Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 19(2): 114-119, Jun. 2021 Regular paper

Maximum Terminal Interconnection by a Given Length using Rectilinear Edge

Minkwon Kim, Yeonsoo Kim, Hanna Kim, and Byungyeon Hwang*, *Member, KIICE*

School of Computer Science and Information Engineering, the Catholic University of Korea, Bucheon, 14662 Korea

Abstract

This paper proposes a method to find an optimal T' with the most terminal of the subset of T' trees that can be connected by a given length by improving a memetic genetic algorithm within several constraints, when the set of terminal T is given to the Euclidean plane R^2. Constraint (1) is that a given length cannot connect all terminals of T, and (2) considers only the rectilinear layout of the edge connecting each terminal. The construction of interconnections has been used in various design-related areas, from network to architecture. Among these areas, there are cases where only the rectilinear layout is considered, such as wiring paths in the computer network and VLSI design, network design, and circuit connection length estimation in standard cell deployment. Therefore, the heuristics proposed in this paper are expected to provide various cost savings in the rectilinear layout.

Index Terms: Genetic algorithm, Rectilinear steiner tree, Interconnection graph problem, NP-hardness

I. INTRODUCTION

The construction of a maximum interconnection of elements distributed within a given space is a problem abstracted in various industrial fields from network to architecture [1]. In this interconnection construction, connecting all terminals using a minimum length can be obtained using the minimum cost spanning tree (MST) algorithm [2]. It has been proven that the problem of constructing maximum interconnections is NP-hard when the given length is unable to connect all elements and heuristics, which build efficient interconnection through the memetic genetic algorithm has been proposed [3].

This paper aims to develop a construction method for interconnections using only the rectilinear layout edge through the rectilinear memetic genetic algorithm after adding a constraint of using only the rectilinear layout in the random distribution. In many real-world cases only the rectilinear layout edge is used, including the path for overall wiring of the computer network and VLSI design, network design, and circuit connection length estimation for standard cell placement [4-6]. Therefore, by applying the improvements proposed in this paper, it can be used more efficiently in real life.

II. NP-HARDNESS OF A SUBGRAPH

Definition 2-1. A Steiner tree *is composed of a given terminal as an instance. However, to save length, an arbitrary terminal can be added to the given terminal, which is called the Steiner point. Among Steiner trees, one using the minimum length is called the Steiner minimum tree (SMT) [7].*

Terminal set T is assigned to the Euclidean plane R^2. In

Received 01 April 2021, **Revised** 25 May 2021, **Accepted** 02 June 2021
***Corresponding Author** Byungyeon Hwang (E-mail: byhwang@catholic.ac.kr, Tel: +82-2-2164-4363)
School of Computer Science and Information Engineering, the Catholic University of Korea, Bucheon, 14662 Korea
Minkwon Kim, Yeonsoo Kim equally contributed to this work.

Open Access https://doi.org/10.6109/jicce.2021.19.2.114 print ISSN: 2234-8255 online ISSN: 2234-8883

addition, suppose that the candidate of all possible Steiner point distributions of each plane exist as set S:

Definition 2-2. A rectilinear minimum spanning tree (RMST) *is an MST constructed with terminal set T as an instance. However, all interconnections are MSTs constructed using the Manhattan distance for a rectilinear layout [8].*

Definition 2-3. A rectilinear Steiner tree (RST) *is a Steiner tree that uses only vertical and horizontal edges, considering only the rectilinear layout*

Lemma 2-3-1. *RST means RMST with $T \cup S'$ as an instance when Steiner point subset S' is present, not an empty set.*

Definition 2-4. The minimum rectilinear Steiner tree (MRST) *is an optimal RST with the minimum length among all RSTs.*

Lemma 2-4-1. *When the Steiner point subset of MRST is S_{opt}, MRST means RMST with $T \cup S_{opt}$ as an instance.*

Lemma 2-4-2. *MRST is a subgraph of a Hanan grid with a given terminal set as an instance [9].*

Lemma 2-4-3. *The Steiner point subset S_{opt} of MRST exists within the convex hull with a given terminal set as an instance [10].*

Formulate the following problems to prove the proposed problem.

Problem SMTP (Steiner minimum tree problem) INSTANCE: A finite set of given terminals on a Euclidean plane. QUESTION: Find the SMT for the INSTANCE. Problem A (Consider SMT in the problem of the previous paper [3]) INSTANCE 1: Given length L. INSTANCE 2: A finite set of given terminals on a Euclidean plane. QUESTION: Find the SMT that has the max number of terminals on a Euclidean plane within length L.

For any given instance (say α) of the SMTP, transform it into the instance of problem A as follows: First, calculate the length (say L') of MST for α. INSTANCE 1 for problem A is L'. The INSTANCE 2 of Problem A is α. Note that for the same set of terminals on a Euclidean plane, it is proven that the length of MST is ≥ the length of SMT [7]. Note that α has an optimal solution, say T_{opt}, although we can hardly find it practically (say the length of T_{opt} is L_{opt}).

If the solver for problem A exists, we may solve the SMTP using it. Starting from L', by using the binary search method over length L', we may reach L_{opt}, the minimum length, where all terminals are interconnected. Note that in each step of this binary search, check if all the terminals are interconnected, and then lengthen or shorten the length so that we can reach L_{opt} efficiently. This algorithm is as follows:

Step 1) The calculation of L' for MST takes p-time.

Step 2) We know that the binary search process may repeat log(L') times at the most, which makes the time of the binary search log(L'). As a result, the time required to reach L_{opt} from L' takes the p-time * log (L'). SMTP has already been proven to be NP-hard [7]. Therefore, Problem A is NP-hardness. The proposed problem of finding a subgraph is formulated as follows:

Problem B (Consider the MRST in the problem of the previous paper [3]) INSTANCE 1: Given length L. INSTANCE 2: A finite set of given terminals on a Euclidean plane. QUESTION: Find the MRST that has the max number of terminals on a Euclidean plane within the length L.

III. HEURISTIC OF THIS PROBLEM

In this paper, we express the state of the individual as an RST for the correct linear layout of the edge. Therefore, not only the included terminal status but also the Steiner point status should be an individual field. In addition, the Steiner point candidate has a large space compared with the number of given terminals. Only the candidate of the included Steiner point is stored, and the coordinates of each Steiner point must be included as a collection that can react dynamically.

Because the problem in this paper is finding the minimum length connecting all the given terminal set T to obtain MRST, it is NP-hard. Therefore, we set the given length using the following definition: "The ratio of RMST (T) to MRST (T) is lower than 3/2 [11]." Equation 1 is a modified formula for this proposition. At this point, the RMST (T) of (1) can be obtained using the MST algorithm. Thus, the given length was set to be less than 2/3 RMST (T).

$$\frac{2}{3} RMST(T) \le MRS(T) \left(for \ \frac{RMST(T)}{MRST(T)} < \frac{3}{2} \right). \quad (1)$$

Let us assume that the Euclidean plane R^2 given terminal set T is an r*r plane. At this time, there are possible r^2 Steiner point candidates. Therefore, it is necessary to reduce the search space by reducing the number of Steiner point candidates. First, when a certain terminal subset T' exists, the optimal Steiner point subset S' that optimizes RMST (T') must be the subset of Hanan grid H (T) [9]. Therefore, when the number of terminals of the given terminal set T is n, the Steiner point candidate can be reduced to n^2 through H (T). Second, when a certain terminal subset T' exists, the optimal Steiner point subset S' that optimizes RMST (T') always exists within the convex hull consisting of T' [10]. Therefore, the Steiner point candidate can be selected from the Steiner points inside the convex hull consisting of a given terminal set T. This method is used to store the candidate Steiner point of a given terminal set. These data will be used later in the Steiner point-related operation.

Compared with previous studies, this paper cannot evaluate the usability of individual Steiner point subsets using only length. Accordingly, a criterion for determining the usability of the Steiner point subset of a specific individual is required. Therefore, the usability of the Steiner point is defined according to the degree. Fig. 1 shows the layout of the Steiner point according to its degree.

For degree 1, the Steiner point is the end point of the connection. Hence, it wastes length. For degree 2, there is no change in length when considering the Manhattan distance. For degree 3 or more, the Steiner point saves length by overlapping the edge of the RST. Therefore, the Steiner point degree can be a criterion for evaluating the usability of the Steiner point. This information is used for the fitness.

In the case of the usability of the Steiner point, the interconnection of individuals is unknown until it is obtained through the RST. Therefore, it cannot be used when determining the degree of superiority in the RST operation process. Therefore, it is necessary to find a Steiner point with high usability without obtaining an RST for any individual. This paper approaches this through the Steiner point pool, which is a methodology for selecting Steiner point candidates that are more likely to be useful on a terminal subset including an individual, considering the characteristics that the Steiner point is useful when the degree is 3 or more, and the Prim algorithm, which is the logic used to obtain RST, connects from the nearest terminal. The basis of this methodology is that when there are three terminals, the coordinates of the optimal Steiner point have the center values of the X and Y coordinates of each terminal. Suppose that the sequence of an individual's terminal connection is known. In this case, interconnection through the Prim algorithm has the characteristic of selecting the nearest terminal among the unselected terminals. Therefore, all the optimal Steiner points of all 3-terminal subsets that maintain their sequence order are obtained and stored. This information becomes the Steiner point pool of an individual.

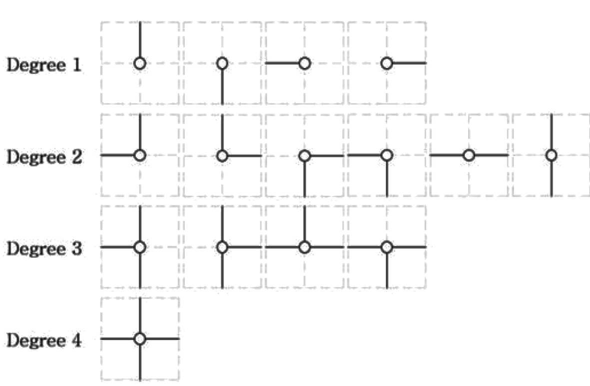

Fig. 1. Layout of Steiner point by degree.

The terminal status, as in the previous paper, is generated via IGP (Initial Generation Probability) [3]. In this case, the Steiner point status has a random number of Steiner points less than the number of terminals that the individual has. This Steiner point is randomly selected from the Steiner point candidates selected through the Hanan grid and the convex hull of the given terminal set.

The fitness [3] of the previous paper was modified to reflect the usability of the Steiner point of this paper. Eq. (2) is a modified fitness function.

$$\text{Fuction of Fitness}(\not{f}) : \not{f} = 1 - \not{f}' \quad (2)$$

$$\mathcal{A} = \alpha * \min\left(1.0, \frac{\textit{Length of Individual}}{0.95 * \textit{given length}}\right)$$

$$\mathcal{B} = \beta * \left(1 - \frac{\textit{\# of included terminal of individual}}{\textit{\# of given terminal}}\right)$$

$$\Gamma = \gamma * ((3 * (\textit{\# of given teminal} - 2)) - \textit{Sum of Steiner Point Degree of Individual})$$

$$(\alpha + \beta + \gamma = 1),\ \not{f}' = \mathcal{A} + \mathcal{B} + \Gamma$$

This fitness is evaluated better when lesser length is used, more number of terminals are included, and when the sum of the degree values of the included Steiner point is greater.

A one-point crossover is used for each parent individual [3, 6]. In this paper, it is necessary to consider the crossover of the Steiner point status in addition to the crossover between the terminal statuses. A division point is set by generating a random number according to the smaller size of the Steiner point status size among the two individuals. At this time, as the generated child individual has two child terminal status and child Steiner point status, respectively, four child individuals are created.

The terminal status was performed in the same manner as in a previous paper [3]. For the Steiner point status, a mutation occurs with a 5% probability. When a mutation occurs, a random number of [0 ... size of Steiner point status] is generated, and two Steiner points from the index are mutated to a random Steiner point within the candidate Steiner point.

There is no guarantee that the generated child individual and the individual created in the initialization of the population do not exceed a given length. Therefore, the length must be adjusted so that it does not exceed the specified length through adjustment. Suppose that a certain child individual has a terminal subset T' and a Steiner point subset S'. Adjustment establishes RST as T'∪S' instance and checks whether it exceed the given length in this process. If the given length is exceeded, the terminal and Steiner points that are excluded are deleted from the status. In this case, the degree value of the Steiner points is maintained for a local search.

A local search is performed for the local optimization of the child individual after adjustment. This operation consists of three steps. Step 1) Delete or modify the Steiner point in the Steiner point status. After checking the degree of each Steiner point, the Steiner point with degree 1 is deleted with a certain probability (say P_{delete}). Steiner points with a degree of 2 are deleted with the same probability and then replaced by a new Steiner point with a certain probability (say $P_{firstAdd}$). After selecting two terminals closest to an arbitrary terminal within the terminal status and making a 3-terminal set, a new Steiner point is selected based on the principle of the Steiner point pool. After the completion of the first step, the length is recalculated and stored through the RST. When all terminals and Steiner points are connected, move on to the next step. Step 2) Repeat the process of containing a terminal closest to the current component among the terminals that are excluded. However, this process calculates the length by rebuilding the interconnection from the beginning when each terminal is added. This is because the component length with a new terminal can be smaller than the sum of the lengths from the existing component to the new terminal.

Theorem 3-1. *The length of the component with a new terminal is less than or equal to the length of the existing component to the new terminals added.*

Proof. *Let us assume that the interconnected terminal set* $T = \{T_1, T_2, T_3 \ldots T_k\}$ *until now exists. Using length T is the length (T). Next, we added* $T_{k+1.}$ *Suppose that the minimum length from the component T to* T_{k+1} *is L. When a set in which* T_{k+1} *is added to T is called* $T' = \{T_1, T_2, T_3 \ldots T_k, T_{k+1}\}$, *and the length of this T' is the length (T'), length (T')* $\leq$ *length (T) + L. This is because when* T_{k+1} *is added, the edge toward the direction from the interconnection within set T to* $(T_1 \rightarrow T_{k+1})$, $(T_2 \rightarrow T_{k+1})$ ... $(T_k \rightarrow T_{k+1})$ *can use a smaller length than that of the existing interconnections. Therefore, the length (T') has a value less than or equal to length (T) + L.*

This operation is repeated until the re-established interconnection exceeds the given length after adding a new terminal. In addition, when a terminal is added, the terminal is stored with a certain probability (say $P_{secondAdd}$). Step 3) Based on the principle of the Steiner point pool, a new Steiner point is added to the stored terminal in step 2. Afterward, the interconnection is reconstructed, and the individual length is stored.

This paper aims to find the optimal terminal subset and optimal Steiner point subset within a given length. However, because the search space is substantial to find two subsets at once, the algorithm proposed in this paper is divided into two parts. Part 1 focuses on finding an excellent terminal subset, whereas Part 2 focuses on finding an excellent Steiner point subset in that terminal subset. In Part 1, the ratio of γ in fitness and the rate of change in Steiner points according to the degree in local search is low. Therefore, the operation is based on an optimal terminal subset. At this time, if the fitness of an individual in the population becomes more than a certain level, the algorithm switches to Part 2, wherein the ratio of γ in fitness and the rate of change in Steiner points according to the degree of local search are high. Therefore, the operation is based on the optimal Steiner point subset that fits the optimal terminal subsets selected in Part 1.

When the number of terminals in a given terminal set is 1000, the algorithm terminates if more than 30,000 generations proceed, or if the fitness of an individual in the population is more than 90%.

IV. EXPERIMENTS

A. Environment of the Experiment

The implementation environment for the experiment is the same as in [3], and the heuristic proposed in this paper is written in Java using Eclipse. Because each instance is created in a different form, the ratio of the number of terminals included in the minimum cost rectilinear tree close to a given length (MCRTL) and the rectilinear memetic genetic algorithm (RMGA) from this paper is constant, but the absolute number is inconsistent. Therefore, a comparable figure for each algorithm is required. In this paper, we compare this with the percentage of the number of terminals contained by each algorithm based on the number of terminals contained by MCRTL. This figure is defined as quality of the best individual (QBI), which is calculated using Eq. (3) as follows:

$$QBI = \frac{\#\ of\ terminals\ included\ in\ Best\ Individual}{\#\ of\ terminals\ included\ in\ the\ MCRTL} * 100. \quad (3)$$

For instance, self-made and OR-Library are used and the setting of self-made is as follows: (1) the number of terminals is 1,000 and they are given to the 1,000 × 1,000 Euclidean plane; (2) the coordinates of each terminal are given as integer values; (3) furthermore, the values of each parameter are fixed throughout the experiment as follows.

Size of population = 150;
Given length = 50% of maximum length;
In Part 1: α = 0.16, β = 0.64, γ = 0.2,
P_{delete}=0.1, $P_{firstadd}$=0.5, $P_{secondAdd}$=0.5;

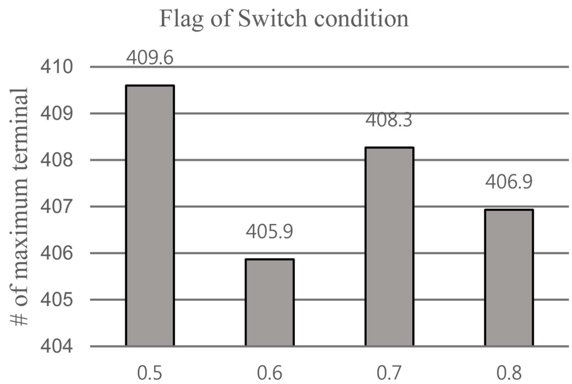

Fig. 2. Flag of Switch condition.

In Part 2: $\alpha = 0.1$, $\beta = 0.4$, $\gamma = 0.5$,
P_{delete}=0.2, $P_{firstadd}$=0.9, $P_{secondAdd}$=0.8.

B. Flag of Switch Condition

The switch condition should be applied after the optimal terminal subset is explored to a certain extent. Therefore, the switch condition was specified by conducting an experiment when the fitness of the population was 50%, 60%, 70%, and 80% identical. Fig. 2 shows a numerical graph of the maximum terminal value when the switch condition is n% for the same instance. This figure is the average value of five self-made instances after conducting 10 experiments each.

As shown in Fig. 2, the switch condition of 50% shows the best result. In other words, when the population is approximately 50% the same, candidates for the optimal terminal subset are sufficiently selected, and finding the optimal Steiner point subset afterward produces the best results.

C. Performance Comparison of Algorithms using Self-made Instance

In this experiment, it is shown that the RMGA proposed in

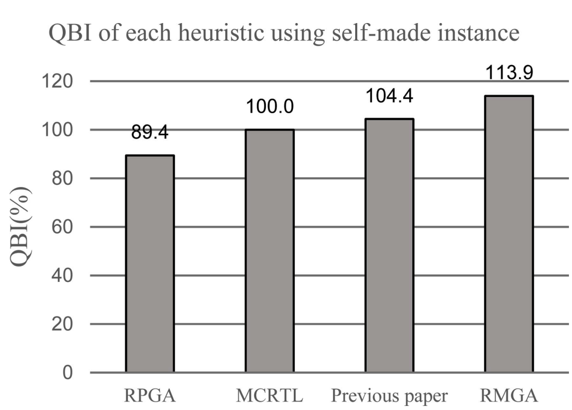

Fig. 3. QBI of each heuristic using a self-made instance.

Table 1. Standard deviation of Fig. 3 (based on QBI)

	RPGA	Previous paper	RMGA
σ	3.373502	1.4767	1.398957

Table 2. Runtime of Fig. 3

	RPGA	Previous paper	RMGA
Time (sec)	64.83027	1077.368018	1764.619

this paper is more efficient than the benchmark algorithm and a previous paper [3]. Furthermore, through the QBI figures of the RPGA, which excluded local search from RMGA, it can be confirmed that the local search proposed in this paper works efficiently. The resulting QBI value is an average value obtained by performing an average of 10 experiments for each instance and proceeding with 30 different instances in the same manner. Fig. 3 shows a graph comparing the QBI values of each heuristic.

As shown in Fig. 3, RMGA showed a 13.9% better QBI compared with MCRTL. It also showed a 10% better QBI value compared with the previous paper's algorithm. Considering that MCRTL connected around 361 terminals for 30 instances, RMGA connected approximately 50 more vertices than did MCRTL. In addition, RPGA has a QBI of 89.4%, which is lower than that of the MCRTL. This result proves that the local search proposed in this paper works efficiently. Table 1 shows the standard deviation of this experiment, and Table 2 lists the average value of each heuristic runtime.

Considering the standard deviation in Table 2, RMGA shows a smaller standard deviation compared with RPGA, which in turn proves that RMGA is more stable compared with RPGA.

V. CONCLUSIONS

This paper has suggested how to find the $T_{optimal}$ with the maximum number of terminals using the RMGA among the MRSTs of the subset T' of T that can establish interconnection only with the rectilinear layout edge when a given terminal set T is given on a Euclidean plane. In addition, it was shown that the proposed algorithm yielded better results compared with existing algorithms. For the actual MRST problem, which indicates the applicability of the algorithm in real life, positive results were derived as well. In addition to the constraints on the rectilinear layout edge, the actual construction of the interconnection has varying specificity, such as an interconnection considering the obstacles in the connection environment, and an interconnection with a limited number of connections to each terminal. We leave these problems considering this specificity as a task for future research.

ACKNOWLEDGEMENTS

This work was supported by the Catholic University of Korea, Research Fund (2019).

REFERENCES

[1] K. Zhou and J. Chen, "Simulation DNA algorithm of set covering problem," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 139-144, Jan. 2014. DOI: 10.12785/amis/080117.
[2] D. Hu, P. Dai, K. Zhou, and S. Ge, "Improved particle swarm optimization for minimum spanning tree of length constraint problem," in *Proceeding of International Conference on Intelligent Computation Technology and Automation*, Nanchang, pp. 474-477, 2015. DOI: 10.1109/icicta.2015.124.
[3] J. Kim, J. Oh, M. Kim, Y. Kim, J. Lee, S. Han, and B. Hwang, "Maximum node interconnection by a given sum of Euclidean edge lengths," *Journal of information and communication convergence engineering*, vol. 17, no. 4, pp. 246-254, 2019. DOI: 10.6109/jicce.2019.17.4.246.
[4] C. N. Chu and Y. C. Wong, "FLUTE: Fast lookup table based rectilinear Steiner minimal tree algorithm for VLSI design," *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, vol. 27, no. 1, pp. 70-83, Jan. 2008. DOI: 10.1109/TCAD.2007.907068.
[5] X. Cheng and D. Du, *Steiner trees in industry*, Kluwer Academic Publishers, 2013.
[6] G. Wei and X. Xie, "Research of using genetic algorithm of improvement to compute the most short path," in *Proceeding of International Conference on Anti-counterfeiting, Security, and Identification in Communication*, Hong Kong, pp. 516-519, 2009. DOI: 10.1109/ICASID.2009.5276990.
[7] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees." *SIAM J. of Appl. Math.*, vol. 16. no. 1, pp. 1-29, 1968. DOI: 10.1137/0116001.
[8] M. Garey and D. S. Johnson, "The rectilinear Steiner tree Problem is NP complete," *SIAM Journal of Applied Mathematics*, vol. 32, no. 4, pp. 826-834, 1977. DOI: 10.1137/0132071.
[9] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM Journal of Applied Mathematics*, vol. 14, no. 2, pp. 255-265, 1966. DOI: 10.1137/0114025.
[10] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letter*, vol. 1, no. 4, pp. 132-133, 1972. DOI: 10.1016/0020-0190(72)90045-2.
[11] F. K. Hwang, "On Steiner minimal trees with rectilinear distance," *SIAM Journal of Applied Mathematics*, vol. 30, no. 1, pp. 104-114, 1976. DOI: 10.1137/0130013.

Minkwon Kim
received his B.S. degree in Computer Science and Information Engineering from the Catholic University of Korea in 2021. His research interests include database and approximation algorithm.

Yeonsoo Kim
received his B.S. degree in Computer Science and Information Engineering from the Catholic University of Korea in 2021. His research interests include database and approximation algorithm.

Hanna Kim
is an undergraduate majoring in Computer Science and Information Engineering at the Catholic University of Korea since 2017. Her research interests include database and approximation algorithm.

Byungyeon Hwang
received his B.S. degree in Computer Engineering from Seoul National University, Korea in 1986, MS and PhD degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1989 and 1994, respectively. He is a professor in the School of Computer Science and Information Engineering at the Catholic University of Korea. His research interests include database, social network analysis, and approximation algorithms.