Robust Solution of Salting Route Optimisation Using Evolutionary Algorithms

Hisashi Handa, Dan Lin, Lee Chapman, Xin Yao

Institutions: Okayama University, University of Birmingham

Published on: 11 Sep 2006 - IEEE International Conference on Evolutionary Computation

Robust Solution of Salting Route Optimisation Using Evolutionary Algorithms

Hisashi Handa
Okayama University

Dan Lin
Tianjin University

Lee Chapman
University of Birmingham, UK

Xin Yao
University of Birmingham, UK

2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Robust Solution of Salting Route Optimisation Using Evolutionary Algorithms

Hisashi Handa, *Member, IEEE*, Dan Lin, Lee Chapman and Xin Yao, *Fellow, IEEE*

Abstract— **The precautionary salting of the road network is an important maintenance issue for countries with a marginal winter climate. On many nights, not all the road network will require treatment as the local geography will mean some road sections are warmer than others. Hence, there is a logic to optimising salting routes based on known road surface temperature distributions. In this paper, a robust solution of Salting Route Optimisation using a training dataset of daily predicted temperature distributions is proposed. Evolutionary Algorithms are used to produce salting routes which group together the colder sections of the road network. Financial savings can then be made by not treating the warmer routes on the more marginal of nights. Experimental results on real data also reveal that the proposed methodology reduced total distance traveled on the new routes by around 10conventional salting routes.**

I. Introduction

In countries with a marginal winter climate, highway authorities are responsible for the precautionary salting of the road network. In the case of UK, there are approximately 3000 salting routes covering 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. To aid this process, a Salting Route Optimisation system (SRO) which combines evolutionary algorithms with the neXt generation Road Weather Information System (XRWIS) has been developed [1]. The SRO system can cope with large-scale instances in the real world within reasonable computation times, to the extent that a daily dynamic salting route optimisation can be realised. However, such a dynamic system which responds daily to changes in temperature distributions, will result in salting routes that will also change on a daily basis. Such a complicated system may confuse maintenance engineers, to the extent that errors will occur in the treatment regime. Therefore, a robust solution of salting route optimisation is also required. Here, emphasis is placed on 'thermally ranking' optimised routes so that the 'warmer' routes could be left untreated on marginal nights. Although financial savings using a robust solution are potentially less than a dynamic solution, this is still a considerable advance on existing techniques.

Hisashi Handa is with the Graduate School of Natural Science and Technology, Okayama University, Okayama, 700-8530, Japan (phone: +81-86-251-8250; fax: +81-86-251-8256; email: handa@sdc.it.okayama-u.ac.jp).
Dan Lin is with the School of Sciences, Tianjin University, Tianjin 300072, P. R. China (email: dlin@tju.edu.cn).
Lee Chapman is with the School of Geography, Earth, and Environmental Science, The University of Birmingham, Edgbaston, Birmingham, B15 2TT, U.K. (email: l.chapman@bham.ac.uk).
Xin Yao is with CERCIA, School of Computer Science, The University of Birmingham, Birmingham, B15 2TT, U.K. (email: x.yao@cs.bham.ac.uk).

In this paper, a new evolutionary SRO which generates robust solutions is presented. Salting routes denote a set of tours for a variable number of trucks which cover the whole area. Each individual, which represents a salting route, is evaluated on a set of typical temperature distributions. For each temperature distribution, a sequence of required edges is extracted from individuals before the distance of the edges is calculated. A fitness function is defined by accumulating the distance for all typical temperature distributions and a weighted sum of the distance for this calculation is also examined.

II. Salting Route Optimisation with XRWIS

The salting route optimisation system represents a synergy of Evolutionary Algorithms with XRWIS. XRWIS is explained in more detail in the next section, but essentially provides past and predicted temperature distributions to the Evolutionary Algorithms module for evolution and operation. The system is fronted by an intuitive Graphical User Interface to display the acquired robust routes generated by Evolutionary Algorithms. Although the road network is inputted into the SRO as vector routing data, additional 'local' information can also be entered using the GUI, such as no right turns, one way streets, new roads and driver preferences. This section explains how XRWIS data is combined with the vector routing to translate the salting route optimisation problem into a Capacitated Arc Routing Problems (CARP).

A. XRWIS

The decision of whether to salt is taken by consulting a Road Weather Information System (RWIS) which combines weather forecast data with road temperature and condition data. The first generation of RWIS currently in use relies on methods and tools developed in the 1980s, but as technology has moved on, it is now being superseded by the neXt generation RWIS (XRWIS) (Thornes et al, 2005). XRWIS is a new route-based forecast system which accurately predicts road temperatures to a high spatial and temporal resolution. Instead of modelling road condition at a single site and interpolating temperatures by thermal maps, XRWIS models road surface temperatures at thousands of sites around the road network by considering the influence of the local geography on the road surface [2], [3]. 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) [4], [5]. 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 be suggested as to whether or not an individual salting route needs treating.

Fig. 1 shows example temperature forecasts of salting routes in the South Gloucestershire, UK. The shaded colour of each point represents the temperature predicted by XRWIS, i.e., the colour is gradually varied from dark grey for cold points to light grey for warm points.

B. Capacitated Arc Routing Problems

SRO can be regarded as an instance of the Capacitated Arc Routing Problem (CARP) [6], [8]. 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.

In the case of SRO, the road network is divided into vertices and edges. 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, at roundabouts (in order to simplify problems without loss of generality, roundabouts are regarded as intersections). For example, using Fig. 1 as an example, there are 419 vertices and 597 edges.

The costs of an edge is defined as the length of these feature. The set of required edges and their demands, i.e. the amount of salt, are then defined by referring to the predicted temperature provided by XRWIS. As described in section II-A, XRWIS predicts road surface temperature typically 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 that section before ice forms. However, 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:

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

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

where $\mathrm{succ}(o, e)$ 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.

C. Representation and Evaluation of Tours

Each salting operation will see several trucks in operation. Suppose that each edge is assigned a unique ID and the number of available trucks, N, is predefined. Tours T_i for trucks i ($i = 1, \ldots N$) can be denoted as a sequence of the edge IDs. For instance, let $N = 3$, a set of all edge IDs is

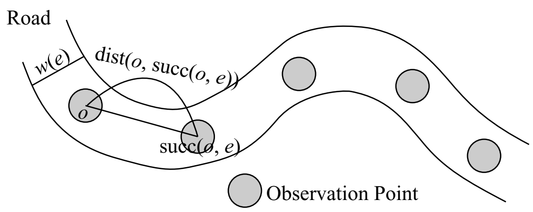

Fig. 2. Translation of SRO to CARP instance

$\{1, 2, 3, 4, \ldots, 10\}$. A set of tours is represented as follows:

T_1	2 3 5 6
T_2	1 8 4
T_3	10 9 7

For a temperature distribution a, a set of edges requireing treatment $R(a)$ are defined. The set of tours is then rewritten by neglecting nonmembers of the set of required edges $R(a)$. Let $R(A) = \{2, 3, 4, 6, 8\}$. The set of tours is rewritten for the temperature distribution a as follows:

$T_1(a)$	2 3 6
$T_2(a)$	8 4
$T_3(a)$	

In this example, $T_3(a)$ is not used for service. Therefore, the number of trucks required is 2.

The evaluation of a set of tours X is represented as follows:

$$E(X, a) = \sum_{T_i(a) \in X} C(T_i(a)) + C_p \times P(T_i(a), a), \quad (1)$$

where $C(\dot{})$ denotes the cost function for a tour, which calculates the distance of the tour, and C_p is a predefined coefficient for the penalty term. $P(\dot{})$ indicates the quantity of constraint violation in each truck. That is, $P(\dot{})$ is defined as follows:

$$P(T, a) = \begin{cases} D(T, a) - L(T) & \text{if } D(T, a) - L(T) > 0 \\ 0 & \text{Otherwise,} \end{cases}$$

where $D(T, a)$ denotes the total services in tour T at a temperature distribution a, and $L(T)$ is a limitation subject to a truck for tour T.

III. Robust Solution of Salting Route Optimisation

A. Robust Solution

Searching for robust solutions is one of the most significant topics of evolutionary optimisation in uncertain environments [11]. Robust solutions are required for problems where the decision variables or environmental parameters[1] are subject to perturbation. The notion of effective fitness function is often used in this research area [9], [10].

$$F(X) = \int_{-\infty}^{\infty} f(X + \delta)p(\delta)d\delta, \quad (2)$$

where $p(\delta)$ indicates the probability distribution of perturbation δ. In the case of where a perturbation is added to the environmental parameters a,

$$F(X) = \int_{-\infty}^{\infty} f(X, a + \delta)p(\delta)d\delta. \quad (3)$$

In practise, an approximation of the effective fitness function is used. The approximation of the effective fitness function in the former is written by

$$\hat{F}(X) = \sum_{i=0}^{N} \frac{1}{N} f(X + \delta_i), \quad (4)$$

where N denotes the number of samplings to estimate $f(X)$. That is, evolutionary algorithms tackle to solve for $\hat{F}(X)$ instead of $f(X)$. Similar approximation is applied for the latter case.

B. Robust Solution of Salting Route Optimisation

In the case of Salting Route Optimisation, a robust solution can be represented by an optimal design value X for the following function:

$$F(X) = \int E(X, a)p(a)da, \quad (5)$$

where a indicates possible temperature distributions. Generally speaking, the optimisation of $F(X)$ does not make sense if a, such that $p(a) \neq 0$, is uncorrelated with X in $E(X, a)$. In the case of Salting Route Optimisation, potentially 'warmer' and 'colder' roads exist due to microclimatological effects caused by the local geography. However, although the distribution in tempearture will vary daily across the road network, warmer sections are nearly always warm and colder sections are nearly always cold. As a result, even on cold nights, some warmer sections of road do not require salt where as the colder sections of road will need treating on even the least marginal of nights.

It is still difficult to compute equation (5) exactly since the number of possible values in a is large and the probability distribution $p(a)$ is yet unknown. Hence, as in inductive learning, the number of typical temperature distributions is prepared for evolution. Let A_e be a set of temperature distributions for evolution. The following function is useful to evaluate the robustness of salting routes:

$$\hat{F}(X) = \sum_{a_i \in A_e} \frac{1}{|A_e|} E(X, a_i). \quad (6)$$

IV. Evolutionary Algorithms for Robust Solution of Salting Route Optimisation

A. Overview

In this section, a new Memetic algorithm for generating robust solutions is proposed. Fig. 3 shows the pseudo-code of the Memetic algorithm. The basic procedure of the

[1]The environmental parameters indicate parameters which characterise the fitness function

Procedure Robust Solutions of SRO by using EAs
begin
  translate SRO for temp. distrib. a_i to CARP ins. I_{a_i}
  initialise population
  evaluate population
  until *Stopping criterion is reached*
    select a CARP instance I_{a_i}
    pick up two parents
    generate 30 offspring by EAX on I_{a_i}
    select the best offspring
    apply local search to a copy of the best on I_{a_i}
    eval. the best and the improved copy by $\hat{F}(X)$
    replacement of new individuals
  end
end

Fig. 3. Pseudo-code of the proposed method

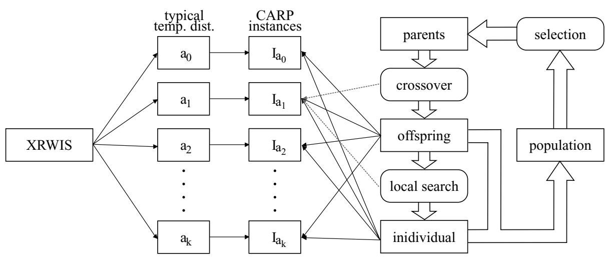

Fig. 4. The graphical user interface of the developed Salting Route Optimisation System

Memetic Algorithm is ordinal: selecting parents, reproducing offspring, applying local search to offspring and replacing the resultant offspring if the offspring is better than the worst individual in the population. A distinguished feature of the proposed method is that a crossover operation and local search methods are applied to only one CARP instance at every generation whilst the fitness function is composed of an ensemble of the evaluations of several CARP instances. Therefore, at the beginning of each generation, a CARP instance is selected for further genetic operations in the generation. This selection is carried out by referring to weights, which are also used to fitness evaluation. By selecting a CARP instance at every generation, the Memetic algorithm can concentrate to optimise each selected CARP instance. The weights are then updated for every predefined intervals of generations as described in section IV-C.

B. Coding Method and Fitness Evaluation

A naive permutation encoding method for solving SRO is employed. The chromosome of an individual is composed of several special symbols and edge IDs. Special symbols $\texttt{s}_1$ are used to indicate the beginning of tours for each truck. Using the following chromosome as example, this chromosome yields tours for two trucks (A permutation representaion is employed in this coding method).: $T_1 = \{$ `5 4 7 1`$\}$, and $T_2 = \{$ `8 3 2 6`$\}$.

`2 6` $\texttt{s}_1$ `5 4 7 1` $\texttt{s}_2$ `8 3`

As Evolutionary Algorithms have a tendency to find optimal solutions $E(X, a_i)$ in the case of equation (6), the normalised function $E_N(X, a_i)$ is employed as a fitness function of our Evolutionary Algorithms:

$$\hat{F}(X) = \sum_{a_i \in A_e} w_i E_N(X, a_i), \tag{7}$$

where w_i $(0 < w_i < 1, \sum_{a_i \in A_e} w_i = 1)$ denotes a weight for each temperature distribution a_i. The normalised function $E_N(X, a_i)$ is defined as follows:

$$E_N(X, a_i) = \frac{E(X, a_i) - E^*(a_i)}{E^*(a_i)}, \tag{8}$$

where $E^*(a_i)$ is a real number indicating the difficulty of solving a CARP instance I_{a_i}, for example, lower bounds which is the distance searched by other algorithms. In this algorithm, $E^*(a_i)$ is defined as the best distance for a CARP instance I_{a_i} searched by using the Memetic Algorithm [1].

C. Weights and their Update

In this section, two kinds of weights w_i are examined: fixed weights and weights updated for every predefined interval. The values of the fixed weights and the initial value of changed weights are set to $1/|A_e|$.

Weight updates correspond to change the directions of evolution of Evolutionary Algorithms. For every predefined interval of generations L, the weight w_j is updated by using the best tour evaluation $E_N^b(X, a_i)$ which is searched by Evolutionary Algorithms:

$$w_j = \frac{\exp E_N^b(X, a_j)}{\sum_{k=1}^{m} \exp E_N^b(X, a_k)} \tag{9}$$

Genetic operators including a crossover and local search methods are applied to only one CARP instance in every generation. A CARP instance is selected by referring to the weights in the beginning of every generation and applying genetic operators. For example, the following proportion s_i is used to randomly decide which CARP instance is applied to genetic operators, as in roulette wheel selection:

$$s_i = \frac{w_i}{\sum w_j}. \tag{10}$$

This application method is employed because the genetic operators adopted use information in problem instances, e.g., distance on edges, amount of services on edges, and so on, to improve the fitness of individuals. Conflicts between the improvement for problem instances by genetic operators might be occurred if genetic operators are applied to several problem instances simultaneously and therefore a conflict resolution mechanism may be required. Furthermore, EAs tend to converge the easier instances at first which can cause EAs to become trapped into the local minimum of the total problem as represented by equation (6) This is because it is difficult to find out an improved solution for the more difficult problem instances whilst the solution quality of easier problems is kept constant.

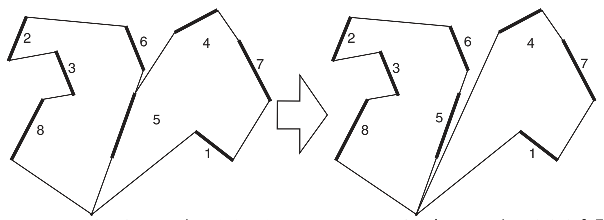

Fig. 5. A depiction of the repair procedure

D. Genetic Operators

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 [13], [14]. The EAX operator can solve Traveling Salesman Problems with 2393 cities with probabilities 90 % over. However, since this operator is specifically designed for Travelling Salesman Problems, it can often yield an infeasible solution. Hence, a repair operator for offspring individuals is incorporated in the Memetic Algorithm:

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 5). 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.

E. Local Search Method

Local search methods are carried out with a probability. Because the EAX operator has similar characteristics to the k-opt operator, three naive local search methods are used in the Memetic Algorithms:

Move an edge
Before: `4 s1 1 3 5 8 s2 2 6 7`
After: `4 s1 1 3 5   s2 2 6 8 7`
Move sequential two edges
Before: `4 s1 1 3 5 8 s2 2 6 7`
After: `4 s1 1 3     s2 2 6 5 8 7`
Swap two edges
Before: `4 s1 1 3 5 8 s2 2 6 7`
After: `4 s1 1 3 5 6 s2 2 8 7`

Upper and lower lines in each local search operation indicate an indvidual before and after applying the operation respectively. For all possbile pairs of variables, above local search operations are applied. A pair of variables with the best improvement is then selected.

F. Initialisation of Population

Almost all the individuals in the population are generated as a random permutation. Two kinds of additional sophisticated individuals are also inserted: Firstly, individuals generated by path scanning heuristics [7], [8] are applied to problem instances with large numbers of required edges.

Secondly, individuals indicating the best solution $E^*(a_i)$ for the corresponding CARP instance I_{a_i} is solely solved by Memetic algorithms in [1]. These individuals are represented by the IDs of required edges for corresponding CARP instance I_{a_i}. Therefore, individuals in the proposed method require the IDs of all edges. In order to insert missed IDs to the best individuals, simple heuristics based on the distance is employed: (1) select the nearest missed edge which is allowable to corresponding truck in the sense of service capacity, and insert selected edge to tours. (2) if no edge is found in (1), other tour is build by using the path scanning heuristics.

V. Experiments

A. Evolutionary process

A series of experiments on real data in South Gloucestershire was conducted. Ten different temperature distributions (shown in Table I) along with an 'ideal' temperature distribution a_{ideal} (where all temperature points are below 0 degree), were used for evolution. The number of trucks refers to the minimum number of trucks which can cover all the demands of required edges in corresponding CARP instance I_{t_i}. The best distance is searched by the Memetic algorithm in [1] and is used for the normalised evaluation $E_N(X, a_i)$ in equation (8) The number of edges in the area and trucks needed to serve in the CARP instance $I_{a_{ideal}}$ for the 'ideal' temperature distribution are 595 and 11 respectively. As a result, the string length is set to 696. The probability of applying local search method, the population size, and the number of generations in each run are 0.1, 300, and 100,000, respectively. Three kinds of changing intervals for weights are examined: 100, 500, and 1000.

Table II shows the average and the best distances over 20 runs by using the proposed method. Bold fonts denote that these results show statistical significance against uncharged weight method. Initialisation by incorporation of the best individuals for each CARP instance, which are searched by former Memetic algorithms, works well. For all weight change intervals and unchanged weights, the averages and the bests are improved.

Fig. 6 depicts the changes of weight during evolutionary process for that changing interval is 500 generations. The horizontal axis denotes the temperature distribution a_i where as the depth axis and the vertical axis indicate the number of generations and corresponding weight values. This graph elucidates that the proposed method appears to find better solutions in CARP instances with less required edges.

B. Test of the acquired robust solutions

The generalisation property of acquired robust solutions are investigated. A further 11 temperature distributions t_i

TABLE I

DETAILS OF PROBLEM INSTANCES FOR EVOLUTIONARY PROCESS

	a_0	a_1	a_2	a_3
No. required Edges	97	257	297	323
No. trucks	2	4	4	5
Best distance $E^*(a_i)$	268029	366410	436006	463263
	a_4	a_5	a_6	a_7
No. required Edges	354	437	428	519
No. trucks	6	7	8	9
Best distance $E^*(a_i)$	509717	576513	640502	665371
	a_8	a_9		
No. required Edges	568	578		
No. trucks	11	11		
Best distance $E^*(a_i)$	727262	747710		

TABLE II

EXPERIMENTAL RESULTS: AVERAGE AND BEST FITNESS

Initialisation	Change ivl.	Ave.	Best
unincorporated	n/a	0.3225	0.2894
	100	0.3233	0.2696
	500	0.3183	0.2883
	1000	0.3342	0.2985
incorporated	n/a	0.3002	0.2730
	100	**0.2860**	0.2652
	500	**0.2769**	0.2525
	1000	0.2916	0.2668

are used for this test. The meaning of each attribute is the same as Table I. The best solutions for algorithms in Table II are evaluated on CARP instance t_i in Table III.

Table IV summarises the results for the test. Numbers with bold fonts indicate the best performance among variable changing interval of weights, including uncharged weights. The 'average' column refers to the average value of $E_N(X, t_i)$ over 11 CARP instances I_{t_i}. For average values, the proposed method with changing interval $L = 500$ and best individual incorporation outperform the other algorithms. Especially, For CARP instances with less total demands, e.g.,

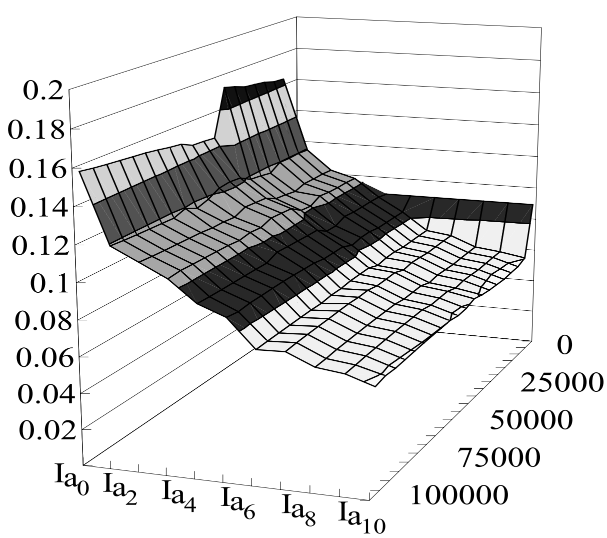

Fig. 6. Change of weights during evolution process

TABLE III

DETAILS OF PROBLEMS INSTANCES FOR TEST

	t_0	t_1	t_2	t_3
No. required Edges	129	246	262	259
No. trucks	3	4	4	4
Best distance $E^*(t_i)$	299189	349094	376647	377277
	t_4	t_5	t_6	t_7
No. required Edges	291	339	344	385
No. trucks	4	5	5	6
Best distance $E^*(t_i)$	422974	481216	482925	532018
	t_8	t_9	t_{10}	
No. required Edges	448	468	537	
No. trucks	8	8	10	
Best distance $E^*(t_i)$	578740	668918	671725	

$I_{t_0}, I_{t_1}, I_{t_2}, I_{t_3}$, the algorithm shows better performance as just 10 trucks are required.

C. Comparison with conventional routes

Finally, acquired robust solutions are compared with conventional routes which are actually used city council. Fig. 7 and Fig. 8 plot routes on CARP instance I_{t_1} and I_{t_8}, respectively. Gray line, coloured thick line and coloured thin line denote edges with no trucks, edges with service by corresponding truck, and deadheading edges, respectively. The hand-made conventional route is generated by dividing into several regions for each truck. By contrast, it might look that the robust solution by Memetic Algorithms is messy from bird's-view. However, the total distance for services in the case of Memetic Algorithms is more than 10 % improved.

VI. CONCLUSIONS

In this paper, a robust solution of salting route optimisation system, which combines Evolutionary Algorithms with the neXt generation Road Weather Information System, was proposed. A new framework of robust solutions not for perturbation but for various values of environmental parameters was also introduced. Weighting approach was examined in the proposed system in order to prevent to convergent to easier CARP instances. In addition, the insertion of the best individual for each temperature distribution was also incorporated to the proposed systems. In the sense of distances, the proposed system improved about 10 % of the one of conventional routes.

REFERENCES

[1] Handa, H, Chapman, L, Yao, X (2005) Dynamic Salting Route Optimisation using Evolutionary Computation, Proceedings of the 2005 Congress on Evolutionary Computation 1:158–165

[2] Chapman, L, Thornes, JE, and Bradley, AV (2001) Modelling of road surface temperature from a geographical parameter database. Part 1: Statistical, Meteorological Applications 8:409–419

[3] Chapman, L, Thornes, JE, and Bradley, AV (2001) Modelling of road surface temperature from a geographical parameter database. Part 2: Numerical, Meteorological Applications 8:421–436

[4] Chapman, L, Thornes, JE, Bradley, AV (2002) Sky-view factor approximation using GPS receivers, International Journal Climatology 22(5):615–621

[5] Chapman, L, Thornes, JE (2004) Real-Time Sky-View Factor Calculation and Approximation Journal of Atmospheric and Oceanic Technology 21:730–741

TABLE IV

RESULTS OF TEST

Initialisation		unincorporated				incorporated			
Change ivl.		n/a	100	500	1000	n/a	100	500	1000
Average		0.3225	0.3233	0.3183	0.3342	0.3002	0.2860	0.2769	0.2916
t_0	$E_N(X, t_0)$	0.8702	**0.8383**	0.8607	0.9086	0.8936	0.8191	**0.7997**	0.8302
	Distance	559550	549999	556708	571026	566551	544241	538455	547583
	No. Trucks	11	10	11	11	11	10	10	10
t_1	$E_N(X, t_1)$	0.5290	0.5610	**0.5281**	0.6207	0.6395	0.5677	**0.5416**	0.5822
	Distance	533775	544953	533453	565787	572348	547276	538166	552332
	No. Trucks	10	10	10	10	11	10	10	11
t_2	$E_N(X, t_2)$	0.6039	**0.5393**	0.6030	0.6887	0.6063	0.5660	**0.5041**	0.5802
	Distance	604089	579757	603767	636040	605015	589845	566504	595163
	No. Trucks	11	10	11	11	11	10	10	11
t_3	$E_N(X, t_3)$	0.6012	**0.5367**	0.6003	0.6859	0.6033	0.5632	**0.5013**	0.5762
	Distance	604089	579757	603767	636040	604891	589749	566408	594680
	No. Trucks	11	10	11	11	11	10	10	11
t_4	$E_N(X, t_4)$	0.5089	**0.4543**	0.5138	0.5917	0.5079	0.5268	**0.4949**	0.5199
	Distance	638232	615144	640314	673239	637787	645780	632284	642882
	No. Trucks	11	10	11	11	11	11	11	11
t_5	$E_N(X, t_5)$	0.3826	0.3916	**0.3802**	0.4652	0.3737	0.3714	0.3692	**0.3648**
	Distance	665324	669677	664168	705057	661054	659956	658886	656745
	No. Trucks	11	10	11	11	11	11	11	11
t_6	$E_N(X, t_6)$	0.3801	0.3906	**0.3753**	0.4624	0.3700	0.3677	0.3655	**0.3611**
	Distance	666486	671561	664168	706219	661615	660517	659447	657306
	No. Trucks	11	10	11	11	11	11	11	11
t_7	$E_N(X, t_7)$	0.3206	0.3418	**0.3055**	0.4186	0.2936	0.3086	0.2904	**0.2874**
	Distance	702595	713887	694553	754733	688219	696179	686511	684918
	No. Trucks	11	11	11	11	11	11	11	11
t_8	$E_N(X, t_8)$	0.2498	0.2592	**0.2439**	0.3451	0.2261	0.2249	**0.2141**	0.2174
	Distance	723293	728747	719915	778443	709600	708890	702662	704530
	No. Trucks	11	11	11	11	11	11	11	11
t_9	$E_N(X, t_9)$	0.1043	0.1260	**0.0971**	0.1868	0.0890	0.0882	**0.0850**	0.0880
	Distance	738676	753212	733860	793891	728464	727936	725755	727766
	No. Trucks	11	11	11	11	11	11	11	11
t_{10}	$E_N(X, t_{10})$	0.1081	0.1355	**0.1065**	0.1975	0.0882	0.0903	**0.0858**	0.1069
	Distance	744370	762735	743280	804358	730994	732408	729375	743514
	No. Trucks	11	11	11	11	11	11	11	11

[6] Golden, B, and Wong, RT (1981) Capacitated arc routing problem, Networks 11:305–315

[7] Golden, B, and DeArmon, JS, and Baker, EK (1983) Computational Experiments with Algorithms for a Class of Routing Problems, Computers and Operational Research 10(1):47–59

[8] Lacomme, P, Prins, C, and Ramdane-cherif, W (2004) Competitive Memetic Algorithms for Arc Routing Problems, Annals of Operations Research 131:159–185

[9] Tsutsui S, Ghosh A (1996) A Robust Solution Searching Scheme in Genetic Search, Parallel Problem Solving from Nature IV, 201–208

[10] Tsutsui S, Ghosh A (1997) Genetic Algorithms with a Robust Solution Searching Scheme, IEEE Trans on Evolutionary Computation 1(3):201–208

[11] Jin Y, Branke J (2005) Evolutionary Optimization in Uncertain Environments, IEEE Trans on Evolutionary Computation 9(3):303–317

[12] Moscato, P (1989) On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826

[13] Nagata, Y, 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 450–457

[14] Nagata, Y (2004) The EAX Algorithms Considering Diversity Loss Proceedings of the 8th International Conference on Parallel Problem Solving from Nature – PPSN VII 332–341

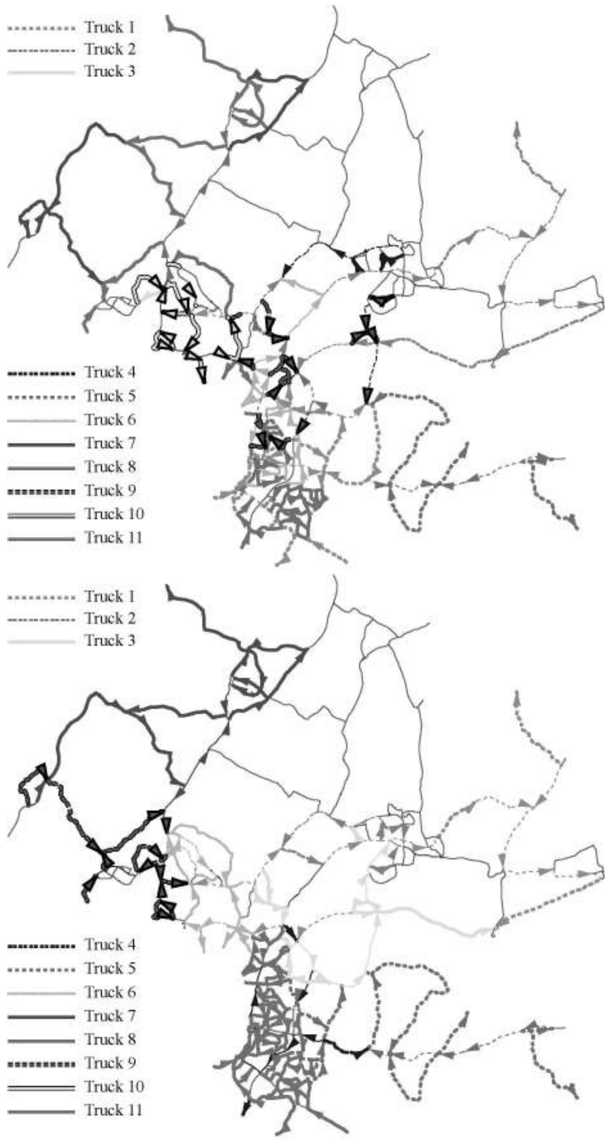

Fig. 7. Acquired routes (UPPER) and conventional routes (LOWER) for a CARP instance I_{t_1}

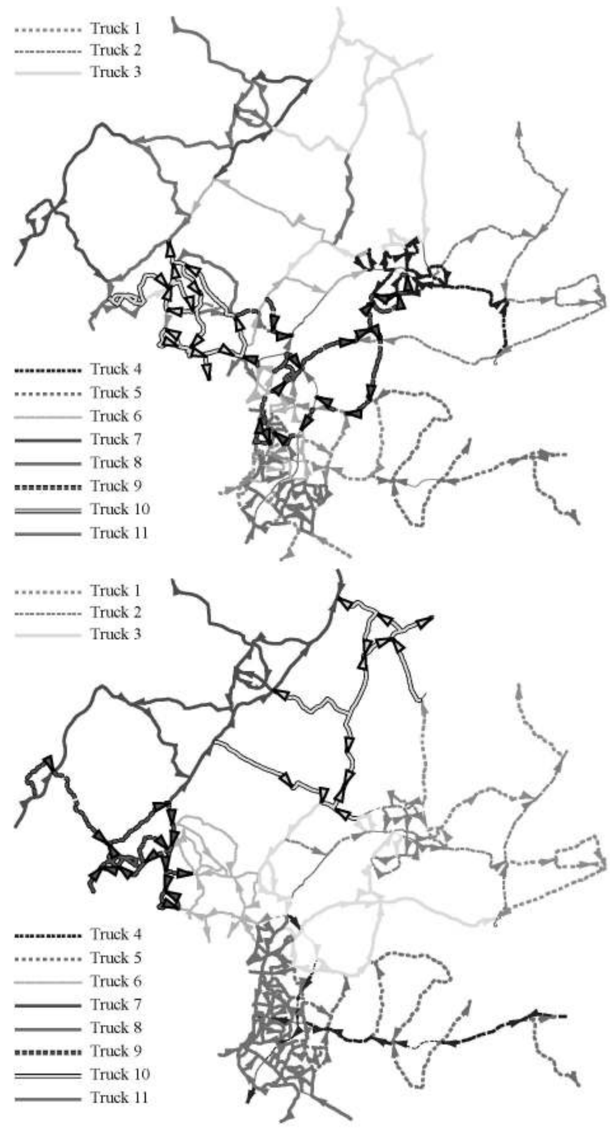

Fig. 8. Acquired routes (UPPER) and conventional routes (LOWER) for a CARP instance I_{t_8}