4 → 3 → 5--→3--→1	1--→4 → 5 → 2--→1	20	23.38
3	1 → 3 → 5--→3--→1	1 → 2 → 5 → 4 → 3--→1	1 → 4--→3 → 2--→1	21	22.13
4	1 → 3 → 4 → 5--→3--→1	1 → 2 → 5 → 3--→1	1 → 4--→3 → 2--→1	21	24.38
5	1 → 3--→1 → 2 → 3--→1	1 → 4 → 3 → 5--→3--→1	1--→4 → 5 → 2--→1	23	28.38
6	1 → 2 → 3 → 1	1 → 4 → 3 → 5--→3--→1	1--→4 → 5 → 2--→1	20	24.88
7	1 → 2 → 5 → 4 → 3--→1	1 → 3 → 5--→3--→1	1 → 4--→3 → 2--→1	21	25.13
8	1 → 2--→1 → 3 → 4--→1	1 → 4 → 5 → 2--→1	1--→2 → 3 → 5--→3--→1	23	28.50
9	1 → 4 → 3 → 5--→2 → 3--→1	1 → 3--→2 → 5--→3--→1	1 → 2--→5 → 4--→1	23	28.38
10	1 → 4 → 5 → 2--→1	1 → 3 → 4--→1 → 2--→1	1--→2 → 3 → 5--→3--→1	22	25.88
11	1 → 4--→1 → 3--→1	1 → 2 → 5 → 3--→1	1--→3 → 4 → 5--→2 → 3--→1	23	25.38

4.1. Experiment 1: multi-objective route-planning

In multi-objective routing problems, it is usually necessary to generate a set of routes and evaluate them under the relevant criteria. In addition to the multi-objective scenario, there may be instances, where a decision-maker is interested in developing backup routes daily in case the best route becomes infeasible due to road construction or a major accident. In our implementation of multi-objective route planning where the total travelling cost and the effect of the environment are considered, the total travelling cost and the average smell associated with each route were computed. We denote by H_i and E_i the total travelling cost and the average smell value, respectively, for the ith route and N be the number of different routes created. We compute $\bar{E} = (\sum E_i)/N$. The E_i's are then normalized using the formula $F_i = (E_i/\bar{E})H_i$. The cost and the normalized values of the smell are then weighted by $W_i = \beta H_i + (1-\beta)F_i$. Different values of β are used to put emphasis on the cost and the smell. For instance $\beta = 0.2$ favours the environment, whereas $\beta = 0.8$ favours the cost.

The solutions obtained by using the data by DeArmon (1981) are summarized in Table 3. Information about the data set are reported in the first five columns of Table 4.

The detailed results showing the effect of the switching mechanism and that of the refinement are not reported here for simplicity but can be found in Amponsah (2003). In this experiment we used discretized values of α as given in Eq. (4) with $\epsilon = 0.25$, t_0, the initial time and $\alpha = 0.0, 0.25, 0.50, 0.75$ and 1 and we record the corresponding α that yields the best results. For instance as depicted from Table 3, for problem number 21, $\beta = 0.2$, and $W_i = 61.83$ and for $\beta = 0.8$, $W_i = 60.39$ but the cost and smell values computed for $\beta = 0.2$ are 63 and 391.91 units, respectively. The corresponding cost and smell values computed for $\beta = 0.8$ are 59 and 498.00 units, respectively. In this case, decision-makers are left to choose the solution that will suit them best using a pool of solutions.

Table 3
Effect of β on cost and smell on DeArmon instances

Problem No.	α	W_i		
		$\beta = 0.2$	$\beta = 0.5$	$\beta = 0.8$
1	1.00	328.52	328.50	328.88
2	0.50	366.58	366.36	366.15
3	0.25	291.67	293.67	295.67
4	0.50	312.16	313.23	313.47
5	0.00	407.87	409.59	410.44
6	0.00	313.21	318.38	323.55
7	0.75	323.38	329.24	335.09
10	0.00	385.37	393.48	401.59
11	0.50	351.16	352.23	353.29
12	0.25	267.38	275.49	283.59
13	1.00	432.87	432.54	432.22
14	0.50	594.50	594.53	594.81
15	0.50	530.71	539.44	548.18
16	0.50	105.27	104.79	104.32
17	0.50	57.10	57.44	57.77
18	1.00	130.91	130.94	130.97
19	0.50	94.49	94.68	94.87
20	0.50	170.11	169.32	168.53
21	0.00	61.83	62.47	60.39
22	1.00	125.45	125.66	125.86
23	0.25	155.73	157.33	158.93
24	1.00	197.90	199.44	200.97
25	0.00	243.04	242.28	241.51

4.2. Experiment 2: single objective route-planning

In this experiment the environmental effect is ignored. The proposed algorithm is then compared with the best known heuristics in the literature. The results are summarized in Table 4.

The first five columns of Table 4 give some characteristics of the problem instances: the name, the number of vertices in the network, $|V|$, the number of edges in the network, $|E|$, the total demand Q_T and the vehicle capacity W. Columns (7)–(16) give the solution value produced with

PS: path-scanning heuristic (Golden et al., 1983).

AM: the augment-merge heuristic (Golden and Wong, 1981).

Table 4
Computational results on DeArmon (1981) instances

No.	$\|V\|$	$\|E\|$	Q_T	W	BEST	PS	AM	CS	MCS	MPS	α	LA	CB	CPU	DY
1	12	22	22	5	316	**316**	326	331	323	**316**	1.00	395	329	0.04	389
2	12	26	26	5	339	367	367	418	345	355	0.50	388	366	0.07	374
3	12	22	22	5	275	289	316	313	**275**	283	1.00	368	296	0.05	348
4	11	19	19	5	287	320	290	350	**287**	292	0.50	371	313	0.03	379
5	13	26	26	5	377	417	383	475	386	401	0.50	474	409	0.06	497
6	12	22	22	5	289	316	324	356	315	319	0.00	354	326	0.05	390
7	12	22	22	5	325	357	**325**	355	**325**	**325**	0.75	414	339	0.04	394
10	27	46	249	27	344	416	356	407	366	380	0.00	491	407	0.06	433
11	27	51	258	27	303	355	339	364	346	357	0.50	369	354	0.06	397
12	12	25	37	10	275	302	302	364	**275**	281	0.25	336	283	0.04	366
13	22	45	225	50	395	424	443	501	406	424	0.00	590	432	0.19	573
14	13	23	212	35	450	560	573	655	645	566	1.00	595	577	0.05	595
15	10	28	245	41	536	592	560	560	544	551	0.75	577	554	0.06	649
16	7	21	89	21	100	102	102	112	102	**100**	1.00	112	103	0.05	122
17	7	21	112	37	58	**58**	**58**	**58**	**58**	**58**	0.50	60	**58**	0.42	62
18	8	28	116	24	127	131	131	149	**127**	131	1.0	145	131	0.06	143
19	8	28	168	24	91	93	**91**	**91**	**91**	93	0.50	97	95	0.13	97
20	9	36	153	37	164	168	170	174	**164**	167	0.50	186	168	0.23	217
21	11	11	66	27	55	57	63	63	63	**55**	0.00	67	59	0.02	67
22	11	22	107	27	121	125	123	125	123	123	1.00	127	125	0.07	153
23	11	33	154	27	156	168	158	165	**156**	163	0.25	175	160	0.19	174
24	11	44	205	27	200	207	204	204	**200**	202	1.00	211	201	0.11	208
25	11	55	266	27	233	241	237	237	**233**	244	0.00	252	241	0.55	249
Average deviation in %						7.6	5.9	14.5	4.0	4.8			7.1		
Worst deviation in %						24.4	27.3	45.6	43.3	25.8			28.2		
Number of optima						2	3	2	11	5			1		
Number of best						2	4	2	15	5			1		

CS: the construct-strike heuristic (Christofides, 1973).
MCS: the modified construct-strike heuristic (Pearn, 1989).
MPS: the modified-path-scanning heuristic (Pearn, 1989).
α: a variable.
LA: at the end of step 0 by use of the look-ahead strategy and the insertion mechanism.
CB: the chain-based heuristic proposed in this paper.
Best: best known value,
CPU: total running time (s), and
DY: problem type of dynamic updating of α.

In addition, we compute the following statistical measures, namely

Average deviation: average ratio (in %) of the proposed heuristic solution over the best known solution value.

Worst deviation: largest ratio of the proposed heuristic over the best known solution value.

Number of best: number of best known solutions produced by the proposed heuristic.

The proposed heuristic appears to perform well considering the fact that it is originally designed to handle both aspects namely environment and cost simultaneously. From Table 4, we can see that the refinement scheme significantly improved the initial solutions obtained by use of the look-ahead strategy with the insertion mechanism. It was also found that the refinement scheme improved the solutions for all the 23 test problems. In summary, the proposed algorithm though it is not the best performer, it produces reasonably good results when compared to the Path-Scanning and the Construct-Strike algorithms.

5. Conclusions

A constructive heuristic that considers environmental aspects to solve the waste collection problem for developing countries is designed. A look-ahead strategy, based on a bi-objective model, namely the minimization of both the cost and the effect of the environment is put forward to solve the routing aspect of garbage collection. Enhancement procedures namely a switching mechanism that adoptively shifts to the least cost rule whenever necessary and a refinement procedure are embedded within the look-ahead strategy. Our new heuristic has been shown to be computationally efficient and performs well on all the test problems, besides having the advantage in producing several solutions. This flexibility, which considers both cost and environment could assist local authorities in choosing from the pool of solutions, the one which suits best their need.

Acknowledgements

The authors are grateful to the referees comments which improved the content as well as the presentation of an earlier version of this paper. The first author also would like to thank the Ghana Government for the sponsorship.

Appendix A. Mathematical formulation

Given a connected graph $G = (V, E \cup A)$, with V as the set of nodes (vertices), E set of edges ($E \subseteq V \times V$) and A a set of arcs ($A \subseteq V \times V$), the Rural Postman Problem (*RPP*) is the problem of finding a minimum cost traversal of a given subset of edges and arcs in $R \subseteq E \cup A$. The Capacitated Rural Postman Problem, usually denoted as the Capacitated Arc Routing Problem (*CARP*), has an additional traversal cost for each edge and arc with edge (arc) demand $q_{ij} \geqslant 0$ for each edge (i,j) which must be serviced by one of a fleet of vehicles of capacity W. The problem is to find a number of circuits each of which passes through the depot which satisfy demands at minimal total cost.

We denote c_{ij} as the cost of an edge (arc) $(i,j) \in E(A)$ and x_{ijk} as the number of times edge (arc) $(i,j) \in E \cup A$ is traversed in trip k

$$y_{ijk} = \begin{cases} 1 & \text{if the edge (arc) } (i,j) \in R \text{ is covered in trip } k, \\ 0 & \text{otherwise.} \end{cases}$$

M is a large constant greater than or equal to the sum of traversals of edges and arcs in a given $S \subseteq R$, $V[S]$ is the set of nodes incident to the arc set S, k denotes a trip, and K is the maximum number of trips allowed.

The CARP formulated by Dror and Langevin (2000) is as follows:

$$\min \sum_{(i,j)\in E} \sum_{k=1}^{K} c_{ij} x_{ijk} \tag{A.1}$$

subject to

$$\sum_{p\in V} x_{pik} - \sum_{p\in V} x_{ipk} = 0 \quad \forall i \in V, \quad k = 1, 2 \ldots, K, \tag{A.2}$$

$$\sum_{k=1}^{K} y_{ijk} = 1 \quad \forall (i,j) \in R, \tag{A.3}$$

$$x_{ijk} \geqslant y_{ijk} \quad \forall (i,j) \in R, \quad k = 1, \ldots, K, \tag{A.4}$$

$$\sum_{(i,j)\in R} q_{ij} y_{ijk} \leqslant W, \quad k = 1, \ldots, K, \tag{A.5}$$

$$M \sum_{i\in V[S], j\in V[S]} x_{ijk} \geqslant \sum_{(j,p)\in S} x_{jpk} \quad \begin{cases} \forall S \subseteq R, \\ 0 \in V[S], \\ k = 1, 2 \ldots, K, \end{cases} \tag{A.6}$$

$$\begin{aligned} & y_{ijk} \in \{0,1\} \quad \forall (i,j) \in R, \quad k = 1, 2, \ldots, K, \\ & x_{ijk} \in Z^{+} \quad \forall (i,j) \in E, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{A.7}$$

The objective function (A.1) seeks to minimize the total cost. Eq. (A.2) ensure route continuity. Eq. (A.3) state that each edge with positive demand is serviced exactly once. Eq. (A.4) guarantee that the traversal circuit k covers the edge $(i,j) \in R$ if it delivers its demand. Vehicle capacity is not violated on account of (A.5). Eq. (A.6) prohibit the formation of (infeasible) subtours. Integrality restrictions are given in (A.7)

References

Alvarez-Valdes, R., Benavent, E., Campos, V., Corberan, A., Mota, E., Tamarit, J., Valls, V., 1993. A computerized system for urban garbage collection. Topology 1, 89–105.

Amponsah, S.K., 2003. The Investigation of a class of capacitated arc routing problems: the collection of garbage in developing countries. PhD. Thesis, School of Mathematics and Statistics, University of Birmingham, UK.

Beltrami, E., Bodin, L., 1974. Networks and vehicle routing for municipal waste collection. Networks 4, 65–94.

Bhide, A., Sundaresan B. 1983. Solid waste management in developing countries. Indian National Scientific Documentation Centre, pp. 1–55.

Chapleau, L., Ferland, J., Lapalme, G., Rousseau, J., 1984. A parallel insert method for the capacitated arc routing problem. Operations Research Letters 3 (6), 95–99.

Christofides, N., 1973. The optimal traversal of a graph. Omega 1 (6), 719–732.

DeArmon, J., 1981. A Comparison of heuristics for the capacitated Chinese Postman problem. Master's thesis, University of Maryland, College Park, MD.

Dror, M., Langevin, A., 2000. Transformations and exact node routing solutions by column generation. In: Dror, M. (Ed.), Arc Routing: Theory, Solutions and Applications. Kluwer Academic Publishers, Dordrecht, pp. 277–326.

Eglese, R., Murdock, H., 1991. Routing road sweepers in a rural area. Journal of the Operational Research Society 42, 281–288.

Gelders, L., Cattrysse, D., 1991. Public waste collection: a case study. Belgian Journal of Operations Research Statistics and Computer Science 31, 5–15.

Golden, B., DeArmon, J., Baker, E., 1983. Computational experiments with algorithms for a class of routing problems. Computers and Operations Research 10, 47–59.

Golden, B., Wong, R., 1981. Capacitated arc routing problems. Networks 11, 305–315.

Greistorfer, P., 2002. A tabu scatter search metaheuristic for the arc routing problem. Computer and Industrial Engineering 3, 129–135, Forthcoming.

Hertz, A., Laporte, G., Mittaz, M., 2000. A tabu search heuristic for capacitated arc routing problems. Operations Research 48, 129–135.

Holmes, J., 1984. Solid waste management decisions in developing countries. In: Holmes, J. (Ed.), Managing Solid Wastes in Developing Countries, pp. 1–28.

Kulcar, T., 1996. Optimizing solid waste collection in Brussels. European Journal of Operational Research 90, 71–77.

Mourão, M., 2000. Lower-bounding and heuristic methods for collection vehicle routing problem. European Journal of Operational Research 121, 420–434.

Oluwande, P., 1984. Assessment of solid waste management problems in China and Africa. In: Holmes, J (Ed.), Managing Solid Wastes In Developing Countries, pp. 77–89.

Pearn, W., 1989. Approximate solutions for the capacitated arc routing problem. Computers and Operations Research 16 (6), 589–600.

Pearn, W., 1991. Augment-insert algorithm for the capacitated arc routing problem. Computers and Operations Research 18 (2), 189–198.

Ulusoy, G., 1985. The fleet size and mix problem for capacitated arc routing. European Journal of Operational Research 22, 329–337.