JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 26, No. 10: 1531~1536, Oct. 2022

점들의 구간 커버에 대한 최대 가중치 맴버쉽 최소화

김재훈*

Minimizing the Maximum Weighted Membership of Interval Cover of Points

Jae-Hoon Kim*

*Professor, Department of Computer Engineering, Busan University of Foreign Studies, Busan, 46234 Korea

요 약

본 논문은 직선상에 n개의 점들과 m개의 구간들이 주어 질 때, 모든 점들을 포함하는 구간들의 집합을 구하는 문제를 다룬다. 이러한 구간들의 집합을 점들의 구간 커버(interval cover)라고 부른다. 이 문제는 NP-hard 문제로 잘 알려진 집합 커버(set cover)의 특별한 경우이다. 이 문제의 최적화 기준으로 커버하는 구간 개수의 최소화, 점을 커버하는 구간이 1개인 점들의 개수 최대화 등을 생각할 수 있다. 본 논문에서는 구간에 가중치가 주어지는 경우, 각 점을 커버하는 구간들의 가중치 합을 그 점의 맴버쉽으로 정의한다. 그리고 점들의 맴버쉽의 최대값을 최소화하는 구간 커버를 찾는 문제를 연구한다. 동적계획법 설계를 이용하여, 이전 연구의 시간 복잡도 $O(nm\log n)$를 개선하는 $O(m^2)$ 시간 알고리즘을 제안한다.

ABSTRACT

This paper considers a problem to find a set of intervals containing all the points for the given n points and m intervals on a line, This is a special case of the set cover problem, well known as an NP-hard problem. As optimization criteria of the problem, there are minimizing the number of intervals to cover the points, maximizing the number of points each of which is covered by exactly one interval, and so on. In this paper, the intervals have weights and the sum of weights of intervals to cover a point is defined as a membership of the point. We will study the problem to find an interval cover minimizing the maximum of memberships of points. Using the dynamic programming method, we provide an $O(m^2)$-time algorithm to improve the time complexity $O(nm\log n)$ given in the previous work.

키워드 : 구간 커버, 집합 커버, 최적화, 맴버쉽, 동적계획법

Key word : interval cover, set cover, optimization, membership, dynamic programming

Received 8 August 2022, **Revised** 12 August 2022, **Accepted** 23 August 2022

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

Open Access http://doi.org/10.6109/jkiice.2022.26.10.1531 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

전체 집합 U와 U의 부분집합 P가 존재하고, U의 원소들로 이루어진 부분집합(subset)들의 모임(family) F를 생각한다. 모임 F의 부분집합 $\overline{F}$에 대해서, $\overline{F}$의 모든 원소들의 합집합이 P를 포함하면, 다시 말해서, $\overline{F} \subseteq F$이고, P의 임의의 원소 u에 대해서, 어떤 집합 $A \in \overline{F}$가 존재해서 $u \in A$를 만족하면, $\overline{F}$를 P의 집합 커버(set cover)라고 한다. 예를 들어, 전체 집합 U를 1부터 10까지의 정수들의 집합이라 하고, $P = \{2,4,6\}$라고 하자. $F = \{\{1,2,3\},\{3,4\},\{5\},\{5,6\}\}$이라 하면, F의 부분집합 $\overline{F} = \{\{1,2,3\},\{3,4\},\{5,6\}\}$는 P의 집합 커버이다.

집합 커버(set cover) 문제는 집합 P를 커버하는 집합 커버 $\overline{F}$ 중에서 크기가 가장 작은 $\overline{F}$를 찾는 것이다. 위의 예에서 F 그 자체도 P의 집합 커버이지만 $\overline{F}$가 가장 작은 크기를 가진다. 이 문제는 컴퓨터 과학에서 잘 알려진 *NP-hard* 문제이다. 따라서 일반적인 집합 커버 문제의 특별한 경우들을 생각해 볼 수 있다. 한 가지는 전제 집합이 이차원 평면이고, P가 평면상의 n개 점들의 집합일 때, 모임 F의 원소들을 평면상의 원, 다각형 등과 같은 기하 물체로 생각할 수 있다. 이 때, $\overline{F} \subseteq F$의 모든 물체들의 합집합이 P를 포함하면 $\overline{F}$를 P의 기하 집합 커버(geometric set cover)라고 부르고, 가장 작은 $\overline{F}$를 찾는 최적화 문제를 생각할 수 있다.

본 논문에서는 전체 집합이 일차원 직선 L로 주어진다. 집합 P는 직선 L상의 n개의 점들로 구성되고, F는 직선 L상의 구간들의 집합이다. 이 때, P의 기하 집합 커버를 생각하고, 이것을 P의 구간 커버(interval cover)라고 부른다. F의 임의의 원소인 구간 I는 자신의 가중치 $w(I)$를 가진다. P의 구간 커버 $\overline{F} \subseteq F$에 대해서, 각 점 $u \in P$에서 u를 포함하는 구간 커버 $\overline{F}$에 속하는 구간들을 생각하고, 이 구간들의 가중치의 합을 u의 맴버쉽(membership)으로 정의한다 (그림 1). 우리는 P의 점들의 맴버쉽의 최대값이 최소가 되는 구간 커버 $\overline{F}$를 찾는 것을 목표로 한다.

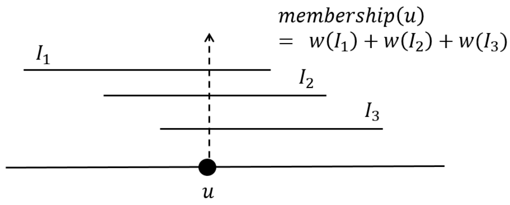

Fig. 1 membership of u

Ⅱ. 관련 연구

전체 집합 U와 부분 집합 P, 그리고 U의 원소들의 부분집합들을 원소로 가지는 모임 F에 대해서, u를 포함하는 정확히 하나의 집합 $A \in \overline{F} \subseteq F$가 존재할 때, P의 원소 u는 $\overline{F}$에 의해 유일하게 커버된다고 한다. [1]의 저자들은 P의 원소들 중 유일하게 커버되는 원소들의 수를 최대화하는 $\overline{F}$을 찾는 문제를 연구한다. 그들은 이 문제가 *NP-hard* 임을 증명하고, 1/18 근사비를 갖는 근사 알고리즘(approximation algorithm)을 제안한다.

본 논문에서는 집합 커버 문제에서 P가 평면상의 점들의 집합이고, 모임 F가 평면상의 물체들의 집합인 경우에 대한 기하 집합 커버 문제에 관심이 있다. 이 경우에 모임 F가 원, 다각형과 같이 평면상의 간단한 물체인 경우에도 *NP-hard*로 알려져 있다. 최근까지도 이 문제에 대한 다양한 결과들이 존재한다 [2, 3, 4, 5].

특별히, [6]에서는 클라이언트를 평면상의 점들로, 기지국들의 통신 범위를 평면상의 원들로 생각해서 원들에 의한 점들의 커버 문제를 연구한다. 원의 반지름을 변화시킬 수 있다고 가정하고 모든 점들을 커버하면서 반지름들의 합을 최소화하는 문제를 생각한다. 저자들은 이 문제에 대한 *PTAS*(polynomial-time approximation scheme)를 제안한다. [7]에서는 원들의 반지름이 고정된 경우, 반지름 1인 원들이 주어질 때, 모든 점들을 커버하는 원들의 최소 개수를 구하는 문제를 연구한다. 저자들은 근사 비 4를 가지는 근사 알고리즘을 제안한다.

점을 커버하는 물체들의 수를 점의 맴버쉽으로 정의하면, [1]의 문제는 맴버쉽이 1인 점들을 최대화하는 커버를 찾는 문제가 된다. [8]에서 맴버쉽이 처음 소개되

었고, 저자들은 점들의 맴버쉽의 최대값이 최소가 되는 커버를 찾는 문제를 연구한다. 그들은 이 문제가 *NP-complete* 임을 증명한다.

[9]에서는 P가 직선상의 점들의 집합이고, 모임 F가 직선상의 구간들의 집합인 경우를 다룬다. 이 때, 점들의 맴버쉽의 최대값이 최소가 되는 구간 커버를 찾는다. 저자들은 점들의 개수 n과 구간들의 개수 m에 대해서, $O((n+m)\log n + m\log m)$ 시간 알고리즘을 제안한다. 또한 구간들에 가중치가 주어지고 점들의 맴버쉽을 커버하는 구간들의 가중치 합으로 정의하는 경우에 $O(nm\log n)$ 시간 알고리즘을 제안한다. 본 논문에서도 구간들에 가중치가 주어진 경우를 다룰 것이다. 우리는 $n = ohm(m)$ 인 경우에 [9]의 결과를 개선하는 $O(m^2)$ 시간 알고리즘을 제안할 것이다.

직선상의 점들에 대한 구간 커버에서 맴버쉽을 고려하지 않고 단지 구간 커버 중 구간의 개수를 최소화하는 문제는 욕심쟁이(greedy) 알고리즘이 존재하는 잘 알려진 문제이다 [10]. 본 논문에서는 점들의 맴버쉽을 고려하는 문제를 다룰 것이다. [11]에서는 단위 길이를 가진 구간들로 구간들을 커버하는 문제를 생각한다. 여기서, 구간들의 끝점 중 하나가 단위 길이를 가진 구간에 포함되면 이 구간이 커버된다고 가정한다. 저자들은 주어진 구간들의 길이가 모두 같은 경우에서도 이 문제가 *NP-hard*임을 증명한다.

Ⅲ. 알고리즘

본 논문의 문제에서 직선상에 n개의 점들이 왼쪽에서 오른쪽으로 $u_1, ..., u_n$로 주어지고, 이들의 집합을 P라 한다. 모임 F는 직선 위의 m개 구간들 $I_1, ..., I_m$을 원소로 가진다. 구간 I_i의 시작점을 $s(I_i)$로 끝점을 $e(I_i)$로 나타낸다. 따라서 $I_i = [s(I_i), e(I_i)]$. 이 때, F의 부분집합으로 P의 구간 커버 $\overline{F}$를 생각할 것이다. P의 점 u_i에 대해서, 구간 커버 $\overline{F}$에 대한 u_i의 맴버쉽을 u_i를 커버하는 $\overline{F}$에 속한 구간들의 가중치 합으로 정의하고, 이를 $membership(u_i)$로 표기한다. 특별히, 이런 구간 커버 $\overline{F}$ 중 P의 임의의 원소 u_i의 맴버쉽의 최대값이 최소가 되는 구간 커버를 찾을 것이다.

이 문제의 답을 최적 구간 커버라고 부르고, 문제의 알고리즘을 설명하기 전에 우선 이 최적 구간 커버가 만족하는 다음 정리를 증명할 것이다.

정리 3. 1 다음을 만족하는 최적 구간 커버 OPT가 존재 한다:

(1) OPT의 어떤 구간도 다른 구간에 포함되지 않는다.
(2) 직선 위의 임의의 수직선은 OPT의 3 개 이상의 구간과 교차하지 않는다.

증명. 첫 번째 명제를 증명하기 위해서 OPT의 어떤 구간 I가 다른 구간 J에 포함된다고 하자. P의 점 u가 구간 I에 포함된다면, u의 맴버쉽은 $w(I)+w(J)$ 이상이다. 따라서, OPT에서 구간 I를 뺀 집합을 OPT'이라고 하면, OPT'이 다시 P의 구간 커버가 됨은 자명하다. 또한 OPT'에서 u의 맴버쉽은 OPT에서의 맴버쉽이하이다. 우리는 맴버쉽의 최대값이 더 작아질 수 있고 구간 I를 포함하지 않는 새로운 구간 커버를 얻을 수 있다 (그림 2).

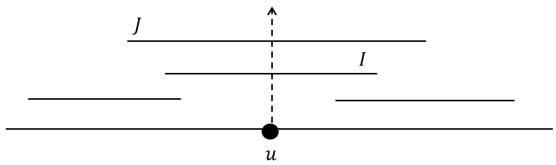

Fig. 2 cancellation of I

두 번째 명제를 증명하기 위해서 직선 위의 어떤 점 p의 수직선에 대해 OPT의 3개 구간 I_1, I_2, I_3가 교차한다고 가정한다. 이 구간들의 시작점 중 최솟값을 s라 하고, 끝점 중의 최대값을 e라고 한다. 일반성의 손실 없이, s는 I_1의 시작점이고 e는 I_3의 끝점이라고 가정하자. 그러면 구간 I_2를 제거하더라도 구간 I_1과 I_3만으로도 구간 $[s, e]$안의 점들을 커버할 수 있다 (그림 3). 다시 말해서, OPT에서 구간 I_2를 제외한 집합을 $\overline{OPT}$라고 할 때, $\overline{OPT}$도 역시 구간 커버가 된다. 또한 I_2가 제외되었음으로 점들의 맴버쉽은 분명히 더 작아진다. 따라서 직선 위의 임의의 수직선에서 2개 이하의 구간만 교차하는 최적 구간 커버를 항상 얻을 수 있다.

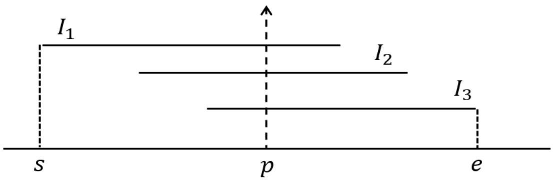

Fig. 3 cancellation of I_2

여기부터 우리는 위 정리 1의 조건을 만족하는 구간 커버를 후보 커버(candidate cover)라고 부를 것이다. 우리는 후보 커버들 중에서 최적 구간 커버 OPT를 찾을 것이다. 그리고 문제의 답을 찾기 위한 동적계획법(Dynamic Programming) 알고리즘을 디자인할 것이다. 이를 위해 임의의 두 구간 I와 J에 대해서, DP 배열 값 $D[I][J]$를 정의할 것이다.

주어진 m개 구간들을 그 오른쪽 끝점이 증가하는 방식으로 정렬한다. 이를 $I_1, ..., I_m$라고 할 때, 후보 커버에 속하는 임의의 두 구간 I_i와 I_j $(i < j)$에 대한 DP 배열 값 $D[I_i][I_j]$를 정의할 것이고, 이를 $D[i][j]$로 나타낸다. 배열 $D[i][j]$의 값은 구간 I_i와 I_j를 포함하고, 끝점 $e(I_j)$ 이하의 모든 점들을 커버하는 후보 커버 C에 대해서, C의 마지막 두 구간이 I_i와 I_j이 경우에 C가 커버하는 점들의 최대 맴버쉽의 최솟값으로 정의한다.

그러면 이렇게 정의된 배열 D의 값들을 구하는 식을 찾는다. 우선 구간 I_i와 I_j가 교차하지 않는 경우를 생각한다. 다시 말해서, $e(I_i) < s(I_j)$인 경우이다. 이 경우에 열린 구간 $(e(I_i), s(I_j))$에 어떤 점 u_k가 존재한다면, 커버 못하는 점이 존재하는 경우임으로 $D[i][j] = \infty$로 정의한다. 만약 구간 $(e(I_i), s(I_j))$안에 점들이 존재하지 않으면, 배열 $D[i][j]$의 값은 다음 식으로 얻을 수 있다 (그림 4):

$$D[i][j] = \max\{\min_{1 \le k < i} D[k][i], w(I_j)\}. \tag{1}$$

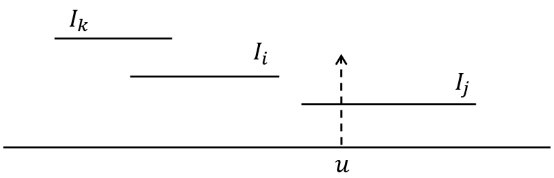

Fig. 4 no point in $(e(I_i), s(I_j))$

이제 구간 I_i와 I_j가 교차하는 경우를 생각한다. 교차하는 부분은 구간을 이룬다. 이 구간을 O라고 하면, 구간 O에 점들이 존재하지 않는 경우부터 살펴본다. 이 경우는 (1)의 식에서 $1 \le k < i$인 구간 I_k 중에서 구간 I_j와 교차하는 않는 구간에 대해서만 $D[k][i]$ 값의 최솟값을 구한다, 왜냐하면, 구간들은 후보 커버가 돼야하기 때문이다 (그림 5).

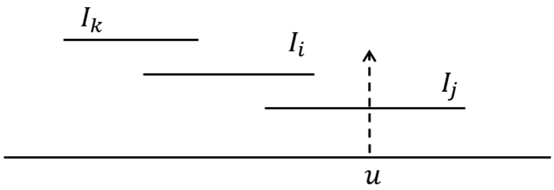

Fig. 5 no point in the overlap of I_i and I_j

마지막으로 교차 구간 O안에 점들이 존재하는 경우를 생각한다. 이 경우 구간 O안에 속하는 점들의 맴버쉽은 $w(I_i) + w(I_j)$이 된다. 따라서 이전 경우와 마찬가지로 $1 \le k < i$인 구간 I_k 중에서 구간 I_j와 교차하는 않는 구간만을 고려해서 식을 세우면 다음과 같은 식을 얻을 수 있다 (그림 6):

$$D[i][j] = \max\{\min_{k < i, I_k \cap I_j = \varnothing} D[k][i], \\ w(I_i) + w(I_j) RIGHT \tag{2}$$

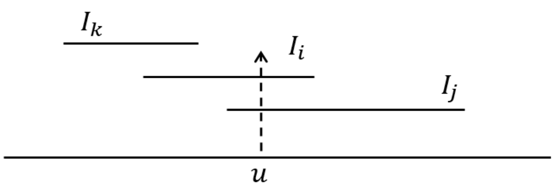

Fig. 6 there is a point in the overlap of I_i and I_j

$i < j$에 대해서 배열 $D[i][j]$의 값은 $O(m^2)$개 존재한다. 따라서 각 $D[i][j]$를 계산하는데 식 (1)과 (2)에서 $\min_{1 \le k < i} D[k][i]$ 또는 $\min_{k < i, I_k \cap I_j = \varnothing} D[k][i]$를 계산하는데 $O(m)$ 시간을 소비하면 배열 D의 값들을 계산하는데 총 $O(m^3)$ 시간이 걸린다. 우리는 이 시간을 줄일 것이다.

우리는 배열 D의 값들을 계산하기 전에 구간들 I_i 에

대한 배열 α와 β를 정의한다. 각 구간 I_i에서 $e(I_k)$이 I_i의 시작점보다 작으면서 $e(I_k)$의 값이 가장 큰 구간 I_k를 찾아서 $\alpha[i]=k$로 정의 한다 (그림 7). 구간의 오른쪽 끝점들이 정렬되어 있으므로 이 점들에 대해서 이진 탐색을 수행해서 점 $s(I_i)$가 $e(I_k)<s(I_i)$를 만족하는 가장 큰 $e(I_k)$를 구할 수 있다. 따라서 배열 α를 총 시간 $O(m\log m)$에 구할 수 있다.

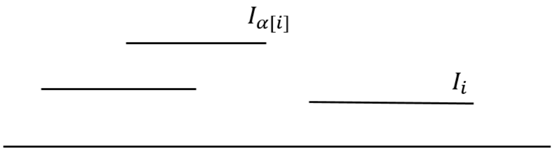

Fig. 7 array α

또한 각 구간 I_i에 대해서, $u_k>e(I_i)$인 가장 작은 점 u_k를 찾아서 $\beta[i]=k$로 정의 한다 (그림 8). 점들을 정렬해서 왼쪽부터 순서대로 스캔하면서 각 구간 I_i에 대해서 위 조건을 만족하는 점 u_k를 찾을 수 있다. 따라서, 배열 β를 총 시간 $O(m+n)$에 구할 수 있다.

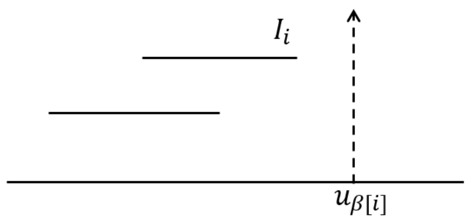

Fig. 8 array β

배열 D의 값들을 row-by-row 순서로 계산할 것이다. 그러면 D의 i번째 행의 계산을 생각한다. 열 $j\geq i+1$를 고정하고 배열 값 $D[i][j]$를 계산하는 경우를 생각하자.

우선 $u_{\beta[i]}<s(I_j)$이면, 점 $u_{\beta[i]}$를 커버하지 못하는 경우이므로 $D[i][j]=\infty$. 또한 $s(I_j)<s(I_i)$이면, 구간 I_i가 구간 I_j에 포함되므로 후보 커버가 되지 못한다. 이 또한 $D[i][j]=\infty$.

위의 경우가 아니면, $u_{\beta[i]}\geq s(I_j)$이고 $u_{\beta[i]-1}\leq e(I_i)$를 만족한다. 우리는 여기서 새로운 배열 B를 소개한다. 배열 $B[i][j]$의 값은 $1\leq k\leq i$인 모든 k에 대해서, $D[k][j]$의 최솟값으로 정의한다. 그러면 식 (1)과 (2)를 배열 B를 포함하는 식으로 다시 쓸 수 있다. 우선 구간 I_i와 I_j가 교차하지 않는 경우에 식 (1)은 다음과 같이 수정될 수 있다:

$$D[i][j]=\max\{B[i-1][i],w(I_j)\}. \tag{3}$$

구간 I_i와 I_j가 교차하는 경우는 $1\leq k<i$인 구간 I_k 중에서 구간 I_j와 교차하는 않는 구간에 대한 $D[k][i]$의 최솟값은 $B[\alpha[j]][i]$와 같다. 따라서 교차 구간에 점들이 존재하지 않는 경우에 다음과 같은 식을 얻을 수 있다:

$$D[i][j]=\max\{B[\alpha[j]][i],w(I_j)\}. \tag{4}$$

또한 위의 식 (2)는 다음과 같이 수정될 수 있다:

$$D[i][j]=\max\{B[\alpha[j]][i],w(I_i)+w(I_j)\}. \tag{5}$$

따라서 위 식 (3), (4), (5)는 모두 $O(1)$ 시간에 계산할 수 있다. 위 식들이 계산되고 난 후, 배열 B에 대한 업데이트가 수행되어야 한다. i번 째 행의 $D[i][j]$ 값들이 모두 계산되고 난 후, 모든 $i+1\leq j\leq m$에 대해서, 배열 $B[i][j]$의 값들이 다음과 같이 업데이트 될 수 있다:

$$B[i][j]=\min\{B[i-1][j],D[i][j]\}. \tag{6}$$

결과적으로 배열 D와 B의 모든 값들을 계산하기 위해서 $O(m^2)$ 시간에 충분함을 보였다.

Ⅳ. 결 론

본 논문에서는 직선상의 n개 점들과 m개 구간들이 주어질 때, 점들을 커버하면서 점들의 맴버쉽의 최대값이 최소가 되는 구간 커버를 찾는 문제를 다룬다. 동적 계획법 알고리즘을 사용하여 두 구간 I_i와 I_j에 대한 DP 배열 값 $D[i][j]$를 총 $O(m^2)$ 시간에 계산한다. 이 알고리즘은 [9]에서 제안한 $O(nm\log n)$ 시간 알고리즘을 $n=ohm(m)$인 경우에 $O(m^2)$ 시간으로 개선한다. 향후에 구간들의 가중치가 모두 동일한 경우, 점의 맴버쉽이 커버하는 구간의 개수로 정의되는 경우에 [9]에서 제안한 알고리즘의 시간 복잡도를 개선하는 알고리즘에 대해서 연구할 수 있을 것이다.

REFERENCES

[1] T. Erlebach and E. J. van Leeuwen, “Approximating geometric coverage problems," in *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, San Francisco: CA, USA, pp. 1267-1276, 2008.

[2] C. Contardo and A. Hertz, “An exact algorithm for a class of geometric set cover problems,” *Discrete Applied Mathematics*, vol. 300, pp. 25-35, Sep. 2021.

[3] T. M. Chan and Q. He, “More Dynamic Data Structures for Geometric Set Cover with Sublinear Update Time,” in *Proceedings of the 37th International Symposium on Computational Geometry*, Buffalo: NY, USA, pp. 25:1 -25:14, 2021.

[4] P. K. Agarwal and J. Pan, “Near-Linear Algorithms for Geometric Htting Sets and Set Covers,” *Discrete & Computational Geometry*, vol. 63, pp. 460-482, Jan. 2020.

[5] T. M. Chan and Q. He, “Faster approximation algorithms for geometric set cover,” in *Proceedings of the 36th International Symposium on Computational Geometry*, Zurich: Switzerland, pp. 27:1-27:14, 2020.

[6] N. Lev-Tov and D. Peleg, “Exact Algorithms and Approximation Schemes for Base Station Placement Problems,” in *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory*, Turku: Finland, pp. 90-99, 2002.

[7] A. Biniaz, P. Liu, A. Maheshwari, and M. Smid, “Approximation algorithms for the unit disk cover problem in 2D and 3D," *Computational Geometry*, vol. 60, pp. 8-18, Jan. 2017.

[8] F. Kuhn, P. Rickenbach, R. Wattenhofer, E. Welzl, and A. Zollinger, “Interference in Cellular Networks: The Minimum membership set cover problem,” in *Proceedings of the 11th International Computing and Combinatorics Conference*, Kunming: China, pp. 188-198, 2005.

[9] S. C. Nandy, S. Pandit, and S. Roy, “Covering Points: Minimizing the Maximum Depth,” in *Proceedings of the 29th Canadian Conference on Computational Geomety*, Ottawa: Ontario, Canada, pp. 37-42, 2017.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed, Cambridge, MA: The MIT Press, 2009.

[11] D. Bergen, E. Eiben, R. Ganian, and I. Kanj, “On Covering Segments with Unit Intervals," in *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science*, Montpellier: France, pp. 13:1-13:17, 2020.

김재훈(Jae-Hoon Kim)

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