Discrete Applied Mathematics 48 (1994) 231–244
North-Holland

Routeing winter gritting vehicles

R.W. Eglese

Department of Operational Research and Operations Management, Management School, Lancaster University, Lancaster LA1 4YX, UK

Received 15 September 1989
Revised 1 July 1990

Abstract

When roads may become dangerously slippery due to frost, ice or snow, local authorities treat the roads by spreading a de-icing agent (usually salt) on them. In order to treat a road, a winter gritting vehicle must travel down the road once, spreading salt on to both sides of the carriageway. An application is described where routes were constructed for gritters in a local authority area.

The formulation of the model is presented which involves dealing with multiple depot locations, limited vehicle capacities, and roads with different priorities (for example, some must be treated within two hours and others within four hours of the start of gritting). The objective function to be optimised depends on both the total distance travelled, and the number and capacity of the gritters.

The solution method is a heuristic algorithm, which involves, as a first stage, the optimal solution of an unconstrained Chinese Postman Problem for the network, and followed by the use of Simulated Annealing for the constrained problem.

1. Introduction

In 1988 a study was carried out into the winter gritting operation of a local authority. The aim of the study was to find the most cost effective way to carry out this service within operational and policy constraints. A key element of this study was the routeing of the winter gritting vehicles when they were required to treat the roads. Each road requiring treatment needs a gritter to travel down it once, spreading salt on both sides of the carriageway.

This is a type of arc routeing problem. Various other arc routeing problems and solution methods for them may be found in Golden and Assad [6] and Bodin et al. [3].

The problem of finding a single route which covers each arc in an undirected network and returns to the starting point while minimising the distance travelled is the Chinese Postman Problem (CPP). The solution to the CPP may be obtained in polynomial time by using an algorithm due to Edmonds and Johnson [5]. However, the Capacitated Chinese Postman Problem (CCPP), where there is a limit on the total length of arcs served in any single route, is NP-hard [8]. This means that to solve large

SSDI 0166-218X(92)00003-D

problems, heuristic methods are required to produce feasible solutions in a reasonable amount of computing time.

Heuristics and lower bounds have been proposed for the CCPP (e.g. [1, 2, 7]) but in the case studied, there were several constraints and other factors which made the design of suitable routes even less tractable than the CCP. These are discussed in the following sections.

1.1. Multiple depot locations

Our application required that a set of one or more depots be considered as the starting points for vehicle routes serving the roads in one area. A depot is where the gritting vehicles are normally garaged and where they may load with salt. A gritter must return to its home depot on the completion of its route as the men are based there. Therefore routes which started and finished at different depots were not allowed. In fact, one purpose of the winter gritting study was to examine a variety of sets of depot locations and the effect on costs. These results were then used in a wider study to make recommendations on the locations of depots which were used as bases for many other types of work apart from winter gritting.

Although one possibility would be to segment the network of roads into independent areas around each depot, this may lead to poor solutions if it is not done carefully. For example, a network split into three equally sized networks around three depots may need two gritters at each depot to service each part, but it may be possible to use only four gritters in total if one area is made larger than the other two. The formulation presented here therefore considers the routes from all depots within one model and does not constrain the number of routes which may start at any single depot.

1.2. Vehicle capacity constraints

There is a limit to the volume of salt which may be carried on a gritter. Salt is spread onto the road through a spinner which can be adjusted so that both sides of a carriageway are treated on a single pass. The length of road which may be treated in a single route before the gritter must obtain more salt depends on the rate at which it is spread on the road. Precautionary gritting may be carried out at a lower rate of spread than other occasions, for example when ice has already formed. The rate of spread may also depend on how the salt is stored. If the salt is kept in the open, it may be more "lumpy" and so require spreading at a higher rate to avoid jamming the spreading mechanism. Salt which is stored in a barn tends to be of a more consistent quality and can be spread more finely. The gritters themselves could be of different capacities.

In this formulation, the capacities of the gritters to be used and the rates of spread were translated into a constraint on the maximum road distance which could be treated in one route. The maximum distance depends on the depot where the route starts to allow for the fact that not all depots are necessarily equipped with barns to store the salt.

1.3. Road treatment time constraints

Winter gritting is normally done during the early morning when the formation of frost, ice or snow makes the road surface dangerous. If gritting is done too early, wind may blow away the salt which has been spread; if gritting is left too late, roads may still be untreated when the amount of traffic begins to increase. There is therefore a time constraint attached to the gritting operation. In this local authority area, roads are given different categories according to the time within which they must be gritted. Category 1 roads must be treated within two hours of call out and Category 2 roads within four hours of call out. There were two further categories of road where the time for treatment was longer, but these would only need treatment if the bad weather conditions persisted into later in the day and were not normally included in the main gritting routes.

When only the first category of road is being considered, this constraint can be modelled as a constraint on the time allowed for any route (not counting the time to return to the depot after finishing gritting). The existence of different road priorities however raises the question of whether a Category 2 road should also be treated within the first two hours. This may have the advantage of reducing the total distance which needs to be travelled, but could imply that more vehicles will be required, so increasing costs. This question will be considered later in Section 3.

1.4. The objective

In this application, the objective is taken to be the minimisation of the total expected annual costs of the operation, subject to the constraints described. For a given set of depots and vehicle capacities, the routes which are designed can affect total costs in terms of the number of gritters required and the total distance travelled. Thus the routes should be designed to minimise

$$c_G G + c_D D N,$$

where

- $c_G =$ expected annual fixed costs of a gritter,
- $G =$ number of gritters,
- $c_D =$ distance related cost per kilometre,
- $D =$ total distance travelled in kilometres,
- $N =$ expected number of days in the year when gritting is required.

In the case studied, the first term relating to the number of gritters used was much higher than the second which depended on distance travelled.

Another feature of the actual case being considered was that for most scenarios, the routes were more often constrained by the time constraint than the gritting distance constraint and even when the gritting distance constraint was operative the time spent on the route was close to the time allowed. This meant that when gritting a single category of road, each route from a depot had to be done by a different gritter. So minimising the number of routes was equivalent to minimising the number of gritters required. The solution method described below is designed for this problem. If gritters cover more than one route within the time allowed, then minimising the number of routes may not necessarily minimise the number of gritters required. In this case routes must also be assigned to gritters in the order in which they will be carried out. An alternative solution method which deals with this situation is described in Xin and Eglese [12].

2. Solution procedure

2.1. Initial partitioning

A computer database was constructed containing information on the road network which was being considered. This was done by making the use of a digitiser to facilitate recording grid references for road junctions and bends. For each road, its gritting category was recorded. Roads which were part of the network but which did not require gritting (because, for example they were the responsibility of another authority) were included as a special category if they might be used by a gritter to travel from one part of the network to another.

The local authority was split into three administrative areas for gritting purposes. Each of the three areas was considered separately, though if there had been time, some investigation would also have been carried out to see whether altering the area boundaries would be beneficial.

The three areas contained 111, 203 and 380 roads respectively.

2.2. Creation of the cyclenode network

The first part of the solution procedure is based on a method described by Male and Liebman [11] for routeing waste collection vehicles.

The first stage in creating a set of routes to cover an area is to solve the unconstrained CPP for the roads which require treatment. Initially we shall just consider the case where routes for Category 1 roads are required. This is solved optimally by making use of the matching algorithm outlined in Edmonds and Johnson [5]. Arcs are added to the original network which represent the roads over which a gritter must travel but not treat. This is referred to as "deadheading". If any of the added arcs correspond to roads of lower category, i.e., ones which require treatment but not necessarily within two hours, it was assumed that the gritter would treat these as well as the Category 1 roads. This appeared to correspond to current practice, but it would be possible to examine the effect of leaving the treatment of all non-Category 1 roads

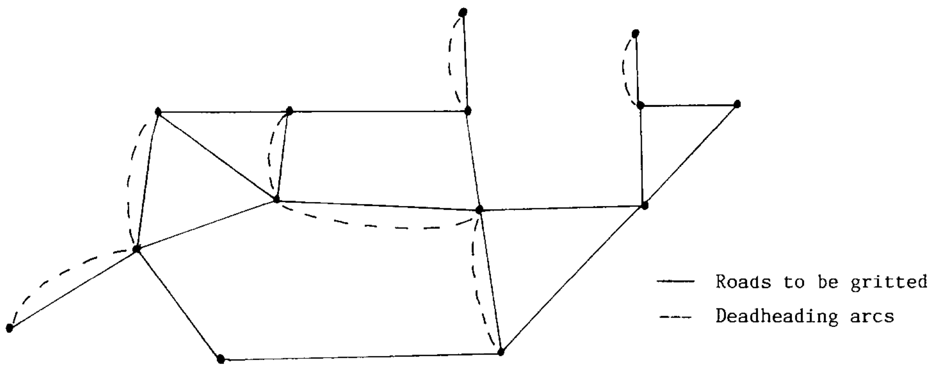

Fig. 1. Original network.

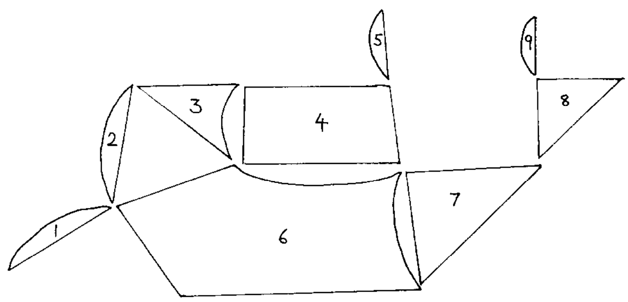

Fig. 2. Cycles formed from solution of CPP.

until later. The average speed of gritters is higher when deadheading than when spreading salt.

Figure 1 shows a small sample network (solid lines) with the deadheading roads (dashed lines), which form the tour of minimum total distance including each Category 1 road. The total arc length represented by the augmented network provides a lower bound for the total distance travelled for any set of feasible routes.

A set of cycles is created from the augmented network so that each arc is in one cycle. There are many such sets, but the set used is constructed according to the "checkerboard pattern" as shown in Fig. 2. An algorithm for producing cycles in this way is given in the Appendix. This helps to preserve the flexibility of the algorithm in later stages by constructing a large number of small cycles. Grouping arcs into fixed cycles which will later be combined to form routes restricts the set of possible solutions. This restriction may lead to a suboptimal solution being produced. However, working with cycles rather than individual arcs has the advantage of reducing the size of the problem (as the number of cycles is less than the number of arcs) and retaining some of the structure of the network in a helpful way. Other methods which do not rely on routes being made up of such cycles are currently under investigation.

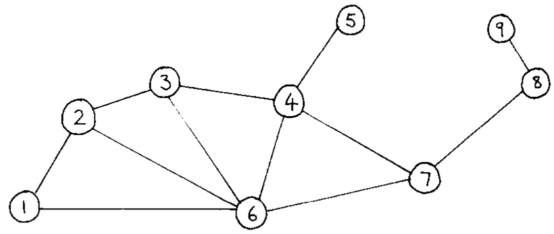

Fig. 3. Cyclenode network.

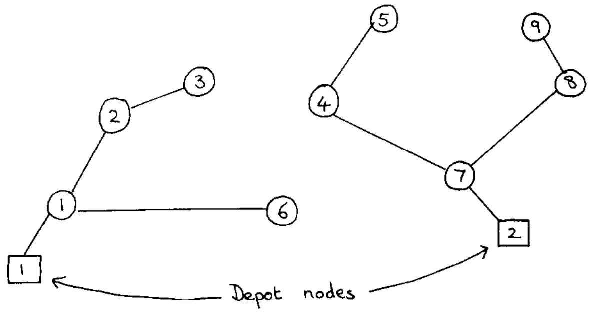

Fig. 4. Two route trees.

For each cycle, the gritting distance, the deadheading distance, and hence the time required for the gritter to cover the cycle are recorded.

A cyclenode network can now be created. In this new network each node represents a cycle in the original network. An edge is included between two cyclenodes where the two cycles share a common node in the original network. Each edge has distance zero, since the distance between adjacent cycles in the original network is zero. The cyclenode network corresponding to the cycles in Fig. 2 is shown in Fig. 3.

This stage only needs to be run once before examining many different scenarios in subsequent stages.

2.3. *Creation of routes*

At this stage the set of depot locations is specified. A depot may be situated at any node of the original network. The cyclenode network is now augmented by adding one node corresponding to each depot and then joining each cyclenode to the depot which is closest to any node on the cycle. These edges have a length corresponding to the shortest distance found. A version of Dijkstra's algorithm was written to find the shortest distances.

Possible routes for gritters are those which may be represented by a tree of cyclenodes which is connected to a depot in this network. Figure 4 shows two trees

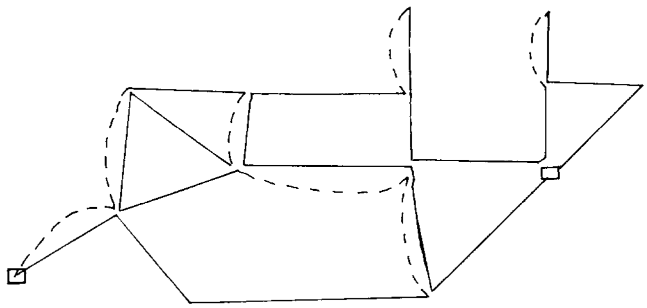

Fig. 5. Two routes on original network.

representing two routes for gritters; one from Depot 1 and one from Depot 2. These trees may simply be decoded into routes on the original network as shown in Fig. 5. This representation means that not all possible routes are considered, since the cyclenode network only includes edges where two cycles have a node in common. However, this representation implies that routes tend to cover roads in a connected district and so the best routes are likely to be included.

2.3.1. Initial solution

A feasible set of routes is first constructed using a quick "greedy" algorithm using a similar heuristic to the Clarke and Wright Savings heuristic for vehicle routeing problems.

Initially we imagine that each cycle is served on an individual route from its nearest depot, i.e., there are as many routes as cyclenodes. Thus the number of routes emanating from any depot is equal to the degree of that depot node in the route tree for that depot. If the set of routes produced is to be feasible, it is necessary to check that each route satisfies the two constraints, one on the road distance which is salted in a route and the other on the time by which treatment must be completed. In practice it was found that these constraints were not always satisfied at this stage. This could be due to the presence of one very large cycle. The problem was overcome by going back and splitting the area under consideration into two or three smaller zones, solving the unconstrained CPP on each zone individually, constructing the set of cycles implied for each zone and then forming the cyclenode network for the whole area.

The number of routes is reduced by attempting to link adjacent cyclenodes to form trees. Each cyclenode is considered in turn. Instead of servicing the cyclenode on an individual route, a new route is considered which is represented by a tree with the cyclenode connected to an adjacent cyclenode in the cyclenode network. If this new route is feasible, i.e., the additional gritting distance and time required do not break the gritting distance and time constraints, then the new route is accepted and the total number of routes has been reduced by one. The saving in the dead distance travelled is equal to twice the length of the edge joining the cyclenode considered to its nearest depot. The cyclenodes are therefore sorted according to their distance from the nearest depot and are examined in order starting with the one farthest from any depot.

2.3.2. Improving the solution by Simulated Annealing

Although the method described gave results which in some cases matched the number of routes used in practice, it was clear that in most cases some improvement was possible. The next stage of the solution process is then to attempt to change the number of routes by making local changes to the set of routes produced by the first stage. In a Simulated Annealing algorithm [4, 10] local changes or neighborhood moves are generated at random. If the objective is improved the move is accepted. If the objective is not improved then the move is accepted with a certain probability. The usual function to use as the acceptance probability is exp($-$ delta/t) where delta is the change in the objective and t is a control parameter referred to as the temperature. The scheme for controlling the temperature as the algorithm proceeds is called the cooling schedule and is described below.

2.3.2.1. Dealing with the constraints. At this stage, instead of keeping all routes feasible by checking the gritting distance and time constraints, both constraints were incorporated into the objective function by adding a penalty if a constraint was broken.

Since the distance related cost was so much smaller than the cost relating to the number of gritters used, the distance travelled was omitted from the objective and Simulated Annealing was used to attempt to minimise

$$R + p_D D^+ + p_T T^+$$

where

- R = no. routes,
- D^+ = total of the distances by which routes exceed the maximum gritting distance,
- T^+ = total of the times by which routes exceed the time allowed for gritting,
- p_D and p_T are the corresponding penalty coefficients.

This strategy means that any set of routes may be transformed into any other set by a sequence of neighbourhood moves as defined in the next section. Use of the penalty function approach also allows the two constraints to be modelled as soft constraints, i.e., we are prepared to consider routes which only break the time or distance constraints by a small amount. This is desirable as average speeds and rates of spreading salt can only be specified approximately.

2.3.2.2. Neighbourhood structure. To describe the neighbourhood structure, a leaf cyclenode is defined as a cyclenode which is only linked to one other cyclenode or only to a depot in the set of route trees. The following changes were defined to be valid

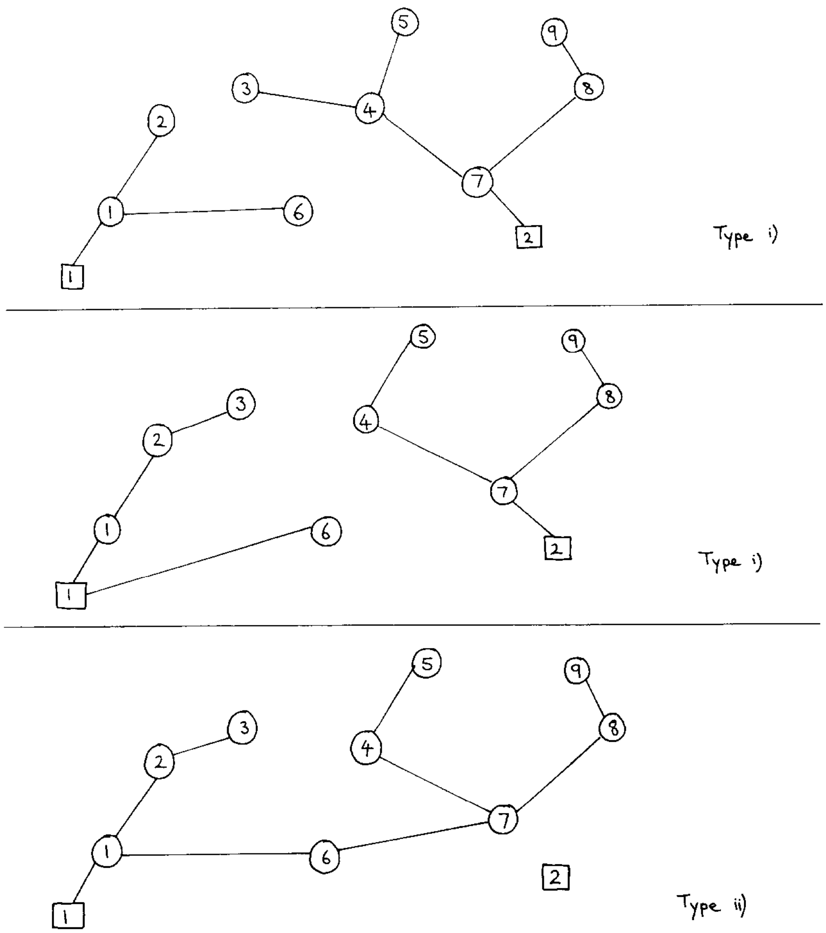

Fig. 6. Examples of neighbourhood moves from the cyclenode network in Fig. 4.

neighbourhood moves:

(i) Any single leaf cyclenode may be removed from its current route tree and linked to any other adjacent cyclenode or linked to a depot to form a new route.

(ii) Any route tree may be combined with another route tree by linking the cyclenode nearest to the depot node to an adjacent cyclenode on another route tree. The longer of the two edges linking a cyclenode to a depot on the two routes is then removed.

Examples of these neighbourhood moves are illustrated in Fig. 6.

Originally only neighbourhood moves of type (i) were allowed. However the results were disappointing. It was recognised that there were significant occasions where two route trees could only be combined into one route tree by several type (i) moves, the

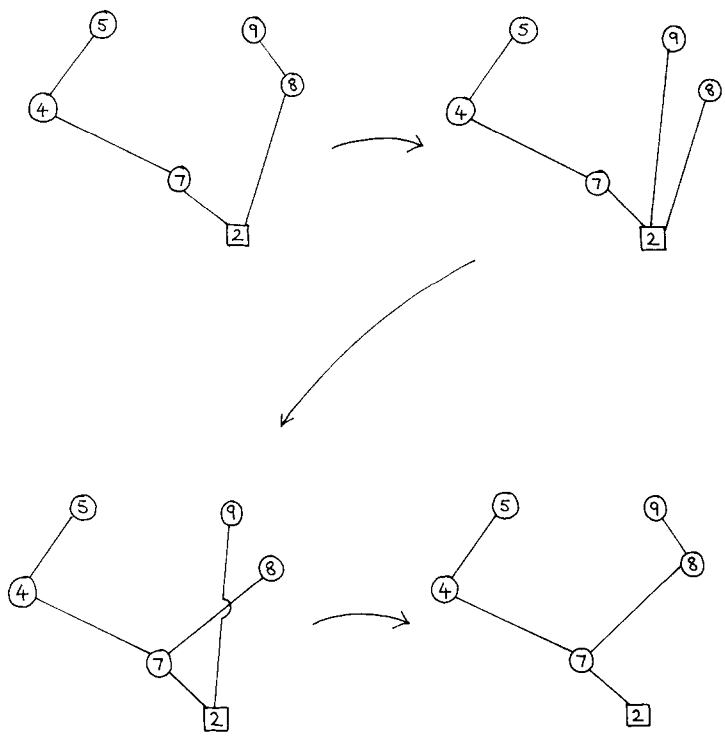

Fig. 7. Transforming two route trees to one route tree through a sequence of type (i) moves.

first of which resulted in three route trees. This situation is illustrated in Fig. 7. Even though the use of Simulated Annealing could potentially allow these transitions to be made, the topology is so "bumpy" that it was difficult to get good results within a reasonable length of computing time. Therefore moves of type (ii) were introduced to produce a "smoother" topology. However there still exist many local optima which means that using a simple local improvement algorithm (equivalent to Simulated Annealing at zero temperature) produces poor results. Other more general neighbourhood moves have been considered, but not tested at this stage.

Instead of generating individual neighbourhood moves at random, a set or block of different moves is generated by considering each cyclenode in turn. For each one, a neighbourhood move involves removing one edge ending at that cyclenode and replacing it with another. Every such exchange is generated for all cyclenodes and then the exchanges are taken in a random order. If the exchange corresponds to a neighbourhood move then the change in the objective is calculated. This means that if only one neighbourhood move will lead to an improvement it will be found within the number of iterations equal to the size of the neighbourhood. It also means that if no improvements have been found after a number of iterations equal to the size of the neighbourhood, then the solution is at least locally optimal.

2.3.2.3. Cooling Schedule. The Cooling Schedule used is similar to the scheme suggested in Johnson et al. [9]. The initial value of the temperature parameter was chosen experimentally so that the algorithm did not become prematurely stuck in a local optimum. At each temperature the number of iterations was a fixed multiple of the size of the neighbourhood. The temperature parameter was then decreased by a constant percentage. The algorithm stopped when the number of routes at a temperature change had remained the same on at least the last five occasions and the number of improvements found at the last temperature was below a set percentage of the number of iterations at that temperature. Suitable parameter values were determined by experimentation.

2.3.2.4. Results. Use of the Simulated Annealing algorithm did lead to improvements compared to the initial solutions which had been found. The number of routes required were equal to or less than the number currently used. However the most significant conclusion from the computer runs was that the number of depots required in the County could be more than halved without increasing the number of gritters.

3. Roads of different priorities

In the case under study, routes were required to cover Category 1 roads within two hours and Category 2 roads within four hours.

The approach described in Section 2 was used first to find routes which covered the Category 1 roads within two hours. The program was then repeated but for the remaining Category 2 roads and routes were constructed which were within two hours. The two sets of routes were then assigned to gritters so that each did a Category 1 route followed by a Category 2 route.

This approach was modified in Area North where there were more Category 2 roads than Category 1 roads. In order to minimise the total number of gritters used, it was necessary to complete all the Category 1 gritting within less than two hours, leaving longer for the Category 2 roads, but which would still all be completed within four hours of the original call out.

An alternative approach was tried where the cycles were formed by solving the CPP for all Category 1 and Category 2 roads together. When routes were formed, attempts were made to form routes containing only Category 2 cycles so that they could be done by a gritter after the Category 1 roads had been completed. However, many cycles contained both Category 1 and Category 2 roads. Constraining all these to be completed within the first two hours forced many Category 2 roads to be gritted during the first two hours and so increased the number of gritters required, even though the total dead distance travelled was less.

When a route is made up of cycles containing Category 1 and Category 2 roads, it may be possible to route a gritter so that the Category 1 roads are completed within two hours, even though the complete route takes longer. But allowing for this means that the time to finish gritting roads of a given category cannot be simply calculated from the sum of the times to cover the cycles and the advantages of working with cyclenodes are lost.

4. Conclusions

The paper has described the formulation of a practical routeing problem which has been implemented as a computer program to run on a microcomputer and used to explore the cost consequences of different scenarios for the gritting operation in a County. The results from this program were used to show how the number of gritters and the number of depots in the County could be reduced while still giving the required level of service.

Current research is now concentrating on refining and improving the models used and comparing alternative approaches.

Appendix: Algorithm for generating a set of cycles in pseudocode

The algorithm assumes that the network is connected and that some roads have a "matching arc" associated with them from the solution of the Chinese Postman Problem and therefore have to be covered twice. Providing a suitable road and node are input at the beginning, the algorithm will create a set of cycles according to a "checkerboard pattern" as illustrated in Fig. 2.

```
{Initialise variables:}
  Input s            {s is the first node for the current cycle}
  Input road         {road is the next arc to be covered for current cycle}
  t := s             {t is node from which current cycle is to be continued}
  kount := 1         {kount records the number of cycles}
  for i := 1 to cymax do    {cymax is the maximum number of cycles}
    with cycleno[i] do      {cycleno[i] refers to the ith cycle}
      cynum := 0            {cynum counts the nodes/arcs in a cycle}
      cypoint := 1          {cypoint indicates the point on a current cycle
                             from which a new one may be started}
  cycleno[1].mark := 0    {for each cycle, mark indicates the root cycle
                           from which this cycle was started}

repeat
  repeat   {this loop builds up the arcs to be covered on this cycle}
    current := kount
```

```
    if (cycleno[current].cynum < > 0) then {i.e. if not first time
                                             through}
      if (lastroad has a matching arc which has only been covered once and
         a new cycle was started from the last node)
      then road := lastroad
      else road := next uncovered road on left from node t
    with cycleno[kount] do
      cynum := cynum + 1
      nodelist[cynum] := t      {for each cycle, nodelist records the nodes
                                 in order}
      arclist[cynum] := road    {for each cycle, arclist records the arcs in
                                 order}
    lastroad := road            {lastroad records the last road covered}
    t := node at other end of road
  until t = s
  repeat  {this loop finds a suitable starting node s for the next cycle}
    foundnode := false  {foundnode indicates whether a suitable node for
                         starting the next cycle has been found}
    if (all arcs meeting at node s have been covered)
    then with cycleno[current] do
      if cypoint < > cynum
      then
        cypoint := cypoint + 1
        s := nodelist[cypoint]
      else
        current := mark
        if current < > 0
        then with cycleno[current] do
          s := nodelist[cypoint]
    else foundnode := true
  until (all arcs have been covered) or (foundnode)
  if (foundnode)
  then  {this section determines the first road for the new cycle}
    t := s
    kount := kount + 1
    cycleno[kount].mark := current
    with cycleno[current] do
      lastroad := arclist[cypoint]
      road := next uncovered road on left at t from lastroad
      unless (lastroad has a matching arc which has only been covered once)
             or (road has a matching arc which has not been covered at all)
      then road := next uncovered road on left at t from road
until (all roads have been covered)
```

References

[1] A.A. Assad, W.L. Pearn and B.L. Golden, The Capacitated Chinese Postman Problem: Lower bounds and solvable cases, Amer. J. Math. Management Sci. 7 (1987) 63–88.
[2] E. Benavent, V. Campos, A. Corberan and E. Mota, The Capacitated Chinese Postman Problem Parts I and II, Presented at CO87, Southampton (1987).
[3] L. Bodin, B.L. Golden, A.A. Assad and M. Ball, Routing and scheduling of vehicles and crews: The state of the art, Comput. Oper. Res. 10 (1983) 63–211.
[4] N.E. Collins, R.W. Eglese and B.L. Golden, Simulated Annealing–An annotated bibliography, Amer. J. Math. Management Sci. 8 (1988) 209–308.
[5] J. Edmonds and E. Johnson, Matching, Euler tours and the Chinese Postman Problem, Math. Programming 4 (1973) 88–124.
[6] B.L. Golden and A.A. Assad, eds., Vehicle Routing: Methods and Studies (North-Holland, Amsterdam, 1988).
[7] B.L. Golden, J.S. DeArmon and E.K. Baker, Computational experiments with algorithms for a class of routing problems, Comput. Oper. Res. 10 (1983) 47–59.
[8] B. Golden and R. Wong, Capacitated arc routing problems, Networks 11 (1981) 305–315.
[9] D.S. Johnson, C.R. Aragon, L.A. McGeoch and C. Schevon, Optimization by Simulated Annealing: An experimental evaluation Part I, Oper. Res. 37 (1989) 865–892.
[10] S. Kirkpatrick, C.D. Gelatt Jr and M.P. Vecchi, Optimatization by simulated annealing, Science 220 (4598) (1983) 671–680.
[11] J.W. Male and J.C. Liebman, Districting and routing for solid waste collection, J. Environmental Engrg. Division ASCE 104 (1978) 1–14.
[12] Xin Zhanhong and R.W. Eglese, The road gritting problem and its heuristic solution, Working Paper (1989).