Computing and Informatics, Vol. 30, 2011, 721–732

A CENTROID-BASED HEURISTIC ALGORITHM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Kwangcheol Shin

Department of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
119, Moonji-Ro, Yuseong-Gu, Daejeon, 305-732, Korea
e-mail: kcshin2005@gmail.com

Sangyong Han*

School of Computer Science and Engineering
Chung-Ang University
221 Heuksuk-Dong, Dongjak-Gu, Seoul, 156-756, Korea
e-mail: hansy@cau.ac.kr

Communicated by Hong Zhu

Abstract. The vehicle routing problem (VRP) is famous as a nondeterministic polynomial-time hard problem. This study proposes a centroid-based heuristic algorithm to solve the capacitated VRP in polynomial time. The proposed algorithm consists of three phases: cluster construction, cluster adjustment, and route establishment. At the cluster construction phase, the farthest node (customer) among un-clustered nodes is selected as a seed to form a cluster. The notion of the geometrical centre of a cluster is introduced in this study to be utilized at the cluster construction and the cluster adjustment phases. The proposed algorithm has a polynomial time complexity of $O(n^{2.2})$. Experimental results on Augerat benchmark dataset show that the proposed 3-phase approach can result in smaller distances than the Sweep algorithm in more cases.

Keywords: Vehicle routing problem, heuristics, cluster-first route-second

* corresponding author

1 INTRODUCTION

The vehicle routing problem (VRP) is famous as a NP-hard problem. The VRP can be stated as follows: Generate a sequence of deliveries for each vehicle in a homogeneous fleet based at a single depot so that all customers are serviced and the total distance travelled by the fleet is minimized. Each vehicle has a fixed capacity and must leave from and return to the depot. Each customer has a known demand and is serviced by exactly one visit of a single vehicle. The capacitated VRP is like a VRP with an additional constraint that every vehicle must have a uniform capacity for a single commodity. The VRP and CVRP are addressed by many researchers. Their research efforts can be sorted as exact approaches and heuristic methods. Since exact approaches try to find the optimal solution, they take much time that cannot be applied in the real world. Heuristic or meta-heuristic methods trying to find optimal or near optimal routes in modest time are more actively researched recently.

In this work, a novel heuristic method is propsed to solve the CVRP with a polynomial time complexity of $O(n^{2.2})$. The proposed algorithm consists of three phases: first, nodes (customers) are clustered into a feasible solution by selecting the farthest node as a seed of a cluster. Second, nodes are moved into the closest cluster by utilizing the notion of the centre of a cluster. Finally, the actual routes are established by applying a travelling salesman problem (TSP) algorithm on the clusters produced in phases one and two.

This study is organized as follows: previous works are reviewed in Section 2, and in Section 3, the CVRP, the objective function of this study, and the notion of the geometrical centre of a cluser are explained. The proposed 3-phase algorithm is explained in Section 4. In Section 5, the experimental results are illustrated and conclusions are made with the directions for future works in Section 6.

2 LITERATURE REVIEW

To solve the VRP, there were two types of research. One involved exact methods of finding the optimal solution by computing all possible solutions and the other was heuristic or meta-heuristic approaches, which performed a relatively limited exploration of the search space and typically produced good quality solutions with modest computing times.

Exact methods can be classified into the following categories: dynamic programming, set partitioning, branch-and-bound, and branch-and-cut. Until recently, exact methods for the CVRP have been dominated by branch-and-cut. One of the best branch-and-cut algorithms was developed by Lysgaard et al. [1]. Recent research results indicated that branch-and-cut-and-price algorithms were more promising approaches for the CVRP as Fukasawa et al. [2] showed.

Heuristic methods can be broadly classified into two main classes: classical heuristics developed mostly between 1960 and 1990, and meta-heuristics actively researched from the 1990's. The Clarke and Wright saving algorithm [3] was a representative classical heuristic approach applied to problems where the number of vehicles was not fixed. Several enhancements to the Clarke and Wright algorithm were proposed [4, 5], which aimed at reducing computation time and memory requirements. Insertion heuristic was another well known classical method; Mole and Jameson [6] expanded one route at a time and Christofides et al. [7] applied, in turn, a sequential and a parallel route construction procedure. A cluster-first, route-second heuristic was introduced by the Sweep algorithm [8, 9] where feasible clusters were initially formed by rotating a ray centered at the depot and a vehicle route was then obtained for each cluster by solving a travelling salesman problem (TSP). Fisher and Jaikumar [10] also tried to solve the VRP by a cluster-first, route-second algorithm. They solved a Generalized Assignment Problem (GAP), instead of using a geometric method to form the clusters. Bramel and Simchi-Levi [11] described a two-phase heuristic where the seeds were determined by solving capacitated location problems and the remaining vertices were gradually included into their allotted route in a second stage.

In meta-heuristics, the emphasis was on performing a deep exploration of the most promising regions of the solution space. Lin [12] introduced λ-opt mechanism to describe the improvement procedure. Here, λ edges were removed from the tour and the λ remaining segments were reconnected in all possible ways. Lin and Kernighan [13] modified λ dynamically throughout the search and Or [14] proposed the Or-opt method consisting of displacing strings of three, two, or one consecutive vertices to another location. Bullnheimer et al. [15] suggested the first ant system for the CVRP. They improved their ant system by using two basic ant system phases: the construction of vehicle routes and the trail update [16]. Baker and Ayechew [17] considered the application of a genetic algorithm (GA) to the basic vehicle routing problem, where customers having a known demand were supplied from a single depot. The basic concept of a Tabu Search (TS), as described by Glover [18], was a meta-heuristics superimposed on another heuristics. The TS explored the solution space by moving at each iteration from a solution S to the best solution in a subset of its neighborhood. Generally the quality of solutions produced by these meta-heuristics is much higher than that obtained by classical heuristics but it is important to provide the good initial solution found by classical heuristic to meta-heuristic algorithm. In this paper, a cluster-first, route-second heuristic algorithm to find solutions minimizing travelling distances with an unfixed number of vehicles is proposed. The proposed algorithm consists of three phases which utilize the notion of geometrical centre of a cluster. The result of the proposed algorithm can be used as an initial solution for meta-heuristic algorithms.

3 DEFINITION

In this section, the capacitated VRP and the objective function that this study tries to solve are defined and the notion of the geometrical centre of a cluster introduced in this study is explained.

3.1 CVRP and Objective Function

The capacitated vehicle routing problem (CVRP) is defined as follows: An undirected graph $G = (V, E)$ is given where $V = \{0, 1, \cdots, n\}$ is the set of $n + 1$ nodes and E is the set of edges. Node 0 represents the depot and the remaining node set $V' = V - \{0\}$ corresponds to n customers. Every edge $\{i, j\} \in E$ is assigned a nonnegative cost $c_{i,j}$. $i \in V'$ is used to refer both to a customer and to its node location. Each customer $i \in V'$ requires a supply of q_i units from the depot. A set of M identical vehicles of capacity Q is stationed at the depot. A feasible solution of the CVRP is composed of: a partition $S = \{r_1, r_2, \cdots, r_m\}$ of V and a permutation σ_i of r_i specifying the order of the customers on route with the restriction that the total demand of all customers supplied on a route r_i does not exceed the vehicle capacity Q. The cost of a given route $r_i = \{v_0, v_1, \cdots, v_{k+1}\}$ where $v_i \in V$ and $v_0 = v_{k+1} = 0$ (0 denotes depot), is given:

$$C(r_i) = \sum_{i=0}^{k} c_{i,i+1}.$$

The total cost of solution S is:

$$TC(S) = \sum_{i=0}^{m} C(r_i).$$

The objective function can be described as designing routes so that all customers are visited exactly once with the minimun cost of the solution regardless of the number of vehicles being used.

3.2 Geometrical Centre of a Cluster

The notion of the geometrical centre (GC) of a cluster used in the proposed algorithm can be defined as follows:

Let $l_i = \{v_0, v_1, \cdots, v_k\}$ be cluster i, where v_j is a member of cluster i.

$$GC(l_i) = \left(\sum_{j=0}^{k} v_j^x / k, \sum_{j=0}^{k} v_j^y / k \right), \text{where } v_j^x \text{ and } v_j^y \text{ are } x \text{ and } y \text{ coordinates of } v_j \quad (1)$$

Figure 1 shows the notion of GCs of clusters. (Circles and diamonds represent seperately the nodes of the same cluster.)

4 PROPOSED 3-PHASE ROUTING ALGORITHM

The proposed algorithm is composed of three parts: cluster construction, cluster adjustment, and route establishment. In this section, these three parts are explained in detail.

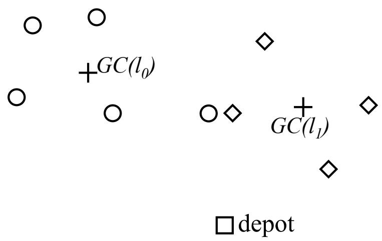


Fig. 1. The geometrical centre of a cluster

4.1 Cluster Construction

First, clusters are constructed by selecting the farthest node from the depot as a cluster seed. The reason of choosing the farthest node as a seed is based on our intuition that the farthest node is the critical point where other nodes can be served along with the route visiting the node.

Once the farthest node, v_i, from the depot is found, the first cluster l_0 is formed with v_i and then the geometrical centre of the cluster, $GC(l_0)$, is calculated by using formula (1). To add nodes to l_0, the cluster construction algorithm finds v_j among un-clustered nodes, which is located closest from $GC(l_0)$, and includes v_j to l_0 only if the demand of v_j does not exceed the available truck capacity of l_0. If v_j is added to cluster l_0, the truck capacity of l_0 is reduced by the demand of v_j and $GC(l_0)$ is recalculated. The same processes above are conducted until the available truck capacity of l_0 becomes smaller than the demand of the closest node from $GC(l_0)$. If the demand of v_j exceeds the available truck capacity of l_0, the algorithm stops to expand l_0, and finds the farthest node, v_k, from the depot among un-clustered nodes again in order to generate another cluster, l_1, with v_k. These processes are repeated until no unvisited node exists. Table 1 shows pseudo code of the cluster construction phase. The time complexity of this phase is $O(n^2)$.

4.2 Cluster Adjustment

Once clusters are constructed, the cluster adjustment algorithm is applied to optimize the clusters. Cluster adjustment means that if node v_k, which belongs to cluster l_i, is closer to $GC(l_j)$ than $GC(l_i)$, and the demand of v_k does not exceed the available capacity of l_j, then move v_k from l_i to cluster l_j. If a node moves from l_i to l_j, $GC(l_i)$ and $GC(l_j)$ are also recalculated. Figures 2 and 3 show the cluster adjustment processes. In Figure 2, v_k is a member of l_0 but it is closer to $GC(l_1)$ than $GC(l_0)$. If the available capacity of l_1 is equal to or bigger than the demand of v_k, v_k moves from l_0 to l_1 and $GC(l_0)$ and $GC(l_1)$ are recalculated. Table 2 shows the pseudo code of the cluster adjustment phase. The time complexity of this phase is also $O(n^2)$.

```
Set i = 0 and Q = truck_capacity
While (unvisited node exist)
    v_j = the farthest node among un-clustered nodes from the depot
    Generate cluster l_i with v_j
    Set capacity of l_i = Q
    While (demand of v_j does not exceed available capacity of l_i)
        Add v_j to l_i
        Reduce available capacity of l_i by demand of v_j
        Calculate GC(l_i)
        v_j = the closest node among un-clustered nodes from GC(l_i)
    End while
    i = i+1
End while
```

Table 1. Pseudo code for the cluster construction

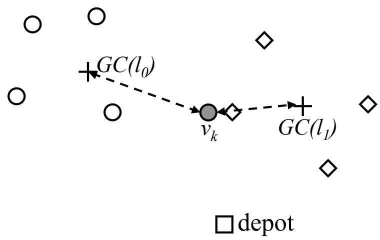


Fig. 2. Before cluster adjustment

4.3 Route Establishment

After the clusters are adjusted in the second phase, finally the route establishment algorithm is applied. Since route establishment means finding the shortest path of visiting every node once in a cluster, an algorithm for the travelling salesman problem (TSP) can be applied in this phase. The Lin-Kernighan heuristic [13] is used in this work to find the shortest path because even though the algorithm is approximate, optimal solutions are very quickly produced with the time complexity of $O(n^{2.2})$.

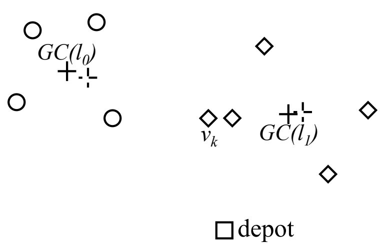


Fig. 3. After cluster adjustment (GC of each cluster is moved)

```
Set cluster_group = {l_0, l_1, ···, l_m}
For i = 0 to m repeat
    For every node v_k in l_i
        For every l_j in cluster_group
            If (i ≠ j and v_k is closer to GC(l_j) than GC(l_i) and available
              capacity of l_j is equal to or bigger than demand of v_k)
                Move v_k from l_i to l_j
                Recalculate GC(l_i) and GC(l_j)
            End if
        End for
    End for
End for
```

Table 2. Pseudo code for the cluster adjustment

4.4 Combined Proposed Algorithm and Its Time Complexity

Table 3 shows the combined whole algorithm. The route establishment algorithm is applied after the cluster construction and the cluster adjustment algorithms. Depot is added to all clusters just before the route establishment since a vehicle should leave from and return to depot. The result at each iteration is recorded in oder to find the best one at the end of the algorithm.

The combined algorithm has polynomial time complexity of $O(n^{2.2})$ since, as shown in Tables 1 and 2, cluster construction and cluster adjustment phases have $O(n^2)$ each and route establisment phase (Lin-Kernighan heuristic) has $O(n^{2.2})$.

```
Cluster_Construction_Phase()
Add depot to the clusters
Get result by applying TSP(clusters)
Remove depot from the clusters
Do
    If (no node move in Cluster_Adjustment_Phase())
        Break while-loop
    Else
        Add depot to the clusters
        Get result by applying TSP(clusters)
        Remove depot from the clusters
    End if
While (TRUE)
Print the smallest result as final among results
```

Table 3. Pseudo code for the whole algorithm

Instance	Best known	Centroid-based 3-phase		Sweep + cluster_adjustment	
		Distance	Routes	Distance	Routes
A-n32-k5	784	881	5	872	5
A-n33-k5	661	728	5	788	5
A-n33-k6	742	770	6	829	7
A-n34-k5	778	812	5	852	6
A-n36-k5	799	814	5	884	5
A-n37-k5	669	756	5	734	5
A-n37-k6	949	1 027	7	1 050	7
A-n38-k5	730	819	6	874	6
A-n39-k5	822	864	5	971	6
A-n39-k6	831	881	6	966	6
A-n44-k6	937	1 037	7	1 092	7
A-n45-k6	944	1 040	7	1 043	7
A-n45-k7	1 146	1 288	7	1 281	7
A-n46-k7	914	992	7	1 013	7
A-n48-k7	1 073	1 145	7	1 143	7
A-n53-k7	1 010	1 117	8	1 116	8
A-n54-k7	1 167	1 209	8	1 320	8
A-n55-k9	1 073	1 155	10	1 192	9
A-n60-k9	1 354	1 430	9	1 574	10
A-n61-k9	1 034	1 201	11	1 184	11
A-n62-k8	1 288	1 470	9	1 559	9
A-n63-k9	1 616	1 766	10	1 823	10
A-n63-k10	1 314	1 405	11	1 523	11
A-n64-k9	1 401	1 587	10	1 597	10
A-n65-k9	1 174	1 276	10	1 351	10
A-n69-k9	1 159	1 283	10	1 254	10
A-n80-k10	1 763	1 883	11	2 014	11
Average	1 041.93	1 134.67	7.67	1 181.44	7.78

Table 4. Experimental results of the Augerat benchmark set A

5 EXPERIMENTAL RESULTS

The proposed 3-phase algorithm is tested on Augerat A, B, and P benchmark dataset [19]: The instances in class A, both customer locations and demands are random and the instances in class B are clustered instances and the instances in class P are modified versions of instances from the literature. The file name contains information about the number of nodes in the data and the minimum number of vehicles needed to solve the problem. For example, A-n32-k5 indicates that this problem is of class A, the number of nodes including the depot is 32, and the minimum number of vehicles needed is 5.

Experiments are conducted on the Windows® XP operating system with a Pentium® Dual Processor 3.40 GHz and 896 MB RAM. The first and second phase

Instance	Best known	Centroid-based 3-phase		Sweep + cluster_adjustment	
		Distance	Routes	Distance	Routes
B-n31-k5	672	700	5	713	5
B-n34-k5	788	851	6	845	6
B-n35-k5	955	969	5	1 002	5
B-n38-k6	805	834	6	863	6
B-n39-k5	549	620	5	560	5
B-n41-k6	829	862	7	881	6
B-n43-k6	742	857	6	812	6
B-n44-k7	909	963	7	1 097	8
B-n45-k5	751	807	6	803	6
B-n45-k6	678	743	7	756	7
B-n50-k7	741	772	7	763	7
B-n50-k8	1 312	1 431	9	1 446	8
B-n51-k7	1 032	1 028	8	1 029	8
B-n52-k7	747	754	7	765	7
B-n56-k7	707	741	7	832	7
B-n57-k7	1 153	1 163	8	1 208	8
B-n57-k9	1 598	1 673	9	1 807	9
B-n63-k10	1 496	1 664	10	1 720	11
B-n64-k9	861	910	10	1 023	10
B-n66-k9	1 316	1 468	10	1 483	10
B-n67-k10	1 032	1 108	10	1 134	11
B-n68-k9	1 272	1 338	9	1 362	9
B-n78-k10	1 221	1 276	10	1 479	11
Average	963.74	1 023.13	7.57	1 060.13	7.65

Table 5. Experimental results of the Augerat benchmark set B

algorithms are implemented using Java and the implemented code of Lin-Kernighan, found in the work of Helsgaun [20], is used in the third phase. For comparison, the Sweep heuristic algorithm [8, 9] is implemented since in the same way as the proposed algorithm, the Sweep is categorized as a cluster-first and route-second algorithm and also targets minimizing the travelling distances with an unfixed number of vehicles.

Tables 4 to 6 show the results. The first column of the tables shows the name of benchmark data and the second column shows the best known (shortest) distance until now. The column named 'Centroid-based 3-phase' is for the results of the proposed algorithm. The column labelled 'Sweep + cluser_adjustement' is for the results of the Sweep algorithm with applying the cluster adjustment of this study. Both cases apply the same Lin-Kernighan heuristic algorithm [13] for route establishments. The column named 'Routes' means the number of vehicles used to solve the problem in each algorithm.

As the tables show, on Augerat class A, the proposed algorithm finds shorter distances in the 20 cases among 27 than the Sweep algorithm with the cluster ad-

Instance	Best known	Centroid-based 3-phase		Sweep + cluster_adjustment	
		Distance	Routes	Distance	Routes
P-n16-k8	450	<u>497</u>	9	568	10
P-n19-k2	212	256	3	<u>236</u>	2
P-n20-k2	216	240	2	<u>238</u>	2
P-n21-k2	211	240	2	<u>238</u>	2
P-n22-k2	216	245	2	<u>237</u>	2
P-n22-k8	603	<u>672</u>	10	687	10
P-n23-k8	529	703	12	<u>645</u>	11
P-n40-k5	458	505	5	<u>499</u>	5
P-n45-k5	510	533	5	<u>525</u>	5
P-n50-k7	554	<u>583</u>	7	585	7
P-n50-k8	631	<u>669</u>	9	675	9
P-n50-k10	696	<u>740</u>	11	779	11
P-n51-k10	741	<u>779</u>	11	806	11
P-n55-k7	568	<u>610</u>	7	611	7
P-n55-k8	588	654	8	<u>601</u>	7
P-n55-k10	694	<u>749</u>	10	763	10
P-n55-k15	989	<u>1 022</u>	17	1 056	18
P-n60-k10	744	<u>786</u>	11	823	11
P-n60-k15	968	<u>1 006</u>	16	1 086	16
P-n65-k10	792	<u>836</u>	10	856	11
P-n70-k10	827	<u>891</u>	11	902	11
P-n76-k4	593	685	4	<u>603</u>	4
P-n76-k5	627	737	5	<u>647</u>	5
P-n101-k4	681	<u>698</u>	4	702	4
Average	587.42	639.00	7.96	640.33	7.96

Table 6. Experimental results of the Augerat benchmark set P

justment; on class B, 18 cases among 23; and on class P, 14 cases among 24 cases. One thing should be noticed, namely that in the case of B-n51-k7, the proposed algorithm finds a shorter distance than the best known one by using one more vehicle. Each of these experiments are executed within a hundred of milliseconds.

6 CONCLUSIONS

The vehicle routing problem (VRP) is a nondeteministic polynomial-time (NP) hard problem which cannot be solved in polynomial time and capacitated vehicle routing prolem (CVRP) is a subset of the VRP and also falls into NP hard. In this study, a 3-phase centroid-based heuristic routing algorithm for the CVRP is proposed. With the experiments on well known Auguerat benchmark dataset, the proposed algorithm gives better results in more cases compared to the Sweep heuristic algorithm. The result of the proposed algorithm can be used as an initial solution for meta-heuristic algorithms. For future study, we plan to extend the algorithm to solve other VRP related problems by utilizing the notion of the geometrical centre of a cluster.

Acknowledgement

This research was supported in parts by the MKE (The Ministry of Knowledge and Economy), Korea, under the Chung-Ang University HNRC (Home Network Research Center)-ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0035) and also partially supported by National Research Foundation of Korea Grant funded by the Korean Government (2009-0076290).

REFERENCES

[1] Lysgaard, J.—Letchford, A. N.—Eglese, R. W.: A New Branch-And-Cut Algorithm for the Capacitated Vehicle Routing Problem. Math Program, Vol. 100, 2004, pp. 423–445.

[2] Fukasawa, R.—Lysgaard, J.—de Aragão, M. P.—Reis, M.—Uchoa, E.—Werneck, R. F.: Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. Math Program, Vol. 106, 2006, pp. 491–511.

[3] Clarke, G.—Wright, J.: Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Opns. Res., Vol. 12, 1964, pp. 568–581.

[4] Yellow, P.: A Computational Modification to the Savings Method of Vehicle Scheduling. Op. Res. Quart., Vol. 21, 1970, pp. 281–283.

[5] Paessens, H.: The Savings Algorithm for the Vehicle Routing Problem. Eur. J. Opl. Res., Vol. 34, 1988, pp. 336–344.

[6] Mole, R. H.—Jameson, S. R.: A Sequential Route-Building Algorithm Employing a Generalized Saving Criterion. Op. Res. Quart., Vol. 27, 1976, pp. 503–511.

[7] Christofides, N.—Mingozzi, A.—Toth, P.: The Vehicle Routing Problem. In: N. Christofides, A. Mingozzi, P. Toth and C. Sandi (Eds.): Combinatorial Optimization. John Wiley & Sons, New York, NY 1979.

[8] Gillet, B. E.—Miller, L. R.: A Heuristic Algorithm for the Vehicle Dispatch Problem. Opns. Res., Vol. 22, 1974, pp. 340–349.

[9] Wren, A.—Holliday, A.: Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. Op. Res. Quart., Vol. 23, 1972, pp. 333–344.

[10] Fisher, M. L.—Jaikumar, R.: A Generalized Assignment Heuristic for Vehicle Routing. Networks, Vol. 11, 1981, pp. 109–214.

[11] Bramel, J. B.—Simchi-Levi, D.: A Location Based Heuristic for General Routing Problems. Opns. Res., Vol. 43, 1995, pp. 649–660.

[12] Lin, S.: Computer Solutions of the Traveling Salesman Problem. Bell. Syst. Tech. J., Vol. 44, 1965, pp. 2245–2269.

[13] Lin, S.—Kernighan, B.: An Effective Heuristic Algorithm for the Traveling Salesman Problem. Opns. Res., Vol. 21, 1973, pp. 498–516.

[14] Or, I.: Traveling Salesman-Type Combinatorial Optimization Problems and Their Relation to the Logistics of Regional Blood Banking. Ph. D. thesis, Northwestern University, 1976.

[15] Bullnheimer, B.—Hartl, R. F.—Strauss, C.: Applying the Ant System to the Vehicle Routing Problem. In: S. Voβ, S. Martello, I. H. Osman and C. Roucairol (Eds).: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer, Boston, MA, 1998.

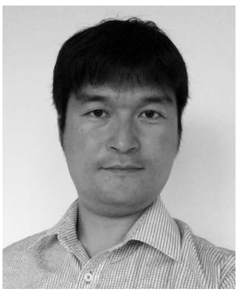
[16] Bullnheimer, B.—Hartl, R. F.—Strauss, C.: An Improved Ant System Algorithm for the Vehicle Routing Problem. Ann. Op. Res., Vol. 89, 1999, pp. 319–328.

[17] Baker, B. M.—Ayechew, M. A.: A Genetic Algorithm for the Vehicle Routing Problem. Comput. Opns. Res., Vol. 30, 2003, pp. 787–800.

[18] Glover, F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Comput. Op. Res., Vol. 13, 1986, pp. 533–549.

[19] Augerat, P.: Approche Polyèdrale du Problème de Tournées de Véhicules. Ph. D. thesis, Institut National Polytechnique de Grenoble-Inpg, 1995. Available on: `http://www.branchandcut.org/VRP/data/`.

[20] Helsgaun, K.: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. Eur. J. Opl. Res., Vol. 126, 2000, pp. 106–130.



Kwangcheol Shin received his B. Sc., M. Sc. and Ph. D. in Computer Science and Engineering from Chung-Ang University, Seoul, Korea, in 1996, 1998, and 2004, respectively. He worked as a visiting postdoctoral researcher in the Department of Computer Science at the University of Minnesota in 2005. He is currently a research professor in the Department of Computer Science at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His research interests include ubiquitous sensor networks, optimization algorithm and embedded systems.

Sangyong Han is a Professor of the School of Computer Science and Engineering, ChungAng University, Seoul, Korea. He received Bachelor of Engineering at College of Engineering of Seoul National University in 1975, and the Ph. D. degree in 1984 from University of Minnesota. From 1984 to 1995, he worked at Poughkeepsie Lab. and Watson Research Center in IBM, USA. His research interests include web technologies (web services, semantic web), social networks, and information retrieval.