



# Maximum Node Interconnection by a Given Sum of Euclidean Edge Lengths in a Cluster Node Distribution

Yeonsoo Kim , Minkwon 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 a tree with the maximum number of terminals that can be connected by a given length when numerous terminals distributed in a cluster form are given to the Euclidean plane  $R^2$  with several constraints. First constraint is that a given terminal is distributed in a cluster form, second is that a given length cannot connect all terminals in the tree, and third is that there is no curved connection between each terminal. This paper proposes a method to establish more efficient interconnections within terminals distributed in a cluster form by improving a randomly distributed memetic genetic algorithm. The construction of interconnections has been extensively used in design-related fields, from networking to architecture. Additionally, in real life, the construction of interconnections is mostly distributed in the form of clusters. Therefore, the heuristic algorithm proposed in this paper can be effectively utilized in real life and is expected to provide various cost savings.

**Index Terms:** Cluster distribution, Genetic algorithm, Interconnection graph problem, NP-hardness

## I. INTRODUCTION

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

This paper aims to create a construction method for efficient interconnection when terminals are distributed in a cluster form through the cluster memetic genetic algorithm, which is an improved memetic genetic algorithm with a random distribution. Most abstracted maximum interconnection

constructions in real life form a distribution of clusters. Therefore, the cluster memetic genetic algorithm can be used more efficiently in real life, to generate various cost-reduction effects [4-7].

## II. RELATED WORK

The genetic algorithm proposed in [3] considered establishing maximum interconnections within a given length such that not all terminals can be connected when the given terminals are in the Euclidean plane. Additionally, it proved that the problem is NP-hard, and proposed a memetic genetic algorithm for solving it [8-11].

But the terminal in the problem considered in [3] had a random distribution and did not apply the properties of a specific type of distribution. Most distributions abstracted in


Received 06 July 2021, Revised 13 May 2022, Accepted 20 May 2022

\*Corresponding Author Byungyeon Hwang (E-mail: [byhwang@catholic.ac.kr](mailto:byhwang@catholic.ac.kr), Tel: +82-2-2164-4363)

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

Open Access <https://doi.org/10.6109/jicce.2022.20.2.90>

print ISSN: 2234-8255 online ISSN: 2234-8883

 This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

real life are in a cluster form. This paper proposes a cluster memetic genetic algorithm that applies the properties of cluster distribution by improving the research technique in [3].

### III. GENETIC ALGORITHM REFLECTING CLUSTER DISTRIBUTION

#### A. Pre-processing and Concepts

##### 1) Clustering

In this paper, we propose each operation such that the corresponding properties can be applied in cases wherein the terminals are distributed in a cluster. Therefore, it is necessary to first distribute these clustered terminals and store the corresponding information. The K-means algorithm was used to cluster the given terminals [12, 13].

##### 2) Representing Individuals

To proceed with the genetic algorithm, the method of expressing an individual must be first considered. In this paper, after clustering a given terminal, the order of the terminals was rearranged by sorting the index in the order of the stored cluster numbers. Fig. 1 shows an example of the terminal with the indexes of the individuals placed in a given order (Case 1, uniform memetic genetic algorithm individual expression) and the indexes of the individuals rearranged in the order of the cluster numbers (Case 2, cluster memetic genetic algorithm individual expression).

Considering the distribution of clustered terminals and one-point crossover, the good properties of the crossover individual are scattered when the terminal arranges the indexes of the individuals in a given order. Therefore, the individuals are sorted in order of their cluster numbers, and the indexes are rearranged such that the indexes of the termi-

nal are grouped in a cluster. This allows the child individual to maintain the good properties of the parent.

##### 3) Cluster Density

Owing to the characteristics of the cluster distribution, connecting terminals concentrated in one cluster is better than connecting those scattered across several clusters. Additionally, the more terminals that are distributed in a narrow range, the more terminals that can be connected with less length. Therefore, the cluster density was defined as an indicator to determine which clusters are advantageous for establishing interconnections. To obtain the cluster density, an index is required to determine the area of the cluster. In the cluster, the upper-left coordinate value is defined by the x value of the leftmost terminal and the y value of the uppermost terminal, and the lower-right coordinate value is defined by the x value of the rightmost terminal and the y value of the lowermost terminal. A rectangle is defined using these two coordinate values, and the area of the rectangle is considered as the area of the cluster. Fig. 2 shows an example of the calculation of the area of a cluster.

Subsequently, the cluster density is defined by assuming that the size of the terminal in the cluster is 1. Equation (1) is used to obtain the cluster density. However, its size is too small to be applied in subsequent operations. Therefore, after obtaining each cluster density, it is normalized to a value between [0, 1].

$$Density\ of\ cluster\ i = \frac{\#\ of\ terminals\ in\ cluster\ i * size}{Area\ of\ cluster\ i}$$

$$size = 1.0 \tag{1}$$

#### B. Genetic Algorithm

##### 1) Mutation

In a random distribution, different types of individuals can be created, regardless of the starting terminal. However, in a cluster distribution, regardless of which terminal in a cluster starts a connection, similar interconnections are established and then transferred to another cluster. This allows individu-

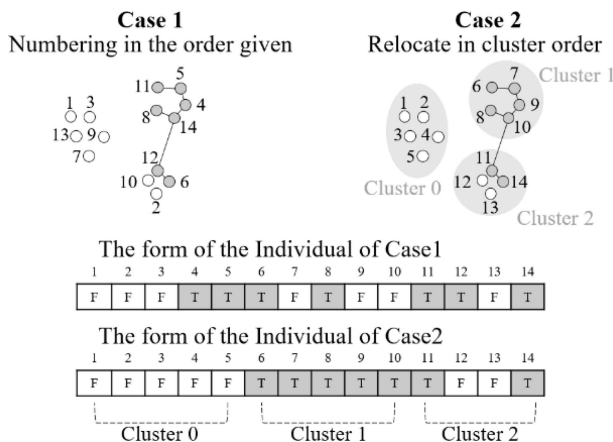


Fig. 1. Example of relocate Individual.

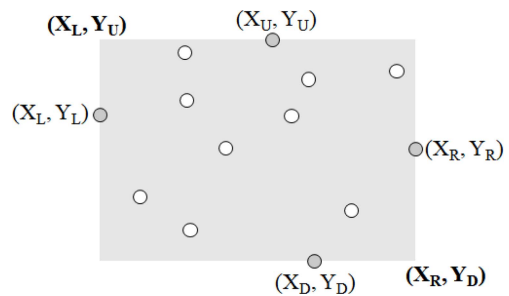


Fig. 2. Area of cluster.

als within a population to converge rapidly and hinders sufficient exploration. Therefore, it is necessary to reduce the convergence speed by setting the mutation probability to be high. The mutation probability of the cluster memetic genetic algorithm proposed in this paper was 0.1%.

**2) Local Search**

The local search of the uniform memetic genetic algorithm is executed as follows: for the child individual whose adjustment has been completed, the current child individual is not included, and the terminal closest to the current child individual's interconnection information is newly included. Thereafter, the length currently used by the individual is calculated through MST, and if it is less than the given length, the process is repeated and the terminal is included. But if the used length is greater than the given length, the most recently included terminal is deleted and the individual is returned.

Local search is similar to Prim's algorithm in that it repeats the operation, including the nearest terminal in the interconnection. However, it even allows for optimal cases that cannot be considered in Prim's algorithm. This is because the problem in this paper is that the given length cannot be connected to all terminals, and that the length is recalculated whenever a new terminal is included.

Assuming that the graph of the interconnection of the individual being computed is  $G$  and the length used by  $G$  is  $W_G$ , the terminal closest to  $G$  that is not included is  $V_i$ , and the length to  $V_i$  is  $W_i$ . Subsequently, if the interconnection graph of the individual including  $V_i$  is  $G'$ , Prim's algorithm measures the length used by  $G'$  as  $W_G + W_i$ . In this case, the length used by  $G'$  cannot be guaranteed to be  $W_G + W_i$  because  $V_i$  is added, the edges from  $V_i$  to  $G$  terminals should also be considered, and the existing  $G$  interconnection information may change. Thus, the length used by  $G'$  can be smaller than  $W_G + W_i$ . Local search accounts for these points and recalculates the used length whenever a new terminal is included; thus, a case that cannot be considered only by the previous operation can be applied. Fig. 3 shows an example of a local search.

Local search is the most important factor in determining the performance of a memetic genetic algorithm. The local search of a uniform memetic genetic algorithm only considers the distance from the current individual's interconnection information to other terminals. However, owing to the characteristics of cluster distribution, it cannot be guaranteed that the terminal closest to the current individual interconnection information is always the best choice. Fig. 4 illustrates the case wherein a search is impossible with the local search of a uniform memetic genetic algorithm.

A random distribution can neglect this case, as shown in Fig. 4. However, cluster distribution cannot be ignored. Although this is not the minimum distance, it is closely

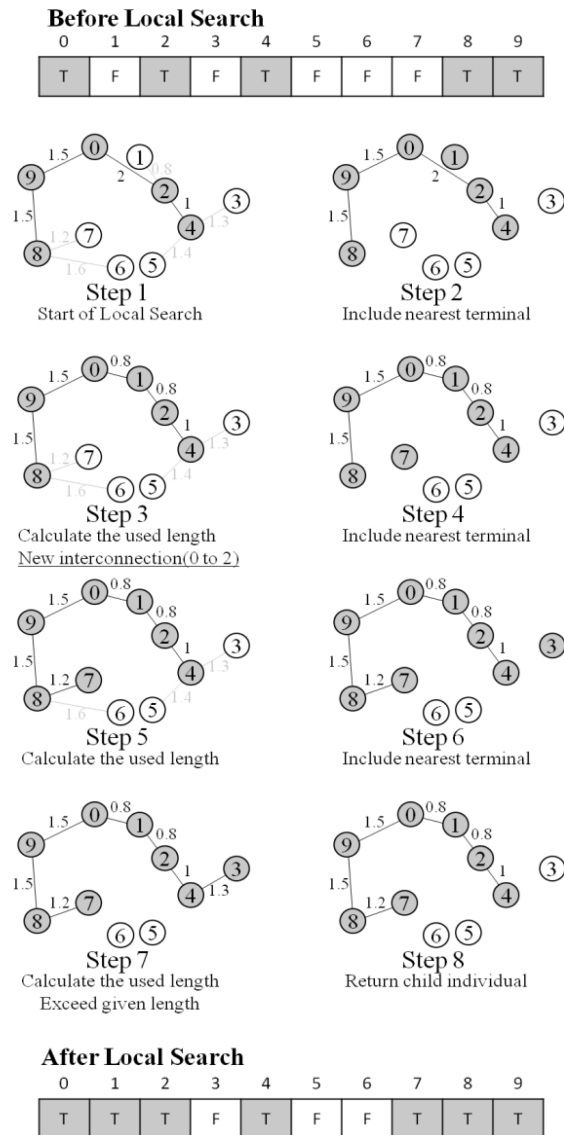


Fig. 3. Example of Local Search.

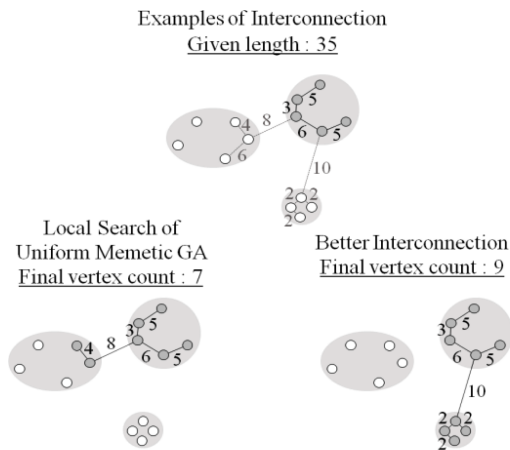


Fig. 4. A case better than previous Local Search.

related to the cluster density because terminals are clustered, and there exists an advantageous terminal for subsequent connections. Therefore, we propose considering the case shown in Fig. 4 in the local search.

Such cases can be considered by reflecting the cluster density along with the distance in the process of including a new terminal in the local search. Equation (2) can be used for calculating the level of distance and density (LDD) values that include the distance in the local search process from the current interconnection information for a particular terminal and the density of the cluster to which the terminal belongs.

$$LDD[k] = A + B \quad (2)$$

$$A = \alpha * \min\left(1.0, \left(\frac{1}{dist[k]}\right)\right)$$

$$B = \beta * \text{Cluster of terminals } k \text{ density } (\alpha + \beta = 1.0)$$

$$dist[k] = \text{distance from interconnection of individual to } v_k$$

The distance is reflected in the part where  $\alpha$  is multiplied and the cluster density is reflected in the part where  $\beta$  is multiplied. The LDD value is between  $[0, 1]$ , and a larger LDD indicates a more advantageous terminal, thereby connecting the terminal. The values of  $\alpha$  and  $\beta$  are 0.9 and 0.1, respectively, which are intuitive values obtained through experiments. By reflecting the LDD[k] value, it is possible to secure a search space suitable for the distribution of clustered terminals.

### 3) Termination Condition

The algorithm is terminated if more than 5,000 generations occur or if the fitness of an individual in the population reaches more than 90%.

## IV. EXPERIMENTS

### A. Benchmark Model

A genetic algorithm is heuristic with randomness and does not return consistent results. Therefore, a heuristic algorithm that returns a constant result for performance evaluation must be defined. The problem of connecting the maximum number of terminals with a small length can be associated with the MST. Therefore, the benchmark model was defined by modifying Prim's algorithm. First, one of the given terminals was selected as the starting terminal. Subsequently, the process of selecting the terminal closest to the current interconnection was repeated until immediately before the given length. After repeating this process for all given terminals as a starting point, the tree wherein most terminals are selected is called MCTL (Minimum Cost Tree close to given Length) [3]. In this paper, three types of algorithms are compared: MCTL, uniform memetic genetic, and cluster memetic genetic.

### B. Environment of the Experiment

The implementation environment of the experiment was as follows: Windows 7 Enterprise K (64-bit) operating system running on an Intel® Core™ i5-4460 CPU @ 3.20GHz processor and 8.00GB of RAM. The heuristic algorithm proposed in this paper was written in C++ using Visual Studio 2017. Because each instance is created in a different form, the ratio of the number of terminals included in the MCTL, uniform memetic genetic, and cluster memetic genetic algorithms in this paper is constant; however, the absolute number is inconsistent. Therefore, a comparable figure is required for each algorithm. In this paper, this figure is compared with the percentage of the number of terminals contained by each algorithm based on the number of terminals contained by MCTL. This figure represents the quality of the best individual (QBI). Equation (3) can be used to calculate the QBI:

$$QBI (\%) = \frac{\# \text{ of vertices included in Best Individual}}{\# \text{ of vertices included in the MCTL}} * 100 \quad (3)$$

Python's Sklearn library was used to create the input for the experiment, and the assumptions were as follows: the number of terminals was 1,000 and they were given to a  $1,000 \times 1,000$  Euclidean plane; the number of clusters was 10 and the deviation of each cluster was given randomly; and the given length was 50% of the minimum distance that could connect all terminals.

### C. Performance Comparison according to the Heuristic Algorithm

Compared to the uniform memetic genetic algorithm, the degree of superiority of the cluster memetic genetic algorithm proposed in this paper is compared with the QBI value. Additionally, by comparing the QBI values of the cluster pure genetic algorithm excluding local search from the cluster memetic genetic algorithm, it can be confirmed that the local search proposed in this paper works efficiently. The corresponding QBI value is the average value obtained by performing an average of 10 experiments for each input and proceeding equally with 30 different inputs. Fig. 5 shows a graph comparing the QBI values for each heuristic algorithm.

As evident from Fig. 5, the cluster memetic genetic algorithm showed 5.1% and 22.8% better QBIs than the uniform memetic genetic and the MCTL algorithms, respectively. Considering that MCTL connects an average of 533 terminals to 30 inputs, the cluster memetic genetic algorithm connects 126 more terminals than MCTL and 27 more terminals than the uniform memetic genetic algorithm. This result proves that the cluster memetic genetic algorithm is superior

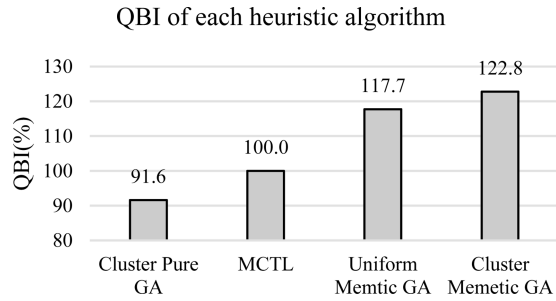


Fig. 5. QBI of each heuristic.

Table 1. Standard deviation of Fig. 5(based on QBI)

Cluster Pure GA	Uniform Memetic GA	Cluster Memetic GA
2.2846	2.6719	1.7523

Table 2. Runtime of Fig. 5

	Cluster Pure GA	MCTL
Runtime(sec)	13.0815	4.2458
	Uniform Memetic GA	Cluster Memetic GA
Runtime(sec)	180.6390	119.5750

to the uniform memetic genetic algorithm. Additionally, the cluster pure genetic algorithm has a QBI of 91.6%, indicating that its performance is lower than that of MCTL. This result proves that the local search proposed in this paper works efficiently. Table 1 lists the standard deviation of the experiment, and Table 2 lists the average value of each heuristic algorithm execution time.

Considering the standard deviations listed in Table 1, the standard deviation of the cluster memetic genetic algorithm is approximately 1.0 less than that of the uniform memetic genetic algorithm. This proves that the cluster memetic genetic algorithm is more stable and reliable than the uniform-memetic genetic algorithm.

#### D. Performance Comparison according to Cluster Density

As a result of the part C experiment, the value of QBI for each input showed a tendency to pivot according to the density of the cluster. Therefore, we analyzed the best performance when each cluster was distributed at a certain density in the cluster memetic genetic algorithm. The corresponding QBI value is the average value obtained by performing an average of 20 experiments for each input and proceeding equally with 10 different inputs. Each input consists of Levels 1–10, wherein the cluster density of Level 1 is very large, whereas that of Level 10 is very small to form a random distribution. At this time, the densities of all the clusters of each instance are the same. Fig. 6 shows a graph that

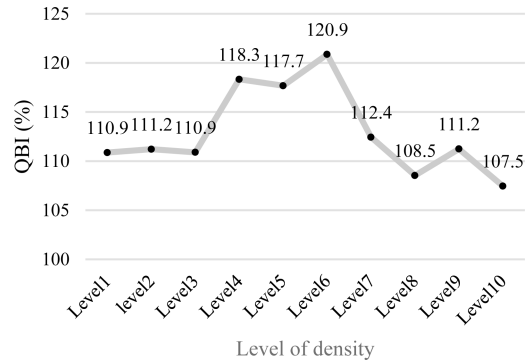


Fig. 6. QBI according to cluster density.

describes the change in the QBI value of the cluster memetic genetic algorithm according to the cluster density.

As observed in Fig. 6, each cluster exhibits the best performance at Level 6, which is in the center of a very large density state and an arbitrary distribution. The values in the experiment showed poor overall performance compared with the experiments based on heuristic algorithm. This is because the corresponding experimental input has almost the same density for all clusters; therefore, the LDD of the local search reflecting the cluster density did not become an environment wherein the search space could be expanded. This proves that the LDD of the local search proposed in this paper appropriately expands the search space in a normal environment.

## V. CONCLUSIONS

To solve the NP-hard problem of connecting the maximum number of terminals with a given length such that not all terminals can be connected, a method using the uniform memetic genetic algorithm has been proposed. However, it only considered a random distribution environment and did not consider a special type of terminal distribution. Therefore, this paper proposed a method of connecting the maximum number of terminals within a given length when the terminals are distributed in a cluster form. Additionally, by comparing it with existing methods, it was proved that it can derive an individual of better quality. The distribution of terminals abstracted in real life is generally in a cluster form. Therefore, the method proposed in this paper is closely related to real life, and significant cost reductions are expected in various fields. The interconnection of the real world has many specificities in addition to a specific distribution. For example, obstacles may exist in a connected environment and the connected edges can be nonlinear. Developing a new method to reflect this particularity remains a challenge for future research.

## ACKNOWLEDGEMENTS

This work was supported by the Catholic University of Korea Research Fund (2020).

## 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, China, 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, Dec. 2019. DOI: 10.6109/jicce.2019.17.4.246.
- [4] B. R. Moon, *Easy to Learn Genetic Algorithm: Evolutionary Approach*, Hanbit Media, Seoul, South Korea, 2008.
- [5] K. Singh and S. Sundar, "A hybrid steady-state genetic algorithm for the min-degree constrained minimum spanning tree problem," *European Journal of Operational Research*, vol. 276, no. 1, pp. 88-105, Jul. 2019. DOI: 10.1016/j.ejor.2019.01.002.
- [6] K. Singh and S. Sundar, "A hybrid genetic algorithm for the degree constrained minimum spanning tree problem," *Soft Computing*, vol. 24, pp. 2169-2186, May. 2019. DOI: 10.1007/s00500-019-04051-x.
- [7] P. C. Pop, O. Matei, C. Sabo, and A. Petrovan, "A two-level solution approach for solving the generalized minimum spanning tree problem," *European Journal of Operational Research*, vol. 265, no. 2, pp. 478-487, Mar. 2018. DOI: 10.1016/j.ejor.2017.08.015.
- [8] L. Zhang, H. Chang, and R. Xu, "Equal-width partitioning roulette wheel selection in genetic algorithm," in *Proceeding of International Conference on Technologies and Applications of Artificial Intelligence*, Tainan, Taiwan, pp. 62-67, 2012. DOI: 10.1109/taai.2012.21.
- [9] Y. Zou, Z. Mi, and M. Xu, "Dynamic load balancing based on roulette wheel selection," in *Proceeding of International Conference on Communications, Circuits and Systems*, Guilin, Chin, pp. 1732-1734, 2006. DOI: 10.1109/icccas.2006.285008.
- [10] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 841-855, Jul. 1996. DOI: 10.1109/icmwi.2010.5648133.
- [11] W. Youping, L. Liang, and C. Lin, "An advanced genetic algorithm for traveling salesman problem," in *Proceeding of International Conference on Genetic and Evolutionary Computing*, Guilin, China, pp. 101-104, 2009. DOI: 10.1109/wgec.2009.127.
- [12] 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, China, pp. 516-519, 2009. DOI: 10.1109/icasid.2009.5276990.
- [13] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley Publishing Company, MA: Boston, USA, pp. 75-109, 1974.



### 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 databases and algorithms.



### 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 databases and algorithms.



### Byungyeon Hwang

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