HYBRID ANT COLONY OPTIMIZATION FOR REAL-WORLD DELIVERY PROBLEMS BASED ON REAL TIME AND PREDICTED TRAFFIC IN WIDE AREA ROAD NETWORK

Junichi Ochiai[1] and Hitoshi Kanoh[2]

[1]Department of Computer Science, Graduate School of Systems and
Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan
ochiai@kslab.cs.tsukuba.ac.jp
[2]Division of Information Engineering, Faculty of Engineering, Information and
Systems, University of Tsukuba, Tsukuba, Ibaraki, Japan
kanoh@cs.tsukuba.ac.jp

ABSTRACT

This paper presents a solution to real-world delivery problems for home delivery services where a large number of roads exist in cities and the traffic on the roads rapidly changes with time. The methodology for finding the shortest-travel-time tour includes a hybrid meta-heuristic that combines ant colony optimization with Dijkstra's algorithm, a search technique that uses both real-time traffic and predicted traffic, and a way to use a real-world road map and measured traffic in Japan. Experimental results using a map of central Tokyo and historical traffic data indicate that the proposed method can find a better solution than conventional methods.

KEYWORDS

Ant Colony Optimization, Dijkstra's Algorithm, Delivery Problem, Traffic, Real-World

1. INTRODUCTION

Ant colony optimization (ACO) is a stochastic search algorithm for problem solving that takes inspiration from the foraging behaviors of ants. The main idea of ACO rests on the indirect communication among individuals in an ant colony based on the pheromone trails that real ants use for communication. ACO has been formalized into a meta-heuristic for combinatorial optimization problems by Dorigo et al., and many applications are now available [1]–[3]. In particular, many studies on ACO have been performed using the traveling salesman problem (TSP [4]), and it has been shown that ACO is superior to other meta-heuristics [5] [6] for this type of problem.

In this paper, we deal with real-world delivery problems (RWDPs) for home delivery services as an extension of the TSP, where a large number of roads exist in cities and the traffic on the roads rapidly changes with time. This scenario reflects the typical traffic congestion in a wide area urban road network.

David C. Wyld et al. (Eds) : CCSIT, SIPP, AISC, PDCTA, NLP - 2014
pp. 379–389, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4232

The problems that deal with finding optimal tours with time-dependent travel time have been studied as TSPs, delivery problems [7], and vehicle routing problems [8]. Conventional problem-solving methods using ACO repeat a search when the traffic flow changes during movement [9]–[11]. However, finding the global optimal solution by this method is difficult because when the traffic changes rapidly, the information obtained from an old search may not be helpful. Furthermore, research based on real road maps and traffic information services in the real world is seldom found.

In this paper, we propose a new method to solve RWDPs using ACO and Dijkstra's algorithm. (DA). Search techniques based on only the predicted traffic have previously been presented for real-world time-dependent vehicle routing problems using ACO [12], an evolution strategy [13], and a genetic algorithm [14]. When these methods make mistakes with the prediction values, solution accuracy may deteriorate. The proposed method aims at improving the accuracy by combining real-time data with predicted traffic data that can actually be obtained.

In the following section, we start by describing the problem. Then, we detail the algorithm of the proposed method. Finally, we present the results of experiments using a map of central Tokyo and real traffic data.

2. PROBLEM DESCRIPTION

In this paper, we regard the RWDP as an extension of the TSP: a vehicle starts from a depot, visits all customers without any time constraint, and finally returns to the depot. This type of problem is also called a one-to-many-to-one delivery problem [7]. Here, we first describe the time-dependent TSP (TDTSP) and the calculation of the tour travel time and then explain the RWDP and a traffic information service in the real world. Finally, a brief account of ACO will be given.

2.1. Time-Dependent Traveling Salesman Problem

The TSP [4] can be represented by a complete graph $G = (N, A)$, where N is a set of nodes, i.e. cities, $n = |N|$ is the number of nodes, and A is the set of arcs fully connecting the nodes. Each arc $(i, j) \in A$ is assigned a value $d_{i,j}$ $(=d_{j,i})$, which represents the distance between nodes i and j. The TSP then is the problem of finding the shortest closed tour that visits each of the nodes of G exactly once. The TSP instances used in this paper are taken from the TSPLIB benchmark library [15].

The TDTSP extends the original TSP so that traffic congestion can be included. Let $T_{i,j}(t)$ be the travel time between nodes i and j at time t; $T_{i,j}(0)$ means the original travel time of a given TSP, i.e. $T_{i,j}(0) = d_{i,j}$. Traffic congestion can be represented by a change in the travel time. Here, this change is defined by the following formula, where Δt is an updated interval of travel time, R_{jam} and B_{upper} are constants showing the rate and the upper bound of traffic congestion, respectively, and $rand \in [-1, 1]$ is a uniform random number. The time when a salesman leaves city 1 is set to $t = 0$, and he always starts from and returns to city 1.

$$T_{i,j}(t + \Delta t) = \left[T_{i,j}(t) \times (1 + R_{jam} \times rand)\right]_{T_{\min}}^{T_{\max}}$$

$$[x]_b^a = \begin{cases} a & if \;\; x > a \\ b & if \;\; x < b \\ x & \text{otherwise} \end{cases}$$

$$T_{max} = d_{i,j} \times B_{upper}$$
$$T_{min} = d_{i,j}$$

In the following, the time required to travel around a tour is called the tour travel time. The tour travel time for solution S can be calculated by the following formulas, where t_i is the time when a salesman reaches node i:

$$T(S) = \sum_{i=1}^{n} T_{i,i+1}(t_i)$$

$$t_i = \begin{cases} t_1 & \text{if } i = 1 \\ t_{i-1} + T_{i-1,i}(t_{i-1}) & \text{otherwise} \end{cases}$$

2.2. Real-World Delivery Problem

The RWDP can be represented by a quadruple $G^{RW} = (N^{RW}, A^{RW}, M^{RW}, T^{RW})$, where $N^{RW} = \{C_i \mid i = 1, ..., n\}$, C_1 is a depot and $\{C_2, ..., C_n\}$ is a set of customers, $A^{RW} = \{R_{i,j}(t_i) \mid i, j = 1, ..., n\ (i{\neq}j)\}$ is the set of optimal routes from C_i to C_j at time t_i when a vehicle reaches C_i, and M^{RW} and T^{RW} are a road map and a set of time-series traffic data in the real world, respectively. Each $R_{i,j}(t_i)$, generally $\neq R_{j,i}(t_i)$, can be calculated using T^{RW} on M^{RW}. It is necessary to calculate the optimal route between customers in RWDPs, while the distance between cities is given in TSPs. The RWDP then is the problem of finding a shortest-travel-time tour when a vehicle starts from C_1 and returns to C_1 visiting each of the customers $\{C_2, ..., C_n\}$ exactly once.

The road map M^{RW} used in this paper is the standard map database that is used in actual car navigation systems. This map includes all drivable roads in Japan and its format was developed and established by the Navigation System Researchers' Association.

Historical time-series traffic data T^{RW} are also used to calculate the travel time of a vehicle. In Japan, traffic meters are installed at more than 20,000 locations along principal roads throughout the country. These meters measure the average travel time of cars passing through specific road links at 5-minute intervals. The data so obtained is collected at a traffic information center and provided to subscribers in real time. Figure 1 shows an example of time-series traffic data. The vertical axis represents the average speed of cars on a link calculated directly from the traffic data. In this figure, data in the range of 0:00 to 5:00, 5:00 to 8:00, 8:00 to 18:00, and 18:00 to 20:00 correspond to no congestion, outbreak of congestion, heavy congestion, and dissolution of congestion, respectively. Thus, time-series traffic data is highly nonlinear, which makes it difficult to perform accurate predictions. Application to real-world scenarios must take the prediction error rate into consideration.

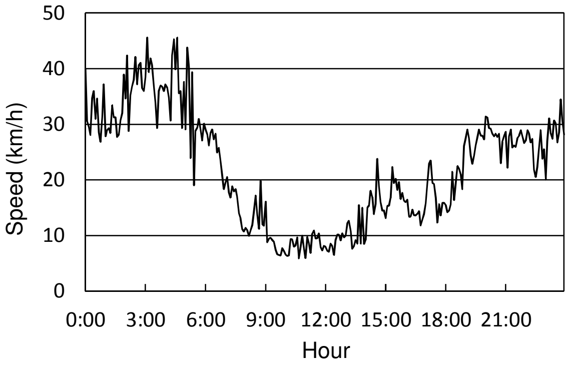


Figure 1. Example of time-series traffic data.

2.3. Ant Colony Optimization

The generic ACO meta-heuristic [1] is shown below. After initialization, the meta-heuristic iterates over two phases. First, a number of solutions are constructed by the ants, and second, the pheromone trails are updated.

```
Procedure ACO ( )
    Set parameters;
    Initialize pheromone trails;
    While (terminal condition not met) {
        Construct ant solutions;
        Update pheromone;
    }
```

There have been many attempts to improve the performance of ACO. Max-Min Ant System (MMAS) has demonstrated an especially impressive performance [1], so we used the MMAS, except for constructing the ant solution described in section 3.3, as the ACO in the proposed method.

3. PROPOSED METHOD

3.1. Configuration of System and Data Flow

Figure 2 shows the configuration and the data flow of the proposed system. A target road map including the depot and all customers is prepared along with the historical traffic data for principal roads on the road map. The latter are required for the prediction. The real-time traffic data for the principal roads are input at Δt intervals, so updating the travel time and recalculating the predicted traffic are performed at this interval. The prediction system, as we have already reported [17] [18], has an interpolation function as well as a prediction function. This system can estimate traffic on roads not installed with detectors from the traffic on roads installed with detectors. The other systems and data in Fig. 2 are described in the following sections.

3.2. General Procedure

The general procedure of the proposed method is shown in Fig. 3. The index i in the outermost loop corresponds to the turn at which a vehicle visits customers. While the vehicle is moving, the search is repeated. When the vehicle reaches the next customer, if new traffic data is available, a new tour is constructed, where k is an ant number, $C_{l(i)}$ is the i-th customer to which the ant moves, S^{ib} is the best solution in the current iteration, and S^{gb} is the best solution found since the first iteration. Constructing a tour by hybrid ACO is described in the next section.

3.3. Constructing a Tour by Hybrid ACO

In the proposed method, the routes between customers are planned by Dijkstra's algorithm (DA) and the turn of visiting customers is constructed by ACO. The DA is widely used as a route planning method, and ACO is superior to other meta-heuristics including genetic algorithms [5] and simulated annealing [6] in terms of constructing a tour.

When the ant k is at the customer C_i in constructing a tour, the customer to which the ant k moves next can be selected as follows.

[Step 1] The DA calculates the shortest-travel-time routes using predicted traffic for all roads at time t_i: $\{R_{i,j}(t_i) \mid C_j \in N^k\}$, where N^k is the set of customers not yet visited by the ant k and t_i is the time when a vehicle reaches C_i.

[Step 2] Selection probabilities for all $C_j \in N^k$ are calculated by the following formulas, where $\tau_{i,j}$ is a pheromone on $R_{i,j}(t_i)$, $\eta_{i,j}(t_i)$ is a heuristic value, $T_{i,j}(t_i)$ is a travel time between C_i and C_j, and α and β are constants.

$$p_{ij}^k(t_i) = \frac{[\tau_{i,j}]^\alpha \cdot [\eta_{i,j}(t_i)]^\beta}{\sum_{l \in N^k} [\tau_{i,l}]^\alpha \cdot [\eta_{i,l}(t_i)]^\beta}$$

$$\eta_{i,j}(t_i) = \frac{1}{T_{i,j}(t_i)}$$

[Step 3] Select a customer according to the probabilities above.

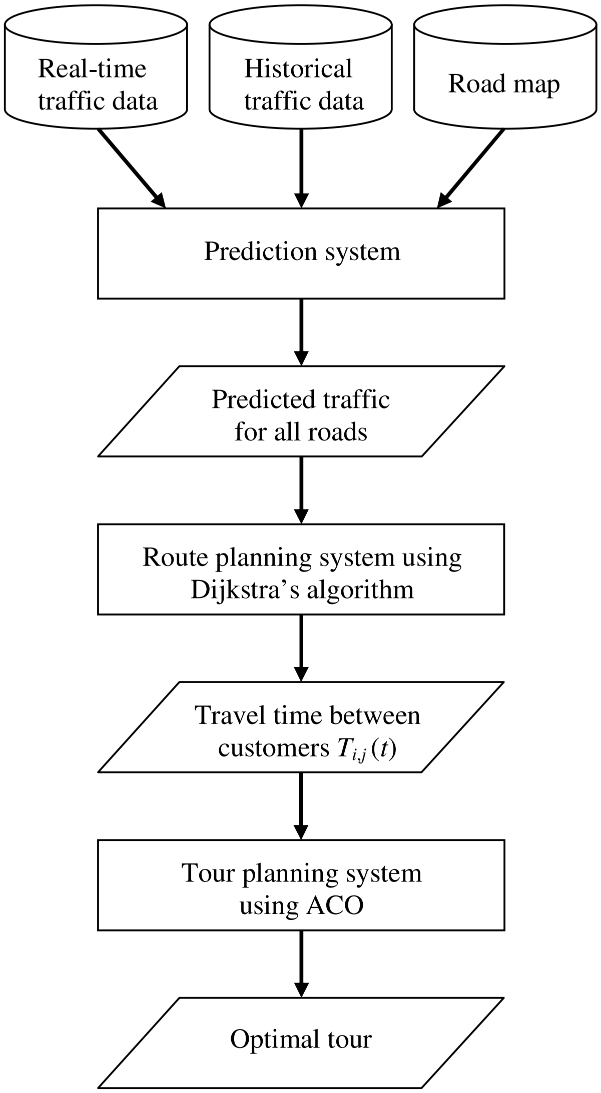

Real-time traffic data
Historical traffic data
Road map
Prediction system
Predicted traffic for all roads
Route planning system using Dijkstra's algorithm
Travel time between customers $T_{i,j}(t)$
Tour planning system using ACO
Optimal tour

Figure 2. Configuration and data flow of proposed system.

```
Procedure main ( )
  Input RWDP;
  for ($i$ = 1 to $n$-1) {
    if ($\Delta t$ passed or $i$ = 1) {
      Input $T^{RW}$(real-time);
      Predict travel time for all roads on $M^{RW}$;
      While (terminal condition not met) {
        for ($k$ = 1 to $n$) {
          Construct tour from $C_{l(i)}$ to $C_{l(n)}$ by hybrid ACO (=$S^k$);
          Plan route from $C_{l(n)}$ to depot by DA (=$R^k$);
          $S^k \leftarrow S^k + R^k$;
        }
        $S^{ib} = \arg\min_k T(S^k)$;
        if ( $T(S^{ib}) < T(S^{gb})$ ) $S^{gb}$=$S^{ib}$;
        Update pheromone;
      }
    }
    Move vehicle to next customer according to $S^{gb}$;
  }
  Move vehicle to depot according to $S^{gb}$;
```

Figure 3. General procedure of proposed method.

4. EXPERIMENTS

4.1. Experiments with TSP Instances (TDTSP)

4.1.1. Experimental Methods

To evaluate how well the proposed method performs, we first conducted experiments using the TSP instances eil51, eil76, kroA100, u159, and d198 from the TSPLIB [15]. The number of cities seems small in TSPLIB. However, considering correspondence with real-world problems, since the number of cities (or customers) that a salesman (or a vehicle) can visit in one day is at most 200, this number should be suitable as a scale for benchmark problems.

TDTSPs were generated by the method described in Section 2.1. Constants R_{jam} and B_{upper} were respectively set to 0.5 and 5 by reference to real-world traffic. Table 1 shows the number of cities, the optimal solution known in a static environment, the update interval of travel time in a dynamic environment, and the number of updates for each instance. Minutes and seconds are assumed as units of travel time for instances {eil51, eil76} and {kroA100, u159, d198}, respectively. The number of updates for each optimal solution was as shown in Table 1. The number of updates ranged from 53 to 140—in other words, the traffic flow was changed very frequently.

We compared the minimal tour travel time obtained by the proposed method with those by the conventional methods below.

- Plain method: Search is conducted once before a vehicle starts using static traffic $T_{i,j}$ (0).
- Repeat method [9]–[11]: Search is repeated while a vehicle is moving using real-time traffic $T_{i,j}$ (t).
- Prediction method [12] [19]: Search is conducted once before a vehicle starts using predicted traffic $T_{i,j}$ (t_i).
- Prediction and repeat method: The proposed method.

We assumed two kinds of prediction errors, i.e., 2% and 5% in Fig. 4. The predicted traffic with error rate can be calculated as follows, where *error*(t) is given by Fig. 4, t is the current time, and $rand \in [-1, 1]$ is a uniform random number:

$$T'_{i,j}(t_i) = T_{i,j}(t_i) \times (1 + error(t_i) \times rand) \quad \text{for prediction method}$$
$$T'_{i,j}(t, t_i) = T_{i,j}(t_i) \times (1 + error(t - t_i) \times rand) \quad \text{for proposed method}$$

This is based on experience in which short-term prediction has a smaller error rate than long-term prediction. For each solution, i.e. tour, obtained in this way, the tour travel time was calculated using predicted traffic without error. In addition, the values of parameters of the MMAS were those generally used [1].

Table 1. TSP instances used in experiments and updated travel time.

TSP	**Number of cities**	**Optimal solution**	**Update interval**	**Number of updates**
eil51	51	426	5	86
eil76	76	538	5	108
kroA100	100	21282	300	71
u159	159	42080	300	140
d198	198	15780	300	53

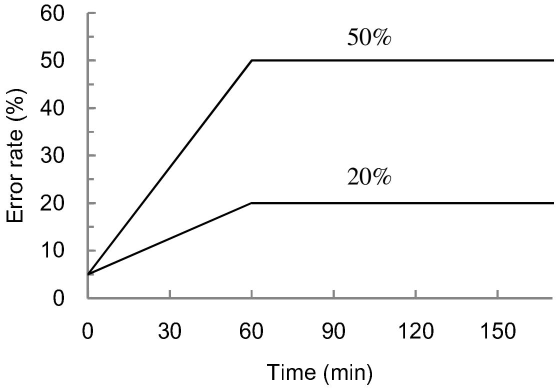


Figure 4. Relationship between prediction error and time.

4.1.2. Experimental Results

Table 2 lists the ratio of the tour travel time of each method to the optimal tour travel time. Since the optimal solutions of TDTSPs are not known, we assumed that the solution obtained by the prediction method without error, which knows the exact travel time on all the links at all times, is the optimal solution. Each value in Table 2 is the average of 30 trials using different random number sequences; the standard deviation was about 5 to 10%. Table 2 reveals three main findings.

- Compared with the plain method, the performances of the other methods are improved.
- When the prediction error rate becomes large (50%), the prediction method is inferior to the repeat method.
- The proposed method is superior to the other methods, even when the error rate is large (50%).

Table 2. Experimental results for TDTSP. Each value indicates the ratio of the tour travel time of each method to the optimal tour travel time.

TSP	Plain	Repeat	Prediction		Prediction + Repeat	
			20%	50%	20%	50%
eil51	1.50	1.31	1.05	1.41	1.02	1.12
eil75	1.39	1.36	1.29	1.44	1.06	1.10
kroA100	1.27	1.18	1.16	1.17	1.03	1.05
u159	1.26	1.27	1.29	1.39	1.03	1.08
d198	1.55	1.24	1.06	1.26	1.03	1.04
(Mean)	1.39	1.27	1.17	1.33	1.03	1.08

4.2. Experiments with Real-World Problems (RWDP)

4.2.1. Experimental Methods

Next, to evaluate the proposed method in a real-world environment, we applied it to the RWDP described in section 2.2. Figure 5 shows a map of central Tokyo, which was the target area of this experiment. This map includes 58,222 links and 19,963 nodes and represents the most congested area in Japan. The positions of 100 customers are randomly selected from the nodes. The travel time of a vehicle was calculated from historical traffic data on June 17, 2003. An example of the data is shown in Fig. 1. We also assumed the prediction error rate in Fig. 4. We performed three experiments with the starting time of a vehicle at 6:00 (morning), 12:00 (afternoon), and 18:00 (night).

4.2.2. Experimental Results

Figure 6 shows examples of the tours obtained by the proposed method. The red and black circles indicate a depot and customers, respectively. As shown, the tour depends on the time period. Crosses and returns appear in some parts of the tour in 2D, but in the 3D real-world, the tour is a complete circuit. Table 3 lists the tour travel time (minutes) of each method. Each value in Table 3 is the average of 30 trials using different random number sequences, and the standard deviation was about 2 to 5%. Table 3 reveals two main findings.

- The same results as the experiments for TDTSPs can be obtained in a real-world environment.
- The performance of the proposed method does not deteriorate when the prediction error rate is large (50%), while that of the prediction method deteriorates.

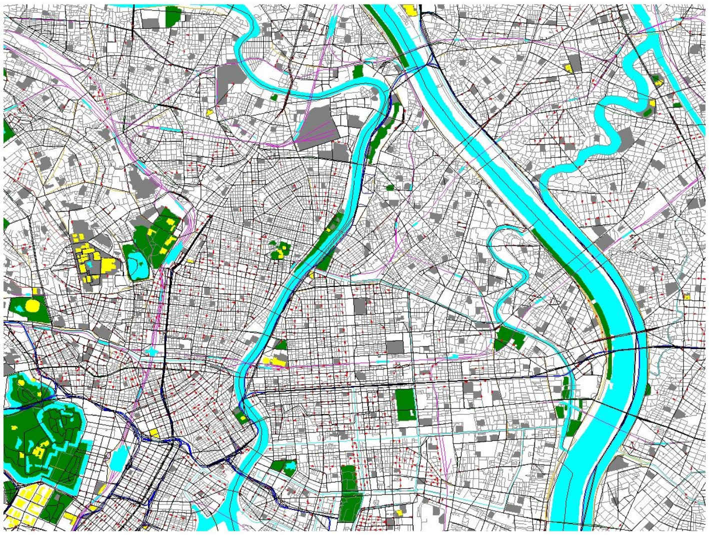
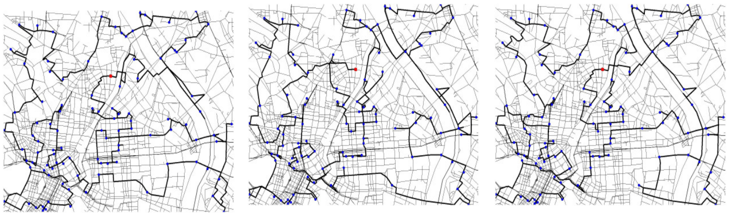


Figure 5. Map of central Tokyo. There are 58,222 links and 19,963 nodes.



(a) Morning (b) Afternoon (c) Evening

Figure 6. Examples of tours obtained by proposed method.

Table 3. Experimental results for RWDP. Each value is shown in minute.

Time	**Plain**	**Repeat**	**Prediction**		**Prediction + Repeat**	
			20%	**50%**	**20%**	**50%**
Morning	371	336	308	323	294	294
Afternoon	349	341	327	336	316	320
Night	281	266	266	280	265	269

5. CONCLUSIONS

In this paper, we presented three techniques: a hybrid meta-heuristic that combines ACO with Dijkstra's algorithm, a search method that combines repeat and prediction, and a way to use a real-world road map and measured traffic data in Japan. The experimental results suggest that the proposed method is effective in a wide area road network. The results presented in this paper are based on five benchmark problems and a real-world problem. Further investigation using other maps and traffic from additional days is necessary. Although the proposed method is for delivery problems, the basic idea can be used for other combinatorial optimization problems in networks. In future work, we will improve the ACO model and shorten the computational time.

ACKNOWLEDGEMENTS

The authors wish to thank Mr. Yosuke Kameda for his considerable advice in carrying out this research. This research was partly supported by a Grant-in Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (23500169).

REFERENCES

[1] Dorigo, M., Stutzle, T. , (2004) Ant colony optimization, The MIT press.

[2] Dorigo, M., Birattari, M., Stutzle, T., (2006) "Ant colony optimization – Artificial ants as a computational intelligence technique", IEEE Computational Intelligence Magazine. Vol. 1, No. 4, pp 28-39.

[3] Dorigo, M., Stutzle, T., (2010) "Handbook of Metaheuristics", International Series in Operations Research & Management Science 146, pp 227-263.

[4] Reinelt, G., (1994), "The Traveling Salesman: Computational Solution for TSP Applications", Lecture Notes in Computer Science 840.

[5] Albayrak, M., Allahverdi, N., (2011) "Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms", Expert Systems with Applications 38, pp 1313-1320.

[6] Genga, X., Chen, Z., Yang, W., Shi, D., Zhao, K., (2011) "Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search", Applied Soft Computing 11, pp 3680-3689.

[7] Berbeglia, G., Cordeau, J., Laporte, G., (2010) "Dynamic pickup and delivery problems", European Journal of Operational Research 202, pp 8-15.

[8] Pillac, V., Gendreau, M., Gueret, C., Medaglia, A. L., (2013) "A review of dynamic vehicle routing problems", European Journal of Operational Research 225, pp 1-11.

[9] Mavrovouniotis, M., Yang, S., (2010) "Ant Colony Optimization with Immigrants Schemes in Dynamic Environments", Lecture Notes in Computer Science, vol. 6239, pp 371-380.

[10] Guntsch, M., Middendorf, M., (2001) "Pheromone modification strategies for ant algorithms applied to dynamic TSP", Lecture Notes in Computer Science 2037, pp 213–222.

[11] Eyckelhof, C.J., Snoek, M., (2002) "Ant systems for a dynamic TSP: ants caught in a traffic jam", Lecture Notes in Computer Science 2463, pp 88-99.

[12] Donati, V.A., Montemanni, R., Casagrande, N., Rizzoli, E.A. and Gambardella, M. L., (2008) "Time dependent vehicle routing problem with a multi ant colony system", European Journal of Operational Research, Vol. 185, No. 3, pp 1174-1191.

[13] Kanoh, H., Tsukahara, S., (2010) "Solving Real-World Vehicle Routing Problems with Time Windows using Virus Evolution Strategy", International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol. 14, No. 3, pp 115-126.

[14] Haghani, A. and Jung, S., (2005) "A dynamic vehicle routing problem with time-dependent travel times", Computers & Operations Research, Vol. 32, No. 11, pp 2959-2986.

[15] Traveling Salesman Problem (TSPLIB). http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

[16] Stutzle, T., Hoos, H. H., (2000) "MAN-MIN ant system", Future Generation Computer System, Vol. 16, No. 8, pp 889–914.

[17] Kanoh, H, Furukawa, T., Tsukahara, S., Hara, K., Nishi, H., Kurokawa, H., (2005) "Short-Term Traffic Prediction Using Fuzzy C-Means and Cellular Automata in a Wide-Area Road Network", IEEE International Conference on Intelligent Transportation Systems (ITSC 2005), pp984-988.

[18] Ichiba, D., Hara, K., Kanoh, H., (2006) "Spatial Interpolation of Traffic Data by Genetic Fuzzy System", IEEE 2nd International Symposium on Evolving Fuzzy Systems (EFS 2006), pp 280-285.

[19] Kanoh, H, Ochiai, J., (2012) "Solving Time-Dependent Traveling Salesman Problems using Ant Colony Optimization Based on Predicted Traffic", International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2012), Advances in Intelligent and Soft Computing 151, pp 25-32.

AUTHORS

Junich Ochiai received M.E. degree in computer science from University of Tsukuba, Japan, in 2011. He is with the Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include ant colony optimization and evolutionary computation.

Hitoshi Kanoh received M.S. degree in physics and Ph.D. degree in computer science from University of Tsukuba, Japan, in 1980 and 1992, respectively. In 1980, he joined Hitachi Cable, Ltd. where he has been engaged in Research of expert systems, fuzzy systems and neural networks as a section chief. In 1993 he moved to the University of Tsukuba. He currently is a professor with the Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba. His main research interests include evolutionary computation, swarm intelligence, and soft computing with special application to intelligent transportation systems.