

Engineering

Industrial & Management Engineering fields

Okayama University

Year 2005

Dynamic salting route optimisation using
evolutionary computation

Hisashi Handa
Okayama University

Lee Chapman
University of Birmingham, UK

Xin Yao
University of Birmingham, UK

This paper is posted at eScholarship@OUDIR : Okayama University Digital Information Repository.

<http://escholarship.lib.okayama-u.ac.jp/industrial-engineering/5>

Dynamic Salting Route Optimisation using Evolutionary Computation

Hisashi Handa
 Graduate School of Natural Science
 and Technology
 Okayama University
 Tsushima-Naka 3-1-1
 Okayama, 700-8530, JAPAN
 handa@sdc.it.okayama-u.ac.jp

Lee Chapman
 School of Geography, Earth, and
 Environmental Science
 The University of Birmingham
 Edgbaston
 Birmingham, B15 2TT, U.K.
 l.chapman@bham.ac.uk

Xin Yao
 CERCIA
 School of Computer Science
 The University of Birmingham
 Edgbaston
 Birmingham, B15 2TT, U.K.
 x.yao@cs.bham.ac.uk

Abstract- On marginal winter nights, highway authorities face a difficult decision as to whether or not to salt the road network. The consequences of making a wrong decision are serious, as an untreated network is a major hazard. However, if salt is spread when it is not actually required, there are unnecessary financial and environmental consequences. In this paper, a new salting route optimisation system is proposed which combines Evolutionary Computation (EC) with the neXt generation Road Weather Information Systems (XRWIS). XRWIS is a new high resolution forecast system which predicts road surface temperature and condition across the road network over a 24 hour period. ECs are used to optimise a series of salting routes for winter gritting by considering XRWIS temperature data along with treatment vehicle and road network constraints. This synergy realises daily dynamic routing and it will yield considerable benefits for areas with a marginal ice problem.

1 Introduction

Local authorities in marginal winter climates are responsible for the precautionary salting of the road network. In the case of UK, there are approximately 3000 salting routes which cover about 120,000km or 30% of the road network. With limited resources and treatment time constraints it is imperative that salting routes are planned in advance for efficient and effective operation. This has traditionally been a manual task and is heavily reliant on local knowledge and experience. Currently, a 'static,' often paper based, approach is used to optimise salting routes within the given constraints to enable effective use of resources (i.e. treatment vehicles, personnel and de-icing chemical material). The aim is to maintain safe road conditions, whilst minimising financial and environmental costs.[Thornes96].

The decision as to whether to salt the road network on a particular night is based on weather forecasts issued via a Road Weather Information System (RWIS). The first generation RWISs were developed in the 1980's and relies on thermal mapping measurements to interpolate site-specific forecasts around the road network. Despite the inherent limitations of thermally projecting forecast data, the methods and tools utilised by highway authorities have changed little over the last 20 years, and it is for this reason that the neXt generation Road Weather Information System (XRWIS) was developed. Instead of simply measuring road sur-

face temperatures across the network, XRWIS models road surface temperatures by considering geographical parameters such as the sky-view factor, altitude and landuse. The forecast is displayed in a GIS environment and disseminated to the highway engineer via the Internet. Although highway authority trials are still ongoing, XRWIS is demonstrating significant benefit and advantages compared to the first generation products still widely used throughout the UK. However, despite the large advance in applying technology provided by XRWIS, for local authorities to achieve 'best value' in winter road maintenance, additional tools are required to make the most effective use of resources.

In this paper, a new salting route optimisation (SRO) system is proposed, which combines XRWIS with Evolutionary Computation (EC). ECs are used to optimise a series of salting routes for winter gritting by considering XRWIS temperature data along with treatment vehicle and road network constraints. This synergy realises daily dynamic routing and it will yield considerable benefits for areas with a marginal ice problem.

2 XRWIS

XRWIS is a new intuitive route based forecast system that provides the highway engineer with all the information required to make improved salting decisions. Instead of modelling road condition at a single site and interpolating temperatures by thermal maps, XRWIS models surface temperature and condition at thousands of sites around the road network. This is achieved by considering the influence of local geography on the climatology of the road [Chapman01a, Chapman01b]. Data is collected along each salting route by conducting a survey of the sky-view factor (a measure of the degree of sky obstruction by buildings and trees) [Chapman02, Chapman04]. This is then combined with other geographical parameters (latitude, longitude, altitude, slope, aspect, road construction, thermal map residual temperature, landuse and traffic volume) to produce a high resolution geographical parameter database.

The geographical data is combined with mesoscale meteorological data in an energy balance model to predict road conditions at typical spatial and temporal resolutions of 20 metres and 20 minutes respectively. The output is displayed as a colour-coded map of road temperature and condition that is disseminated over the Internet to the highway engineer. From this it can provide a suggested action whether or

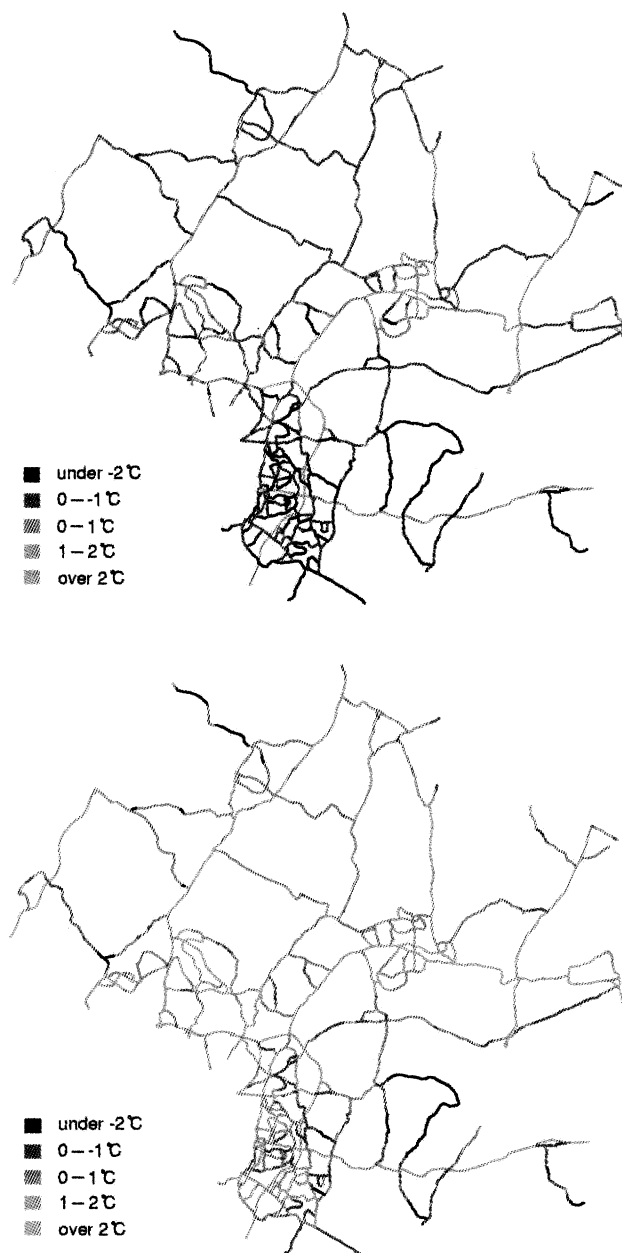


Figure 1: Temperature distributions for two nights in South Gloucestershire: on a cold night (UPPER) and on a marginal night (LOWER)

not an individual salting route needs treating.

Figure 1 shows example temperature forecasts of salting routes in the South Gloucestershire, UK. The colour of each point represents the temperature predicted by XRWIS, i.e., the colour is gradually varied from blue for cold points (dark grey in the case of B.W. prints) to red for warm points (light grey). Figure 2 shows an example of the changes of temperature, predicted by the XRWIS, at a single site (point) in the South Gloucestershire during a day.

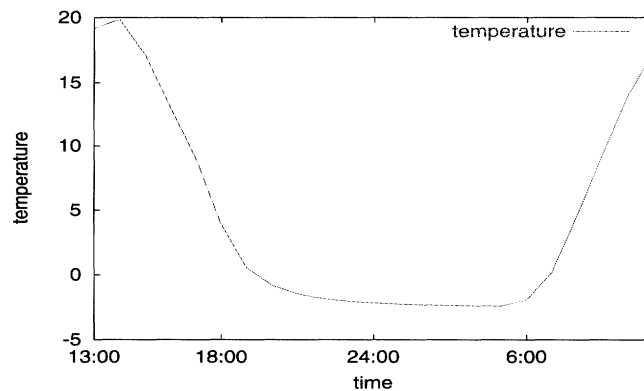


Figure 2: The changes of predicted temperature at a single site over a 24 hour period.

3 Salting Route Optimisation

3.1 Capacitated Arc Routing Problems

SRO can be regarded as an instance of the Capacitated Arc Routing Problem (CARP) [Golden81, Lacomme04]. Suppose that a graph $G = (V, E)$ is given, where V and E are sets of vertices and edges, respectively. Each edge e in E has a cost C_e . Additionally, a set $R \subset E$ of required edges is defined in the CARP. A demand D_e is defined to each edge e in R . There are several vehicles to fill the demands, where each vehicle has the predefined capacity of services for the demands. A depot is defined elsewhere in V . All vehicles must depart from this depot and return there at the end of their service tour. The problem is to find a set of tours which have a minimum total cost for all vehicles, ensuring the demands of all required edges are filled by at least one vehicle, whilst ensuring the total services capabilities of each vehicle are not exceeded.

3.2 Calculation of the Costs of Tours

It is difficult to predefine the deadheading costs between required edges. A tour for a vehicle is defined as a sequence of required edges, whereas the actual path of the tour is defined as a sequence of all edges which the vehicles must traverse. For example, assuming that there are four required edges A, B, C, and D (Figure 3a). The minimum actual paths of tour "BC", "ABC", and "ABCD", which are the minimum length of all possible actual paths, are those as shown in Figure 3 (b), (c), and (d), respectively. Note that deadheading edges between required edges B and C in these figure are depicted in dashed lines. These dashed lines are different from each other. That is, deadheading costs between adjacent required edges varies in accordance with a sequence of required edges in a tour. Therefore, all the required edges in a tour need to be taken into account in order to calculate the total cost.

A distance matrix is employed between vertices which is unchanged during evolutionary search. The distance matrix is calculated by using Dijkstra's Algorithm before evo-

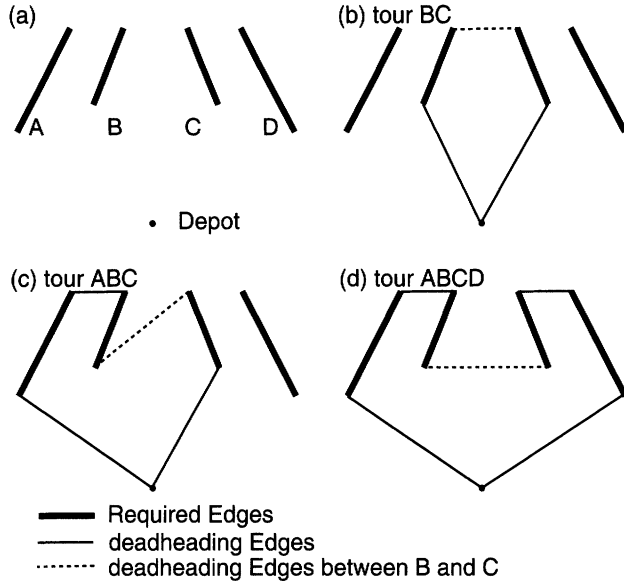


Figure 3: (a) An alignment of required edges A, B, C, D, actual paths of tour (b) “BC”, (c) “ABC”, and (d) “ABCD”

lutionary computation is run. The cost of tours is calculated as follows: Firstly, consider the graph in Figure 4, which consists of a depot node and a sequence of required edges whose order is the same as a sequence in tours (Note that although there are two depot nodes in this figure, these nodes are the same). The minimum cost $d_{min}(t_{e_1}^0)$, $d_{min}(t_{e_1}^1)$ to arriving at the terminal nodes $t_{e_1}^0$, $t_{e_1}^1$ of the first required edge e_1 in the tour is set to $D(depot, t_{e_1}^0)$ and $D(depot, t_{e_1}^1)$, respectively, where $D(\cdot, \cdot)$ denotes the element of the distance matrix, and $depot$ stands for the depot node. The minimum cost $d_{min}(t_{e_i}^0)$, $d_{min}(t_{e_i}^1)$ to arriving at the terminal nodes $t_{e_i}^0$, $t_{e_i}^1$ of required edge e_i is recursively calculated by using the minimum cost $d_{min}(t_{e_{i-1}}^0)$, $d_{min}(t_{e_{i-1}}^1)$ of previous required edge e_{i-1} . That is,

$$d_{min}(t_{e_i}^0) = \min(d_{min}(t_{e_{i-1}}^0) + C_{e_{i-1}} + D(t_{e_{i-1}}^1, t_{e_i}^0), \\ d_{min}(t_{e_{i-1}}^1) + C_{e_{i-1}} + D(t_{e_{i-1}}^0, t_{e_i}^0))$$

$$d_{min}(t_{e_i}^1) = \min(d_{min}(t_{e_{i-1}}^0) + C_{e_{i-1}} + D(t_{e_{i-1}}^1, t_{e_i}^1), \\ d_{min}(t_{e_{i-1}}^1) + C_{e_{i-1}} + D(t_{e_{i-1}}^0, t_{e_i}^1)),$$

where C_{e_i} is the cost of i^{th} required edge. Finally, the cost C_{T_i} of tours T_i is defined as

$$C_{T_i} = \min(d_{min}(t_{e_{last}}^0) + C_{e_{i-1}} + D(t_{e_{last}}^1, depot), \\ d_{min}(t_{e_{last}}^1) + C_{e_{i-1}} + D(t_{e_{last}}^0, depot)),$$

where e_{last} indicates the last required edge in the tour T_i .

3.3 Mapping from SRO to CARP

In the case of SRO, vertices are set on intersections or branch points of roads, whereas edges are defined as roads between vertices. In accordance with this definition, several vertices and short edges are generated at some of the more complicated features of the network, for example, round-

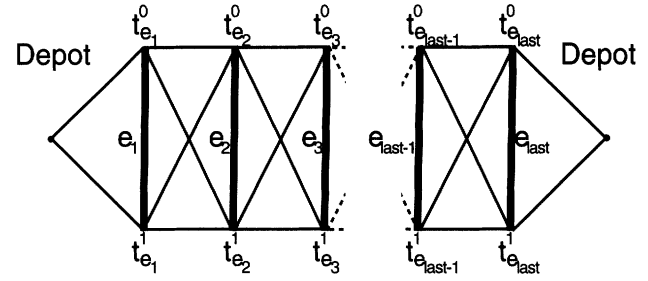


Figure 4: A graph for calculating the cost of tours. Note the identical depot nodes.

abouts. In order to simplify problems without loss of generality, roundabouts are regarded as intersections. Using Figure 1 as an example, there are 419 vertices and 597 edges. The costs on edges are defined as the distance of these features. The set of required edges and their demands, i.e. the amount of salt, are defined by referring to the predicted temperature provided by XRWIS. As described in section 2, road surface temperature is predicted at 20m intervals along the route. If a road section is predicted to go below freezing, then 10g/m^2 salt is required to be spread on the section before ice forms. Moreover, the actual amount of salt required will vary with road width (type), e.g. Motorway, A-Road, B-Road etc. Thus, the amount of salt $S(e)$ on an edge e is defined as follows:

$$S(e) = \sum_{o \in e} d(o, \text{succ}(o, e)) \times w(e) \times f(t(o) - \theta),$$

where $\text{succ}(o, e)$ and $w(e)$ denote the succeeding prediction point of the prediction point o on the edge e , and width of the edge e , respectively. $f(x)$ is the threshold function such that $f(x)$ returns 1 if $x < 0$, otherwise 0. $t(o)$ and θ are the predicted temperature at o and threshold value fixed in advance. If $S(e)$ is greater than 0, the edge e is regarded as a member of the set of required edges.

4 Evolutionary Computation for SRO

A new Memetic Algorithm for solving large-scale SRO problems is designed in this paper. In order to cope with large scale problems, the edge assembly crossover (EAX) operator proposed by Nagata *et al.* is used due to its search ability [Nagata97, Nagata04]. The EAX operator can solve for Traveling Salesman Problems with 2393 cities with probabilities 90 % over. However, since this operator is designed for solving Travelling Salesman Problems, it can often yield an infeasible solution. Hence, a repair operator for offspring individuals is incorporated in the Memetic Algorithm. The EAX operator and the repair operator are explained in subsection 4.2.

As with Lacomme’s Memetic Algorithms for CARP, some initial individuals are generated by path scanning [Golden83, Lacomme04]. Because the EAX operator has similar characteristics to the k-opt operator, three naive local search methods are used in the Memetic Algorithms: Move 1-edge, Move 2-edges, and Swap 2-edges.

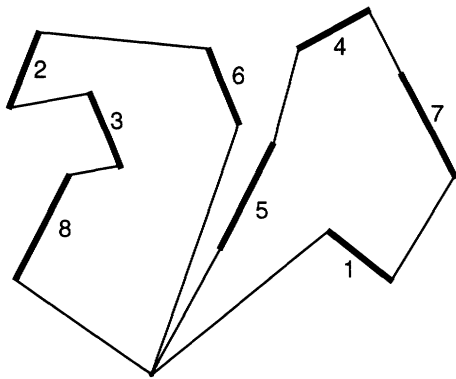


Figure 5: A tour example for a chromosome 2 6 s_1 5 4 7 1 s_2 8 3

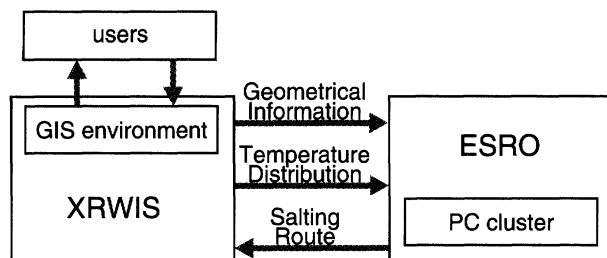


Figure 6: Diagram of the prototype system

4.1 Coding Method and Fitness Evaluation

A naive permutation encoding method for solving SRO is employed. In the case of Travelling Salesman Problems, the permutation representation of a chromosome detailing the order of cities in which a salesman will visit is often used. However, for the SRO, the IDs of required edges are assigned in each loci instead of cities. Special symbols are used indicating the beginning of tours for each truck into the chromosome. Suppose that the special symbol for truck i is denoted by s_i , and the ID for each required edge is uniquely assigned as one of the natural numbers. Then, the following chromosome stands for tours in Figure 5.

2 6 s_1 5 4 7 1 s_2 8 3

In order to cope with constraints with respect to the capacity of the services for the demands, the following fitness function is used [Goldberg87]:

$$F = \sum_{i=0}^m (C_{T_i} + C_p \times E_{T_i})$$

where C_{T_i} denotes the cost of the i^{th} truck's trip T_i as described in equation (1), and C_p is a predefined coefficient for the penalty term. E_{T_i} indicates the quantity of constraint violation in each truck. That is, E_{T_i} is defined as follows:

$$E_{T_i} = \begin{cases} D_{T_i} - L_i & \text{if } D_{T_i} - L_i > 0 \\ 0 & \text{Otherwise,} \end{cases}$$

where D_{T_i} and L_i denote the total services for the demands by truck i and the capacity of the services for truck i , respectively.

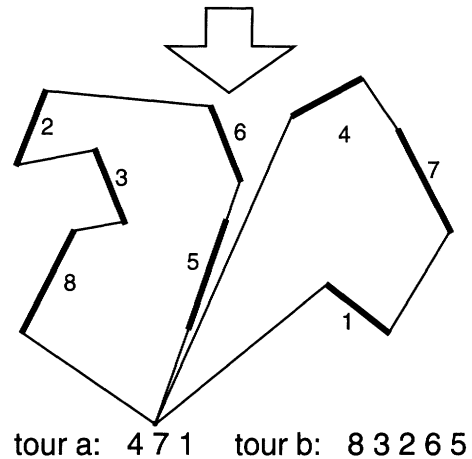
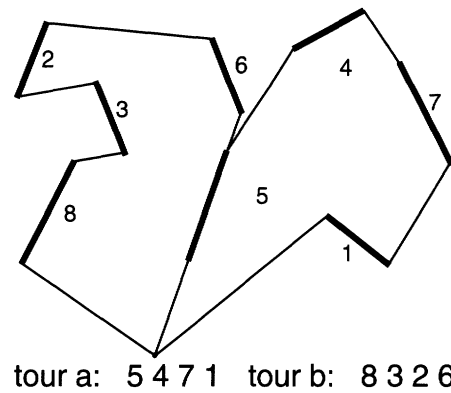


Figure 7: A depiction of the repair procedure

4.2 Repair Operation

Since the EAX operator is designed for Travelling Salesman Problems, it has great capability to identify the shortest tours. However, it often yields unfeasible solutions where the capacity of the services is exceeded. These unfeasible solutions are fixed by using the following repair operation:

1. A counter variable *count* is set to 0.
2. Find a tour a which has maximum violation with respect to the constraint of the service capacity.
3. Randomly choose a required edge r in the tour a .
4. Find a tour b , which has an opening for the required edge t , such that the required edge must be traversed as a deadheading path (Figure 7). If no tour is found, increment *count* and go to 7. Otherwise go to the next step.
5. Move the required edge r from the tour a to the tour b .
6. Increment *count* and recalculate the total amount of services for the tours a and b .
7. Loop back to 2. until there is no violation in all tours or *count* exceeds 30.



Figure 8: Required edges for Figure 1: on a cold night (UPPER) and on a marginal night (LOWER)

5 Prototype Systems

5.1 Overview

The prototype system proposed in this paper is shown in Figure 6 and consists of the XRWIS and Evolutionary Salting Route Optimisation (ESRO) module intercommunicating via a Socket. A vector representation of the road network is initially stored in the ESRO module where it is combined with the daily temperature distribution predicted by the XRWIS. The ESRO module then transfers the temperature distribution into a CARP instance as described in section 3 which is then translated to a PC cluster. The PC

nodes then try the same CARP instance with different random seeds. By requesting a response from the XRWIS, the current best solution from all solutions found by all PC nodes is presented from the ESRO to the XRWIS. The XRWIS then displays the resultant salting routes to users by using its visualisation environment.

5.2 Experiment Settings

In this paper, results are shown for two nights in the South Gloucestershire (Figure 1). The temperature distributions are transformed into CARP instances as mentioned in section 3. The required edges in generated CARP instances are shown in Figure 8. There are 385 and 97 required edges in the CARP instances for the two nights respectively. As described in section 3, an edge is regarded as a required edge if there is at least one XRWIS point with a predicted temperature less than the predefined threshold θ . Thus, although the required edges are depicted in red lines, the amount of required salt on the same required edge for two days will be different. The number of trucks required is derived from the total amount of required salt; 11 trucks are needed on the first night and just 3 on the second night (The capacity of all trucks is assumed to be the same and the threshold θ is set to 0).

Parameters of the ESRO module are described as follows: population size is set to 300, the probability of carrying out local search is set to 0.1, finally, no mutation operator is used. In this paper, a PC cluster which consists of 22 Itanium 2 processors is used. Hence, for a CARP instance, 22 different random seeds are examined. The number of generations for each run is set to 60,000. The computation times involved are 6 hours for the first day and 50 minutes for the second example.

5.3 Experimental Results

Figures 9 (a) and 10 show the acquired solutions for CARP solutions on two nights. Thick lines denote that truck should salt on the line whereas thin lines denotes the truck passes through without treatment. For clarity, 11 tours are separately depicted into 3 diagrams Figure 9 (b) – (d). Note that Figure 9 (c) is a zoomed-in area of a dense section of the road network. In Figure 9 (b) – (d), grey lines indicate required edges treated in other pictures. The lower bounds of these CARP instances has not yet been investigated. [Belenguer03]. However, Figure 9 and 10 elucidate that the prototype systems can generate reasonable salting routes.

The change of best fitness for the CARP instance on the cold day is depicted in Figure 11. The lower horizontal line, the middle line, and the upper line denote the best value, averaged value, and the worst value of best fitness in each run, respectively. As mentioned in the previous subsection, it takes approximately 6 hours to run 60,000 generations, so this prototype system could provide better solutions within 3 hours.

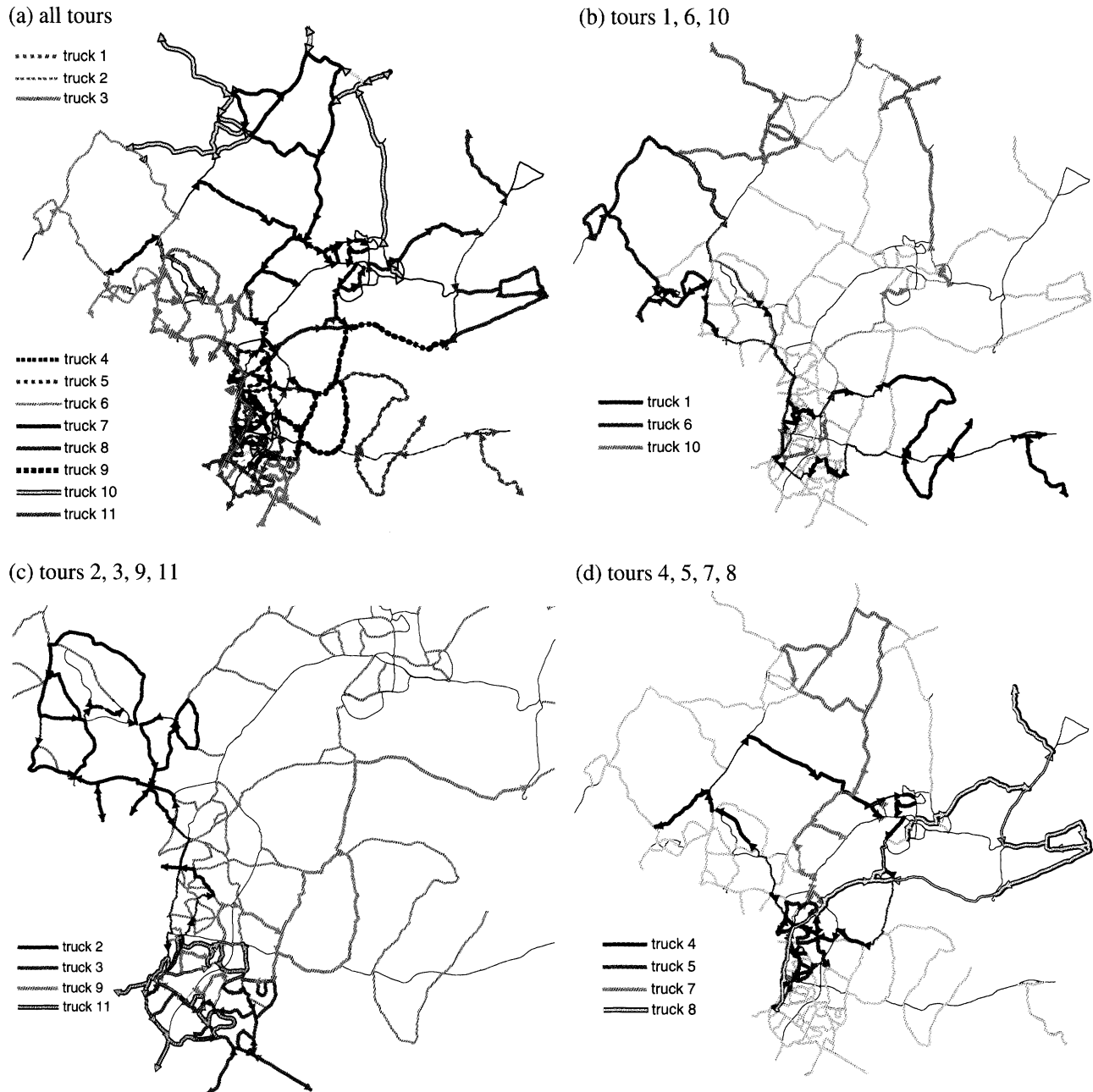


Figure 9: Acquired solution for a CARP instance on the cold day: (a) all tours, (b) tours 1, 6, 10, (c) tours 2, 3, 9, 11 and (d) tours 4, 5, 7, 8; The thick lines denote that truck should salt on the line. The thin lines denotes the truck traverses without service on the line.

6 Conclusions

In this paper, a new salting route optimisation system was proposed which combines Evolutionary Computation with XRWIS. By using temperature data predicted by the XRWIS and by taking into account constraints regarding treatment vehicles and the road network, it has been shown that ECs have the ability to optimise a series of salting routes for winter gritting. The prototype system was constructed as described in section 5, and was examined on two typical marginal nights. The resultant salting routes elucidated the effectiveness of the proposed method.

Future work will include: Firstly, the quality of acquired solutions needs to be evaluated by using lower bounds of CARP instances. It might also be useful for deciding terminal criteria of evolutionary search if the calculation time of the lower bounds is small. Secondly, the re-use of acquired previous solutions may be effective for daily salting route optimisation in the same area. The difference between temperature distributions must be one of the key indexes for re-use since close temperature distributions yield similar CARP instances. Finally, the prototype system will be extended to treat larger areas than that of South Gloucester-

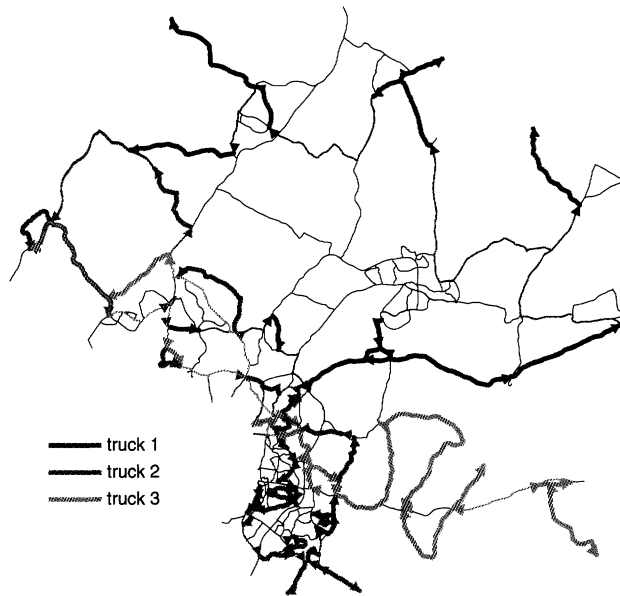


Figure 10: Acquired solution for a CARP instance on the second night. Key to each line is the same as Figure 9

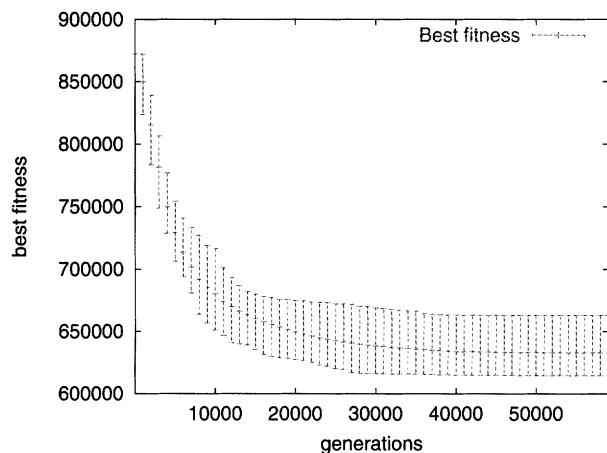


Figure 11: The changes of best fitness: the lower horizontal line, the middle line, and the upper line denote the best value, averaged value, and the worst value of best fitness in each run, respectively.

shire.

Acknowledgements

This work was carried out while the first author was a visiting research fellow at CERCIA. This visit was supported by the Ministry of Education, Science, Sports and Culture, Japan through Overseas Advanced Educational Research Practice Support Program.

Bibliography

- [Belenguer03] Belenguer, J.M., Benavent, E. (2003) "A Cutting Plane Algorithm for the Capacitated Arc Routing Problem" *Computers and Operations Research*, Vol. 30, No. 5, pp. 705–728
- [Chapman01a] Chapman, L., Thornes, J.E., and Bradley, A.V. (2001) "Modelling of road surface temperature from a geographical parameter database. Part 1: Statistical," *Meteorological Applications*, Vol. 8, pp. 409–419
- [Chapman01b] Chapman, L., Thornes, J.E., and Bradley, A.V. (2001) "Modelling of road surface temperature from a geographical parameter database. Part 2: Numerical," *Meteorological Applications*, Vol. 8, pp. 421–436
- [Chapman02] Chapman, L., Thornes, J.E., Bradley, A.V. (2002) "Sky-view factor approximation using GPS receivers," *International Journal Climatology*, Vol. 22, No. 5, pp. 615–621
- [Chapman04] Chapman, L., Thornes, J.E. (2004) "Real-Time Sky-View Factor Calculation and Approximation" *Journal of Atmospheric and Oceanic Technology*, Vol. 21, pp. 730–741
- [Goldberg87] Goldberg, D.E., and Smith, R.E. (1987) "Nonstationary Function Optimization using Genetic Algorithms with Dominance and Diploidy," *Proceedings of Second International Conference on Genetic Algorithms and their Applications*, pp. 59–68
- [Golden81] Golden, B., and Wong, R.T. (1981) "Capacitated arc routing problem," *Networks*, Vol. 11, pp. 305–315
- [Golden83] Golden, B., and DeArmon, J.S., and Baker, E.K. (1983) "Computational Experiments with Algorithms for a Class of Routing Problems," *Computers and Operational Research*, Vol. 10, No. 1, pp. 47–59
- [Lacomme04] Lacomme, P., Prins, C., and Ramdanecherif, W. (2004) "Competitive Memetic Algorithms for Arc Routing Problems," *Annals of Operations Research*, Vol. 131, pp. 159–185
- [Maniezzo04] Maniezzo, V. (2004) "Algorithms on large directed CARP instances: urban solid waste collection operational support," Technical Report UBLCS-2004-16
- [Moscat089] Moscato, P. (1989) "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," *Caltech Concurrent Computation Program*, C3P Report 826

- [Nagata97] Nagata, Y., and Kobayashi, S. (1997) "Edge Assembly Crossover: A High-power Genetic Algorithm for the Traveling Salesman Problem," Proceedings of the 7th International Conference on Genetic Algorithms, pp. 450–457
- [Nagata04] Nagata, Y. (2004) "The EAX Algorithms Considering Diversity Loss," Proceedings of the 8th International Conference on Parallel Problem Solving from Nature – PPSN VII, pp. 332–341
- [Thornes96] Thornes, J.E. (1996) "The Cost-Benefit of Winter Road Maintenance in the UK," Proceedings of the 8th International Road Weather Conference, pp. 1–10