JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 24, No. 10: 1355~1360, Oct. 2020

동일선상 위치들에 대한 지도 레이블링

김재훈*

Map Labeling for Collinear Sites

Jae-hoon Kim*

*Professor, Division of Computer Software, Busan University of Foreign Studies, Busan, 46234 Korea

요 약

지도에서 특별한 지형지물에 이름이나 설명문에 해당하는 레이블을 붙이는 것을 지도 레이블링(map labeling)이라고 한다. 본 논문에서는 직선상에 n개의 점들이 주어지고 점들에 대한 직사각형의 레이블을 위치시키는 문제를 다룬다. 특별히 레이블들의 높이는 모두 같고, 점들이 위치한 직선의 위쪽에 아래 변들이 모두 동일 선 위에 위치하도록 놓는다. 그리고 점과 레이블을 꺾은선으로 연결하고 이것을 지시선이라 부른다. 지시선은 수직선만으로 구성된 직선 지시선과 수직선, 수평선, 수직선으로 구성된 꺾인 지시선으로 나뉜다. 문제는 꺾인 지시선의 수를 최소화하도록 레이블을 위치시키는 것이고, 우리는 $O(n\log n)$ 시간 알고리즘을 제안함으로서 [13]에서 제안한 $O(n^2)$ 시간 알고리즘을 개선한다.

ABSTRACT

In a map, placing the labels, corresponding to names or explanations of specific features, is called map labeling. In this paper, n points on a line are given, and placing rectangular labels for the points is considered. Particularly, the labels have a same height and their lower sides lie on a straight line in the upper of the line on which the given points are. The points and the labels are connected by polygonal lines, which are called leader lines. The leader lines are classified into straight leader lines and bended leader lines, where the straight leader line consists of only the vertical line and the bended leader line consists of vertical, horizontal, vertical lines. The problem is placing the labels to minimize the number of bended leader lines, and we propose an $O(n\log n)$-time algorithm, which improves the $O(n^2)$-time algorithm previously provided in [13].

키워드 : 지도 레이블링, 레이블, 꺾은선, 지시선, 꺾인 지시선

Key word : map labeling, label, polygonal line, leader line, bended leader line

Received 10 July 2020, **Revised** 23 July 2020, **Accepted** 3 August 2020

*** Corresponding Author** Jae-Hoon Kim(E-mail: jhoon@bufs.ac.kr, Tel:+82-51-509-6226)
Professor, Division of Computer Software, Busan University of Foreign Studies, Busan, 46234 Korea

Open Access http://doi.org/10.6109/jkiice.2020.24.10.1355 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

지도나 도표 안에서 이름이나 설명문을 붙이는 작업은 이러한 것을 만들 때 가장 기본적이면서도 중요한 작업이다. 여기서 이름이나 설명문을 레이블(label)이라고 부르고, 지도나 도표 안에서 레이블의 위치를 결정하는 문제를 레이블 놓기(label placement) 또는 레이블링(labeling)이라고 한다. 레이블 놓기 문제는 여러 공학 분야의 정보 시각화(information visualization) 응용문제에서 나타난다. 특별히, 다이어그램(diagram)이나 지도(map) 그리기(drawing)에서 다각형, 선분, 점들로 표현된 지점들에 대한 정보를 전달하기 위해서 레이블이 사용된다.

레이블 놓기 문제를 다룬 초기 논문들 중 하나인 [1]에서 저자는 지도의 질을 측정하는 기준들을 소개했다. 이전에 지도 제작자들은 자신의 경험에 기초해서 지도에서 레이블이 잘 놓여 졌는지 그렇지 않은지 판단한다. [2]에서 저자는 레이블링 과정의 자동화를 최초로 시도하였다. 이후로 레이블링 문제는 계산 기하학(computational geometry)이나 그래프 드로잉(graph drawing) 과 같은 분야에서 많이 연구되는 주제이다[3-9].

레이블링 문제는 나타내는 지형지물(feature)에 따라서 크게 세 가지 종류로 분류할 수 있다: 첫 번째는 도시와 같이 작은 지역의 위치를 나타내는 점(point) 지형지물의 레이블을 놓는 문제이다. 두 번째는 지도에서 강, 도로, 철로와 같은 것을 나타내는 선(line) 지형지물의 레이블링이다. 마지막으로 세 번째는 산, 호수, 섬과 같은 지역(area) 지형지물에 대한 레이블링이다. 하지만 이 세 가지 종류의 지형지물의 레이블링은 지도의 스케일에 의존한다. 예를 들어, 작은 스케일의 지도에서는 섬들은 점 지형지물로 다루어 질 수 있다.

본 논문에서 다룰 지도 레이블링(map labeling)은 점 지형지물에 대한 레이블링이다. 지도안의 위치들을 평면 상의 점들로 표현할 것이고, 사각형 R안에 포함된 n개의 점들로 나타낸다. 특별히 본 논문에서는 점들이 같은 수평선 상에 위치하는 경우를 다룰 것이다.

지도 레이블링에서 일반적으로 위치들에 대한 레이블들은 서로 겹치지 않고, 가급적 자신의 위치에 가깝게 놓여야 한다. 하지만 위치들이 서로 너무 밀집되어 있으면, 레이블들이 이 조건을 만족시키지 못할 수 있다. 본 논문에서는 위치들의 레이블이 서로 겹치지 않는다면 위치로부터 떨어져 있을 수 있는 경우를 다룰 것이다. 위치들의 레이블은 모두 같은 높이를 가진 직사각형들로 표현하고 위치들이 놓여 있는 직선의 위쪽에 아랫변이 모두 같은 수평선 위에 있도록 놓는다고 가정한다.

구체적으로, 지도상의 위치들을 점으로 표현하고 n개의 점들 p_i는 직선 L위에 놓여있다. 각 점 p_i는 레이블 l_i를 가지고, 레이블 l_i는 너비 w_i와 동일한 높이 1인 위, 아래 변이 직선 L과 평행한 직각 사각형이다. 레이블들은 서로 겹치지 않도록 놓여야 하고 아래 변이 모두 직선 L의 위쪽으로 일정한 수직 거리 h가 되도록 놓여야 한다. 다시 말해서, 직선 L의 위쪽으로 수직 거리 h인 직선 L'에 모든 레이블들의 아래 변이 놓여 있어야 한다(그림 1).

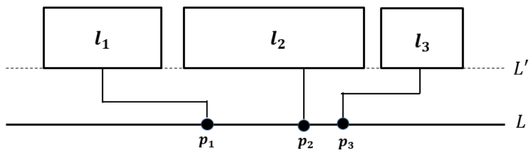

Fig. 1 Map labeling

각 레이블 l_i은 점 p_i에 꺾은선(polygonal line)으로 연결되어야 한다. 이 꺾은선을 지시선(leader line)이라고 한다. 지시선은 두 가지 종류가 있다. 첫 번째는 수직선만으로 구성된 직선 지시선(straight leader line) 이고 두 번째는 수직선, 수평선, 수직선으로 구성된 꺾인 지시선(bended leader line)이다. 본 논문에서는 지시선이 단지 이 두 종류뿐이라고 가정한다. 물론, 지시선들은 서로 교차하지 않아야 한다. 위 그림 1의 레이블링에서 직선 지시선 1개와 꺾인 지시선 2개가 사용되었다.

본 논문에서 레이블 놓기 문제의 최적화 목적 함수는 꺾인 지시선의 개수이다. 우리는 꺾인 지시선들의 개수를 최소화하기 위해서 레이블을 놓는 위치를 결정할 것이다. 아래 그림 2에서 위 그림 1의 점들에 대해서 꺾인 지시선의 개수를 최소화한 레이블링을 보여준다.

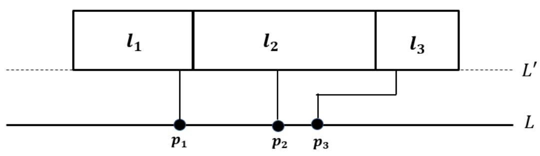

Fig. 2 Minimizing the number of bended leader lines

Ⅱ. 관련 연구

[10]에서는 점에 대한 사각형 레이블을 레이블의 2개 또는 4개 모서리에 점이 위치하도록 놓는 문제를 다루었다. 각각 2-position 또는 4-position 이라고 부른다(그림 3). 2-position 문제에 대해서 저자들은 2-SAT문제로의 변형을 이용해서 $O(n\log^2 n)$ 알고리즘을 제안했다. 그러나 그들은 4-position에 대해서는 결정문제가 NP-complete 임을 증명한다. 따라서 4-position 문제에 대해서 저자들은 모든 레이블이 같은 크기일 때, 최적에 적어도 반 크기의 레이블을 보장하는 $O(n\log n)$ 알고리즘을 제안한다.

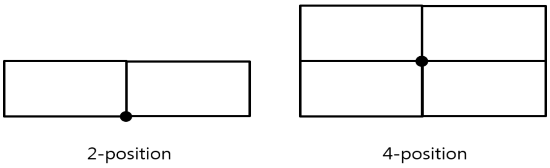

Fig. 3 Fixed-position models

[11]에서는 점들에 대해 놓을 수 있는 레이블의 수를 최대화하는 문제를 다룬다. 저자들은 $O(n\log n)$에 동작하는 $1/\log n$의 근사비를 갖는 근사 알고리즘(approximation algorithm)을 제안한다. 특별히, 모든 레이블의 높이가 동일할 때, 저자들은 근사비 1/2의 근사 알고리즘을 제안한다. 또한 이런 결과들을 확장해서, 임의의 정수 $k \geq 1$에 대해서, 근사비 $k/k+1$인 $O(n\log n + n^{2k-1})$ 시간 근사 알고리즘을 제안한다.

[12]에서는 점이 그것의 레이블의 테두리에 닿으면 충분한 경우에 아래 그림 4와 같은 3가지 모델을 다룬다. 저자들은 1-slider 모델의 경우 레이블 놓기의 결정 문제가 NP-complete 임을 증명한다.

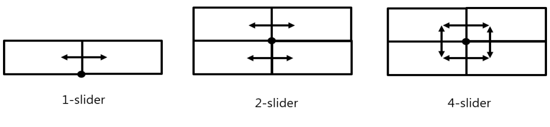

Fig. 4 Slider models

본 논문에서처럼 점들이 밀집되어 있는 경우, 상대적으로 큰 레이블들을 놓기 위해서 레이블을 점들과 떨어져 놓는 대신에 지시선으로 연결하는 문제에 대한 연구들이 있었다[13, 14].

[13]에서 저자들은 변형 가능한 레이블을 다룬다. 이 레이블은 정해진 넓이를 가지면서 높이와 너비를 바꿀 수 있는 레이블을 말한다. 저자들은 직사각형의 테두리에 n개의 점들이 존재할 때, 이 직사각형 안에 변형 가능한 레이블을 놓는 문제에 대한 $O(n^4)$ 시간 결정 알고리즘을 제안한다.

[14]에서는 본 논문에서와 같이 사각형 안에 점들이 존재하고 레이블들을 이 사각형 밖의 네 구역, 위쪽, 아래쪽, 왼쪽, 오른쪽 구역에 레이블을 놓고 직선 지시선 또는 꺾인 지시선으로 연결하는 문제를 다룬다. 특별히, 우리 문제에서처럼 점들이 동일 직선 위에 있을 때, 꺾인 지시선 개수를 최소화하는 문제를 다룬다. 그들은 이 문제에 대해서 $O(n^2)$ 시간 알고리즘을 제안하였다. 본 논문에서는 이 시간 복잡도를 개선하는 새로운 알고리즘을 제안할 것이다.

Ⅲ. 문제의 변형

직선 L위의 n개의 점 p_i는 각각 레이블 l_i을 가지고, 레이블 l_i는 너비 w_i에 동일한 높이를 가지고 있어서 편의상 높이 1로 가정한다. 레이블들은 아랫변이 직선 L의 위쪽에 수직거리 h의 직선 L' 위에 있도록 놓여야 한다. 또한 레이블 l_i과 점 p_i사이에는 직선 지시선 또는 꺾인 지시선으로 연결한다. 꺾인 지시선은 점 p_i로부터 수직, 수평, 수직 선분으로 레이블과 연결한다.

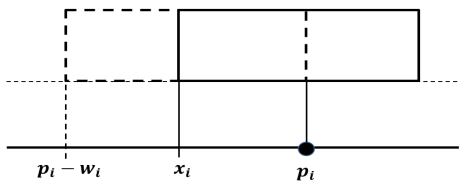

Fig. 5 Constraint for straight leader line

점 p_i는 직선 L상의 점의 위치를 나타내고, 정렬되어 있다고 가정한다. 레이블 l_i를 놓을 위치를 x_i라고 하면, x_i는 레이블 l_i의 왼쪽 변의 x좌표이다. 레이블들은 서로 겹치지 않아야 하기 때문에 다음과 같은 부등식이 성립한다:

$$x_i + w_i \leq x_{i+1},\ \forall i = 1, ..., n-1. \tag{1}$$

우리 문제에서 꺾인 지시선의 수를 최소화하는 것은 직선 지시선을 최대화하는 것과 같고, 이것은 다음 식을 만족하는 i의 개수를 최대화하는 것이다(그림 5):

$$p_i - w_i \le x_i \le p_i. \tag{2}$$

문제를 변형하기 위해서 p_i와 x_i에 대해서 다음을 만족하는 $p_i{}'$와 $x_i{}'$를 생각할 것이다:

$$p_i{}' = p_i - \sum_{j=1}^{i-1} w_j,\ x_i{}' = x_i - \sum_{j=1}^{i-1} w_j. \tag{3}$$

$p_i{}'$와 $x_i{}'$에 대해서 위 식 (1)은 다음 식과 동치이다:

$$x_i{}' \le x_{i+1}{}',\ \forall i = 1, ..., n-1. \tag{4}$$

또한 위 식 (2)는 다음 식과 동치이다:

$$p_i{}' - w_i \le x_i{}' \le p_i{}'. \tag{5}$$

우리 문제가 위 식 (1)과 (2)를 만족하는 x_i들의 최대 개수를 구하는 것이므로, 동치 식 (4)과 (5)에 대해서, 두 식을 만족하는 $x_i{}'$들의 최대 개수를 구하는 문제로 변형할 수 있다. 위 식 (5)를 만족하는 i는 직선 지시선을 가진 레이블을 나타낸다. 앞으로 변형된 문제의 $p_i{}'$과 $x_i{}'$에 대해서 문제를 풀 것이다.

Ⅳ. 알고리즘

본 장에서 우리는 문제에 대한 동적계획법(DP, dynamic programming) 알고리즘을 고안할 것이다. 동적계획법 알고리즘에서 답을 찾기 위한 배열 S의 값들을 재귀식을 이용해서 구한다.

$x_i{}'$, $i = 1, ..., j$, 만을 레이블링 한다고 할 때, 배열 $S[j][x]$의 값은 $x_j{}' \le x$일 때, 꺾인 지시선의 최소 개수로 정의한다. 그러면 $\forall x$, $S[0][x] = 0$이고 우리의 답은 $S[n][\infty]$임을 알 수 있다.

정리 4. 1 위에서 정의된 배열 S는 다음 식을 만족 한다:

$$S[j][x] = \begin{cases} S[j-1][x], & \text{if } p_j{}' - w_j \le x \le u_j \\ S[j-1][x]+1, & otherwise. \end{cases} \tag{6}$$

여기서, u_j는 $S[j-1][p_j{}'] = S[j-1][x]$, $x \ge p_j{}'$를 만족하는 최댓값 x이다.

증명. 배열 $T[j][x]$를 $x_i{}'$, $i = 1, ..., j$, 만을 레이블링 할 때, $x_j{}' = x$ 인 경우, 꺾인 지시선의 최소 개수로 정의한다. 그러면 배열 S는 다음과 같이 T로부터 구할 수 있음을 알 수 있다:

$$S[j][x] = \min_{y \le x} T[j][y]. \tag{7}$$

$x_j{}'$가 위 식 (5)를 만족하면 j는 직선 지시선을 가지고 이외의 경우는 꺾인 지시선을 가지므로 다음 식을 얻을 수 있다:

$$T[j][x] = \begin{cases} S[j-1][x], & \text{if } p_j{}' - w_j \le x \le p_j{}' \\ S[j-1][x]+1, & otherwise. \end{cases} \tag{8}$$

우선 우리는 임의의 j에 대해서, $f(x) = S[j][x]$는 배열 S의 정의에 의해서 $f(x)$는 단조 감소 함수임에 주목한다.

위 식 (7)에 의해서, $S[j][x]$는 $T[j][1]$, ..., $T[j][x-1]$, $T[j][x]$의 최솟값이다. 그런데 $p_j{}' - w_j \le x \le p_j{}'$이면, $T[j][x] = S[j-1][x]$. 따라서 $S[j][x]$는 $S[j-1][1]+1$, ..., $S[j-1][p_j{}' - w_j - 1]+1$, $S[j-1][p_j{}' - w_j]$, ..., $S[j-1][x]$의 최솟값이다. 그런데 f가 단조 감소하므로 $S[j][x] = S[j-1][x]$.

$p_j{}' \le x \le u_j$이면, $S[j][x]$는 $S[j-1][1]+1$, ..., $S[j-1][p_j{}' - w_j]$,..., $S[j-1][p_j{}']$, $S[j-1][p_j{}'+1]+1$, ..., $S[j-1][x]$+1 의 최솟값인데, f가 단조 감소하므로 최솟값은 $S[j-1][p_j{}']$이다. 따라서 $S[j][x]$는 $S[j-1][p_j{}'] = S[j-1][x]$와 같다.

$x > u_j$또는 $x < p_j{}' - w_j$인 경우는, f가 단조 감소하므로 $S[j][x] = S[j-1][x]+1$.

함수 $f_j(x)$를 $f_j(x) = S[j][x]$로 정의하면, 위 정리 4.1에 의해서 f_j는 f_{j-1}로부터 구할 수 있다. 특별히, $p_j{}' - w_j \le x \le u_j$이면, $f_j(x)$는 $f_{j-1}(x)$와 일치하고, $x < p_j{}' - w_j$ 또는 $x > u_j$이면, $f_j(x)$는 $f_{j-1}(x)+1$와 같다. 이 식에 따라 f_0로부터 순서대로 만들어 가면, 함수 f_j는 점들 $p_1{}' - w_1$, u_1, ..., $p_j{}' - w_j$, u_j 에서만 값이

변화하는 계단함수(step function)가 됨을 알 수 있다. 또한 위 정리의 증명에서 언급한 것처럼 f_j는 단조감소이다 (그림 6).

우리의 답은 $f_n(\infty)$의 값이기 때문에, 계단함수 f_j들을 생성하고 $f_j(x)$의 값을 계산할 수 있는 자료구조를 활용할 것이다. 계단함수 f_j는 점들 $x_1, ..., x_k$에서만 값이 변화하고, 구간 $[x_i, x_{i+1}]$에서는 일정한 값을 가진다. 우리가 생성할 자료구조는 각 구간 $[x_i, x_{i+1}]$에서의 함수 f_j의 값을 효율적으로 계산할 것이다.

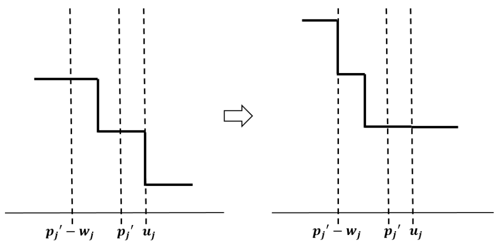

Fig. 6 Transformation of f_{j-1} to f_j

우리는 구간들 $[x_i, x_{i+1}]$에 관한 정보를 저장하기 위해서 구간 트리(range tree) R을 생성할 것이다 (그림 7). 또한 구간 트리 R의 각 노드에 노드가 나타내는 구간의 누적 합을 저장할 것이다. 정리 4.1에 의해서, 함수 f_{j-1}로부터 f_j를 생성할 때, 세 가지 연산을 구간 트리 R에서 수행한다. 우선 구간 $[u_j, \infty)$에 대해서 +1을 수행한다. 다음으로 구간 $[p_j' - w_j, \infty)$에 -1을 수행한다. 마지막으로 구간 $(-\infty, \infty)$에 +1을 수행한다. 그러면 정리 4.1의 식을 만족시키게 됨을 알 수 있다. 이 세 가지 연산은 각각 구간트리 R에서 $O(\log n)$ 시간에 수행할 수 있다. 결과적으로 함수 f_n을 계산하고 $f_n(\infty)$을 답하는데 $O(n \log n)$에 수행할 수 있다.

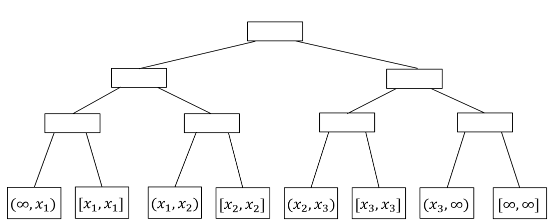

Fig. 7 Range tree R

정리 4. 2 직선 위의 n개 점들에 대한 동일 높이의 직사각형 레이블을 놓을 때, 꺾인 지시선의 수를 최소화하는 $O(n \log n)$ 시간 알고리즘이 존재한다.

Ⅴ. 결 론

본 논문에서는 직선 상의 n개의 점들에 대해 높이가 일정한 직사각형 레이블을 위치시킬 때, 점과 레이블을 연결하는 꺾인 지시선의 수를 최소화하는 문제를 연구하였다. [14]에서는 이 문제에 대해서 $O(n^2)$ 시간 알고리즘을 제안하였다. 본 논문에서 이전 연구의 시간 복잡도를 개선하는 새로운 알고리즘을 제안한다. 이 알고리즘의 시간 복잡도는 $O(n \log n)$으로 이전 연구의 $O(n^2)$을 개선하였다. 향후에 레이블들을 점들의 위쪽, 아래쪽, 왼쪽, 오른쪽에 위치시킬 수 있고 꺾인 지시선들의 수를 최소화하는 문제에 대한 연구를 수행할 수 있을 것이다.

ACKNOWLEDGEMENT

This work was supported by the research grant of the Busan University of Foreign Studies in 2020.

REFERENCES

[1] E. Imhof, “Die anordnung der namen in der karte,” in *International Yearbook of Cartograghy*, pp. 93-129, 1962.

[2] P. Yoeli, “The logic of automated map lettering,” *The Cartographic Journal*, vol. 9, pp. 99-108, Mar. 1972.

[3] F. Klute, G. Li, R. Loffler, M. Nollenburg, and M. Schmidt, “Exploring semi-automatic map labeling,” in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 13-22, 2019.

[4] B. Niedermann and J. -H. Haunert, “An algorithmic framework for labeling network maps,” *Algorithmica*, vol. 80, pp. 1493-1533, Jan. 2018.

[5] F. Krumpe, “Labeling points of interest in dynamic maps using disk labels,” in *Proceedings of the 10th International Conference on Geographic Information Science*, pp. 1-14,

2018.
[6] R. G. Cano, C. C. Souza, and P. J. Rezende, “Fast optimal labelings for rotating maps,” in *Proceedings of the 11th International Conference and Workshops on Algorithms and Computation*, pp. 161-173, 2017.
[7] C. -S. Liao, C. -W. Liang, and S. -H. Poon, “Approximation algorithms on consistent dynamic map labeling,” *Theoretical Computer Science*, vol. 640, pp. 84-93, Aug. 2016.
[8] R. Loffler, M. Nollenburg, and F. Staals, “Mixed map labeling,” *Journal of Spatial Information Science*, vol. 13, pp. 3-32, Dec. 2016.
[9] A. Gemsa, M. Nollenburg, and I. Rutter, “Evaluation of labeling strategies for rotating maps,” *ACM Journal of Experimatal Algorithmics*, vol. 21, pp. 1-21, Aug. 2016.
[10] M. Formann and F. Wagner, “A packing problem with applications to lettering of maps,” in *Proceedings of the 7th Annual ACM Symposium on Computational Geometry*, pp. 281-288, 1991.
[11] P. K. Agarwal, M. Kreveld, and S. Suri, “Label placement by maximum independent set in rectangles,” *Computational Geometry: Theory and Applications*, vol. 11, pp. 233-238, December 1998.
[12] C. Iturriage and A. Lubiw, “NP-hardness of some map labeling problems,” University of Waterloo, Canada, Technical Report CS-97-18, 1997.
[13] C. Iturriaga and A. Lubiw, “Elastic labels around the perimeter of a map,” *Journal of Algorithms*, vol. 47, pp. 14-39, Apr. 2003.
[14] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff, “Boundary labeling: Models and efficient algorithms for rectangular maps,” *Computational Geometry: Theory and Applications*, vol. 36, pp. 215-236, Apr. 2007.

김재훈(Jae-Hoon Kim)

1994 서강대학교 수학과 이학사
1996 KAIST 수학과 이학석사
2003 KAIST 전산과 공학박사
2003~ 부산외국어대학교 컴퓨터소프트웨어학부 교수
※관심분야 : 알고리즘, 최적화, 스케줄링