최소 지름 신장 트리를 구하는 근사 알고리즘

(Approximation Algorithms for a Minimum-Diameter Spanning Tree)

신 찬 수 † 박 상 민 ††
(Chan-Su Shin) (Sang-Min Park)

요 약 이차원 평면에 주어진 n개의 점 집합 P에 대한, 최소 신장 트리(minimum spanning tree, MST)는 P의 점들을 연결한 신장 트리 중에서 에지 길이의 총합이 최소가 되는 트리로 정의된다. P에 대한 신장 트리의 지름(diameter)은 트리의 두 점을 연결한 트리 경로 중에서 최장 경로의 길이로 정의되며, 최소 지름 신장 트리(*minimum-diameter spanning tree*, MDST)는 P에 대한 신장 트리 중에서 지름이 가장 작은 트리를 의미한다. 현재까지 알려진 가장 좋은 알고리즘[3]은 MDST를 $O(n^3)$ 시간에 구한다. 본 논문에서는 MDST의 지름보다 최대 5/4배 이내의 지름을 보장하는 신장 트리를 구하는 $O(n^2\log^2 n)$ 시간 근사 알고리즘(approximation algorithm)을 제시한다. 이것은 MDST 문제에 관한 첫 번째 근사 알고리즘이다.

키워드 : 기하알고리즘, 근사 알고리즘, 최소 지름 신장 트리

Abstract Let P be a set of n points in the plane. A minimum spanning tree(MST) is a spanning tree connecting n points of P such that the sum of lengths of edges of the tree is minimized. A diameter of a tree is the maximum length of paths connecting two points of a spanning tree of P. The problem considered in this paper is to compute the spanning tree whose diameter is minimized over all spanning trees of P. We call such tree a minimum-diameter spanning tree(MDST). The best known previous algorithm[3] finds MDST in $O(n^3)$ time. In this paper, we suggest an approximation algorithm to compute a spanning tree whose diameter is no more than 5/4 times that of MDST, running in $O(n^2\log^2 n)$ time. This is the first approximation algorithm on the MDST problem.

Key words : Geometric algorithm, approximation algorithm, MDST

1. 서 론

이차원 평면에 주어진 n개의 점들의 집합을 P라 하자. P에 속한 점들이 모두 연결되도록 구성한 트리를 *신장 트리*(spanning tree)라 한다. 본 논문에서 언급하는 모든 트리는 신장 트리라고 가정한다. P에 대한 신장 트리 중 하나를 T라 하자. 그러면, T를 구성하는 노드 개수는 n개이고 각 노드는 P의 한 점을 나타낸다. 본 논문에서는 알고리즘 서술을 간단히 하기 위해, 노드와 점을 구별하지 않고 사용하겠다. T의 점들을 연결하는 에지 개수는 $n-1$개이며, 에지의 길이는 에지의 양 끝 점 사이의 유클리디언 거리로 정의된다. 트리 T에서 두 점 사이의 거리는 두 점을 연결하는 트리의 경로에 포함되는 에지의 길이의 합으로 정의된다. 트리 T에서 가장 멀리 떨어져 있는 두 점 사이의 거리를 T의 지름(diameter)이라 정의한다. P에 대한 신장 트리 중에서 지름이 가장 작은 트리를 *최소 지름 신장 트리*(minimum-diameter spanning tree, 줄여서 MDST)라 한다. 반면에 T에 속한 모든 에지들의 길이의 합이 최소가 되는 신장 트리를 최소 (길이) 신장 트리(minimum spanning tree, 줄여서 MST)라 한다.

이차원 점 집합 P에 대한 MDST는 n개의 점들을 연결하는 기하 네트웍(geometric network)의 하나이다. 각 점을 도시로 생각하고, 여러 도시를 연결하는 트리

· 본 연구는 한국과학재단 목적기초연구 R05-2002-000-00780-0 지원으로 수행되었음.

† 종신회원 : 한국외국어대학교 정보산업대학 디지털정보공학 교수
cssin@hufs.ac.kr
†† 비 회 원 : 한국과학기술원 전산학과
smpark@jupiter.kaist.ac.kr

논문접수 : 2001년 12월 1일
심사완료 : 2003년 2월 21일

네트워을 구성한다고 가정하자. 만약, 트리 네트워을 MST로 한다면 도시를 연결하는 전선의 길의 합이 최소가 되므로 최소의 비용으로 네트워을 설치할 수 있다. 그러나, 트리 네트워에서 가장 멀리 떨어진 두 도시간의 통신 비용이 중요하다면, 네트워의 지름을 최소로 하는 MDST로 구성하는 것이 바람직하다.

점 집합 P에 대한 MST와 MDST를 계산하는 문제는 오래 전부터 많은 연구 결과가 발표되었다. MST는 원래 그래프에서 정의된 문제로 유명하며, $O(e+n\log n)$ 시간에 동작하는 최적 알고리즘[1]이 알려져 있다. 여기서 e와 n은 각각 그래프의 에지와 노드의 개수를 의미한다. 반면에 이차원 점 집합 P에 대한 MST는 $O(n\log n)$ 시간에 구할 수 있다. 이 알고리즘은 점들을 연결하여 삼각형들의 집합으로 표현한다. 이것을 *삼각화*(triangulation)라 한다. 만약, 모든 삼각형의 외접원의 내부에 P의 점이 존재하지 않는다면, 그 삼각화를 특별히 *딜러니 삼각화*(Delaunay triangulation)라 한다. 이 딜러니 삼각화는 P의 점에 대한 평면 그래프(plane graph)로 표현되며, P에 대한 MST는 이 평면 그래프의 부 그래프임이 증명되어 있다[2]. 그래서 그래프의 MST를 계산하는 알고리즘을 그대로 사용할 수 있다. 그런데, 이 평면 그래프의 에지 개수는 노드 개수에 비례하기 때문에, 전체 수행시간이 $O(e+n\log n)=O(n\log n)$이 된다. 물론, 이 수행시간은 최적이다.

반면에 점 집합 P에 대한 MDST를 계산하는 최적 알고리즘은 알려져 있지 않다. 현재까지 알려진 가장 좋은 MDST 알고리즘[3]은 $O(n^3)$에 동작하며, 10년이 지나도록 더 좋은 알고리즘이 발표되지 않고 있다. 그래서, MDST 알고리즘의 수행시간을 줄이는 것은 매우 어렵다고 판단하고, MDST를 직접 구하는 것이 아니라, MDST의 지름과 거의 같으면서 $O(n^3)$ 보다 빠른 시간에 신장 트리를 계산하는 근사 알고리즘(approximation algorithm)을 본 논문에서 제시하려고 한다. 지금까지 알려진 MDST를 구하는 근사 알고리즘은 없다.

근사 알고리즘의 성능은 최적 해와 근사 해의 비율에 따라 달라진다. 예를 들어 P의 MDST의 지름이 D^*이고 어떤 근사 알고리즘에서 계산한 신장 트리의 지름이 D라면 근사 비율은 D/D^*로 정의된다. 그래서, 근사 비율은 일반적으로 1보다 크며, 1에 가까울수록 좋은 성능의 근사 알고리즘이 된다. 본 논문에서는 $O(n^2\log^2 n)$ 시간에 동작하며 근사 비율이 5/4인 MDST 근사 알고리즘을 제시한다.

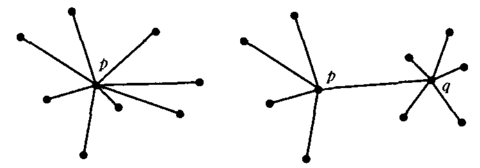

그림 1 (왼쪽) 단항 트리, (오른쪽) 이항 트리

2. (5/4)-근사 비율을 갖는 이항 트리 계산 알고리즘

우선, MDST와 관련된 몇가지 기하학적인 성질부터 살펴보자. 점 집합 P에 대한 신장 트리 중의 하나를 T라 하자. P의 한 점 p에 다른 모든 점들이 연결되어 신장 트리 T가 되었다면, T를 *단항 트리*(monopolar tree)라 부르고, 점 p를 T의 *단항 중심*이라 한다. 만약, P의 두 점 p, q 점을 중심으로 p, q를 제외한 다른 점들이 각각 p, q 중 하나에 연결된 형태라면, T를 *이항 트리*(dipolar tree)라 부르고, 두 점 p, q를 T의 *이항 중심*이라 한다. (그림 1. 참조) Ho 등은 발표한 논문 [Ho]에서 다음의 소정리와 정리를 증명하였다. 자세한 증명은 Ho의 논문에 나와 있다.

소정리 1. [1] *P의 MDST는 단항 신장 트리이거나 이항 신장 트리이다.*

정리 1. [1] *P에 대한 MDST가 단항 신장 트리인 경우엔 $O(n\log n)$시간에, MDST가 이항 신장 트리인 경우엔 $O(n^3)$ 시간에 계산할 수 있다.*

P에 대한 MDST는 위의 소정리 1에 의해 단항 (신장) 트리이거나 이항 (신장) 트리이어야 한다. 결국, 우리는 모든 종류의 신장 트리를 고려할 필요가 없다. 또한, 정리 1에 의해, 최소 지름의 단항 신장 트리는 $O(n\log n)$시간에 구할 수 있으므로, MDST가 이항 신장 트리인 경우에 5/4 배의 근사 비율을 갖는 신장 트리를 $O(n^2\log^2 n)$ 시간에 계산하면 된다.

P에 대한 최소 지름을 갖는 이항 신장 트리를 T_{di}라 하자. T_{di}의 두 이항 중심을 o_1, o_2라 하고, o_1에 연결된 점 중에서 (o_2을 제외하고) 가장 멀리 떨어진 점을 f_1이라 하고 o_2에 연결된 점 중에서 (o_1을 제외하고) 가장 멀리 떨어진 점을 f_2라 하자. 두 점 p, q 사이의 (유클리디언) 거리를 $d(p,q)$로 표기할 때,

$$\delta=d(o_1,o_2),\ \ r_1=d(o_1,f_1),\ \ r_2=d(o_2,f_2)$$

라고 정의한다. 어떤 신장 트리 T의 지름을 $|T|$로 표기한다면, $|T_{di}|=r_1+\delta+r_2$가 된다. 또한, 논의를 간단히 하기 위해, $r_1\ge r_2$ 라고 가정한다. 그러면, 다음과 같은

성질이 성립한다.

소정리 2. *P에 대한 MDST가 이항 트리라면, 즉 MDST가 T_{di}라면, $r_1 \le (\delta + r_2)$ 이다.*

(증명) 만약, $r_1 > (\delta+r_2)$라 가정하자. 중심 o_1에 다른 모든 점을 연결하여 구성한 단항 트리 T를 고려해 보자. 그러면 T_{di}에서 o_2에 연결된 임의의 점부터 o_1까지의 거리는 $\delta+r_2$보다 작거나 같아야 한다. 그래서, $|T| \le \max(r_1+(\delta+r_2), 2(\delta+r_2)) \le r_1+\delta+r_2 = |T_{di}|$가 된다. 결국, 단항 트리 T의 지름이 최소 지름을 갖는 이항 트리 T_{di}의 지름보다 크지 않게 된다. 그렇다면, 최소 지름의 단항 트리의 지름 역시 $|T_{di}|$보다 크지 않다. 결국, P에 대한 MDST는 단항 신장 트리라는 것을 의미하고, 이것은 가정에 모순된다. (QED)

지금부터 지름이 $(5/4)\times|T_{di}|$를 넘지 않는 이항 트리를 계산하는 근사 알고리즘을 설명한다. 이를 위해, δ의 범위에 따라 두 가지 경우로 나누어 고려한다.

2.1 $\delta \ge (r_1+r_2)$인 경우의 (5/4)-근사 이항 트리

이 경우는 T_{di}의 두 중심이 (r_1+r_2)와 비교하여 상대적으로 멀리 떨어져 있다는 것을 의미한다. (5/4)-근사 이항 트리를 구하기 위해, P의 한 점 p에 대한 이항 트리 $T_f(p)$를 먼저 정의한다.

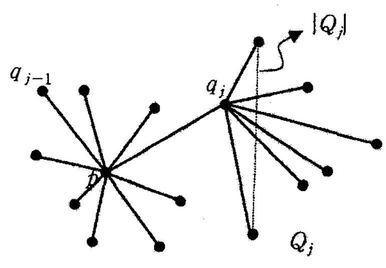

그림 2 이항 트리 $T_f(p, q_j)$.

P의 점들을 p에 가장 가까운 점부터 나열하여 $q_1, q_2, \ldots, q_{n-1}$이라고 하자. (그림 2 참조) 이항 트리 $T_f(p,q_j)$는 점 p와 q_j를 중심으로 하고, $q_1,\ldots,q_{j-1}$은 중심 p에 연결하고, $q_{j+1}, q_{j+2},\ldots,q_{n-1}$까지는 q_j에 연결한다. $Q_j=\{q_j, q_{j+1},\cdots,q_{n-1}\}$라 하고 $|Q_j|$를 집합 Q_j에 속한 점들 가운데 가장 멀리 떨어진 두 점 사이의 유클리디언 거리로 정의하자. 이때, 트리의 비용은 $cost(T_f(p,q_j)) = d(p,q_{j-1}) + d(p,q_j) + |Q_j|$로 정의한다.

$|T_f(p,q_j)| \le cost(T_f(p,q_j))$가 성립함을 쉽게 알 수 있다. 이제 모든 q_j에 대해 $T_f(p,q_j)$를 계산하여, $cost(T_f(p,q_j))$가 최소가 되는 이항 트리를 $T_f(p)$라 하자.

소정리 3. *P의 한 점 p에 대한 이항 트리 $T_f(p)$는 $O(n\log^2 n)$ 시간에 계산할 수 있다.*

(증명) 점 p로부터 다른 점까지의 거리를 계산하여 오름차순으로 정렬하는 것은 $O(n\log n)$시간이 필요하다. 또한, [5]의 결과를 이용하면 $O(n\log^2 n)$ 시간에 모든 j에 대해 $|Q_j|$를 미리 계산할 수 있다. (이것은 $|Q_j|$를 이미 알고 있다면, $|Q_{j-1}|$은 $|Q_j|$를 계산할 때 얻어진 정보를 이용하여 $O(\log n)$ 시간에 계산할 수 있다는 성질을 이용하는 것이다.)

$cost(T_f(p,q_j)) = d(p,q_{j-1}) + d(p,q_j) + |Q_j|$로 정의되는데, 세 항 모두 이미 계산되어 있기 때문에 상수 시간에 $cost(T_f(p,q_j))$를 알 수 있다. 결국, $T_f(p)$를 $O(n\log^2 n)$ 시간에 계산할 수 있다. (QED)

소정리 4. *$\delta \ge r_1$ 이라면, $cost(T_f(o_1)) \le r_1+\delta+2r_2$ 이 된다.*

(증명) 점 o_2보다 o_1에 가까운 P의 점들 중에서 가장 멀리 떨어진 점을 f_1으로 정의했다. 점 f_1을 q_{j-1}이라고 가정하고 $T_f(o_1, q_j)$을 고려해보자. 그러면, $cost(T_p(o_1, q_j)) = r_1 + d(o_1,q_j) + |Q_j|$이 된다.

가정 $\delta = d(o_1, o_2) \ge r_1$ 때문에 o_2가 Q_j에 속하게 된다. 그런데, 점 q_j가 Q_j의 점들 중에서 p에 가장 가까운 점이기 때문에 $d(o_1,q_j) \le d(o_1,o_2) = \delta$가 된다. 또한, 점 f_2 역시 Q_j에 속한다. 이것은 $|Q_j| \le 2r_2$라는 것을 의미한다. 결국,

$cost(T_f(o_1,q_j)) = r_1+\delta+2r_2$이 된다.

$cost(T_f(o_1)) \le cost(T_f(o_1,q_j))$이므로

$cost(T_f(o_1)) \le r_1+\delta+2r_2$ 가 성립한다. (QED)

소정리 5. *$\delta \ge (r_1+r_2)$ 인 경우에, $cost(T_f(o_1)) \le (5/4)|T_{di}|$ 이다.*

(증명) $r_1 \ge r_2$이므로 $r_2 \le (r_1+r_2)/2$가 된다는 것과 $\delta \ge (r_1+r_2)$라는 사실을 이용하면

$cost(T_f(o_1))/|T_{di}| \le (r_1+\delta+2r_2)/(r_1+\delta+r_2) \le 1 + r_2/(r_1+\delta+r_2) \le 1 + (r_1+r_2)/(4(r_1+r_2)) = 5/4$가 된다. (QED)

그러나 P의 어떤 점이 o_1인지 알지 못한다. 그러나, 모든 점 p에 대해, $T_f(p)$를 계산해서 그 중에서 비용이

가장 작은 트리를 새로운 이항 트리 T_f로 선택한다면,
$cost(T_f) \le cost(T_f(o_1)) \le (5/4)|T_{di}|$가 되고
$|T_f| \le cost(T_f)$이기 때문에, $|T_f| \le (5/4)|T_{di}|$가 성립한다. 소정리 3에 의해 한 점 p에 대한 $T_f(p)$는 $O(n\log^2 n)$ 시간에 계산 가능하므로 T_f는 $O(n^2\log n)$시간에 계산할 수 있다.

정리 2. *$\delta \ge (r_1+r_2)$인 경우에 (5/4)-근사 이항 트리 T_f를 $O(n^2\log^2 n)$시간에 계산할 수 있다.*

2.2 $\delta < (r_1+r_2)$인 경우에 (5/4)-근사 이항 트리

3.1 절에서 사용된 이항 트리 $T_f(o_1)$과 함께, 새로운 단항 트리 하나와 이항 트리 하나를 정의하여 (5/4)-근사에 사용한다.

단항 트리 $T_m(p)$는 P의 한 점 p를 트리의 중심으로 하고, P의 다른 점은 모두 p에 연결한 것이다. 그러면 $|T_m(o_1)| \le \max(2r_1,\ r_1+(\delta+r_2),\ 2(\delta+r_2))$가 된다. 그러나 소정리 2에 의해, $r_1 \le \delta+r_2$이므로 $|T_m(o_1)| \le 2(\delta+r_2)$가 된다.

이항 트리 $T_c(p,q)$의 두 중심은 P의 서로 다른 두 점 p, q가 되고, P의 다른 점은 p, q 중에서 가까운 점에 연결된다. $T_c(o_1, o_2)$를 고려하자. (T_{di}에서는 각 점이 반드시 o_1과 o_2 중에서 가까운 점에 연결되지 않을 수도 있기 때문에, $|T_{di}| < |T_c(o_1, o_2)|$인 경우가 발생할 수 있음에 유의하자.) $T_c(o_1, o_2)$의 두 중심에 연결된 점 중에서 가장 먼 점을 각각 f_1, f_2라고 하고 그 점까지의 거리를 r'_1, r'_2라 하자. 그러면 다음 소정리가 성립한다.

소정리 6. $\max(r'_1, r'_2) \le r_1$

(증명) 우선, $r'_1 \ge r'_2$ 이라고 하자. 점 f_1은 T_{di}에서 o_1이나 o_2 중 하나에 연결되어 있다. 만약, o_1에 연결되어 있다면, $r_1 \ge d(o_1, f_1) = r'_1$이 성립한다. 그러나 o_2에 연결되어 있다면, $r_1 \ge r_2 \ge d(o_2, f_1) > d(o_1, f_1) \ge r'_1$ 된다. 결국, $\max(r'_1, r'_2) \le r_1$이 성립한다. $r'_1 \le r'_2$인 경우도 같은 방법으로 증명할 수 있다.(QED)

이 소정리를 이용하면,
$|T_c(o_1, o_2)| \le r'_1 + d(o_1, o_2) + r'_2 \le 2r_1+\delta$가 된다.

r_1과 r_2 상대적인 크기 차이에 따라 다양한 경우로 나누고, 각 경우마다 앞에서 설명한 $T_m(o_1)$, $T_c(o_1, o_2)$, $T_f(o_1)$ 세 트리 중의 하나로 T_{di}를 근사한다. (수식 전개에서 오른쪽에 괄호 안에는 이전 부등식으로부터 전개될 때 사용된 수식 또는 증명된 사실이 나타나 있다.)

경우 1: $r_1 - r_2 \le (r_1+r_2)/3$

이 경우는 $r_1 \le 2r_2$가 됨을 의미한다.

경우 1.1: $r_1 - r_2 \le \delta \le (r_1+r_2)/3$

$$\begin{aligned}
|T_m(o_1)| &\le 2\delta+2r_2\\
&\le (5/4)\delta+(3/4)\delta+(r_1+r_2)\\
&\le (5/4)\delta+(1/4)(r_1+r_2)+(r_1+r_2)\\
&\qquad [\delta \le (r_1+r_2)/3]\\
&\le (5/4)(\delta+r_1+r_2)\\
&\le (5/4)T_{di}
\end{aligned}$$

경우 1.2: $(r_1+r_2)/3 \le \delta \le (r_1+r_2)$

$$\begin{aligned}
|T_c(o_1, o_2)|/|T_{di}| &\le (\delta+2r_1)/(r_1+\delta+r_2)\\
&\le (r_1+\delta+r_2+r_1-r_2)/(r_1+\delta+r_2)\\
&\le 1+(r_1-r_2)/(r_1+\delta+r_2)\\
&\qquad [r_1-r_2 \le (r_1+r_2)/3]\\
&\qquad [\delta \ge (r_1+r_2)/3]\\
&\le 1+((r_1+r_2)/3)/((4/3)(r_1+r_2))\\
&\le 5/4
\end{aligned}$$

경우 2: $(r_1+r_2)/3 \le r_1-r_2 \le 3(r_1+r_2)/5$

이 경우는 $2r_2 \le r_1 \le 4r_2$를 의미한다.

경우 2.1: $(r_1-r_2) \le \delta \le 2(r_1+r_2)/3$

$$\begin{aligned}
|T_m(o_1)|/|T_{di}| &\le 2(\delta+r_2)/(r_1+\delta+r_2)\\
&\le 1+(\delta+r_2-r_1)/(r_1+\delta+r_2)\\
&\le 1+(2(r_1+r_2)/3+r_2-r_1)/(r_1+\delta+r_2)\\
&\qquad [\delta \le 2(r_1+r_2)/3]\\
&= 1+(5r_2/3-r_1/3)/(r_1+\delta+r_2)\\
&\le 1+(r_1/2)/(r_1+\delta+r_2)\\
&\qquad [r_2 \le r_1/2]\\
&\le 1+(r_1/2)/(2r_1)\\
&\qquad [r_1 \le \delta+r_2]\\
&= 5/4
\end{aligned}$$

경우 2.2: $2(r_1+r_2)/3 \le \delta \le (r_1+r_2)$

경우 2.2.1: $\delta \le r_1$

$$\begin{aligned}
|T_m(o_1)| &\le 2\delta+2r_2\\
&\le 2r_1+2r_2\\
&\qquad [\delta \le r_1]\\
&= (r_1+r_2)+(r_1+r_2)\\
&\le 3\delta/2+(r_1+r_2)\\
&\qquad [r_1+r_2 \le 3\delta/2]\\
&= 3(\delta+r_1+r_2)/2-(r_1+r_2)/2
\end{aligned}$$

$$\begin{aligned}
|T_m(o_1)|/|T_{di}| &\le 3/2-(r_1+r_2)/(2(\delta+r_1+r_2))\\
&\le 3/2-(r_1+r_2)/(2(r_1+r_2+r_1+r_2))\\
&\qquad [\delta \le r_1+r_2]\\
&= 3/2-1/4 = 5/4
\end{aligned}$$

경우 2.2.2: $r_1 < \delta$

$$\begin{aligned}
|T_f(o_1)|/|T_{di}| &\le (r_1+\delta+2r_2)/(r_1+\delta+r_2)\\
&= 1+r_2/(r_1+\delta+r_2)\\
&\le 1+r_2/((r_1+r_2)+2(r_1+r_2)/3)\\
&\qquad [\delta \ge 2(r_1+r_2)/3]\\
&= 1+r_2/(5(r_1+r_2)/3)\\
&\le 1+r_2/(5(2r_2+r_2)/3)\\
&\qquad [r_1 \ge 2r_2]\\
&= 1+r_2/(5r_2) = 6/5
\end{aligned}$$

경우 3: $3(r_1+r_2)/5 \le (r_1-r_2)$

이 경우는 $4r_2 \le r_1$를 의미한다.

$$
\begin{aligned}
|T_m(o_1)|/|T_{di}| &\le 1+(\delta+r_2-r_1)/(r_1+\delta+r_2) \\
&\le 1+r_2/r_1 \qquad [\delta \le r_1+r_2,\ r_1 \le \delta+r_2] \\
&\le 1+1/4=5/4 \qquad [4r_2 \le r_1]
\end{aligned}
$$

정리 2에 의해 $T_f(o_1)$은 $O(n^2\log^2 n)$ 시간에 계산할 수 있다. 이제는 $T_m(o_1)$와 $T_c(o_1, o_2)$를 $O(n^2\log n)$ 시간에 구할 수 있음을 보인다. 우선, $T_m(o_1)$은 점 집합 P에 대한 최소 지름을 갖는 단항 트리 T_m와 동일하기 때문에, Ho 등[Ho]이 제시한 $O(n\log n)$-시간에 동작하는 알고리즘을 그대로 이용하면 된다. $T_c(o_1, o_2)$를 구하기 위해선, P의 점들에 대한 모든 쌍 (p, q)에 대해 $T_c(p, q)$를 구해서 그 중에서 지름이 가장 작은 이항 트리 T_c를 구하면 된다. T_c를 구하는 것은 점 집합에 대한 이산 2-중심(discrete 2-center)을 구하는 문제의 변형된 형태이고, 본 논문과 같은 저자에 의해 $O(n^2\log n)$에 계산할 수 있음이 증명되었다[4]. 결국, 다음의 결과가 성립한다.

정리 3. *$\delta < (r_1+r_2)$인 경우에 (5/4)-근사 이항 트리를 $O(n^2\log^2 n)$ 시간에 계산할 수 있다.*

정리 4. *점 집합 P에 대한 (5/4)-근사 이항 트리를 $O(n^2\log^2 n)$ 시간에 계산할 수 있다.*

3. 결 론

본 논문에서는 이차원 점 집합 P의 MDST를 5/4배 이내의 지름을 갖는 신장트리로 근사하는 $O(n^2\log^2 n)$-시간 알고리즘을 제시했다. 앞으로의 연구 방향은 크게 (1) $O(n^2\log^2 n)$ 보다 더 좋은 수행시간을 갖는 근사 알고리즘이나, 5/4배보다 더 좋은 근사율을 갖는 근사 알고리즘을 설계하는 것과 (2) MDST를 근사하지 않고 정확하게 $O(n^3)$ 시간 보다 빨리 계산하는 것이다.

참 고 문 헌

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest: *Introduction to Algorithms,* MIT Press, 1991.

[2] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, 1985.

[3] J.-M. Ho, D. T. Lee, C.-H. Chang, C. K. Wong: Minimum Diameter Spanning Trees and Related Problems. *SIAM J. Comput.* 20(5): 987-997, 1991.

[4] C.-S. Shin, A. Wolff, Min-sum Discrete 2-Center Problem, submitted to *Journal of KISS,* Vol. 29, No. 06, pp. 340-345, 2002.

[5] D. Dobkin and S. Suri, Maintenance of geometric extrema. *Journal of ACM,* 38, 275-298, 1991.

신 찬 수

1991년 서울대 계산통계학과 학사. 1993년 한국과학기술원 전산학 석사. 1998년 한국과학기술원 전산학 박사. 1999년~2000년 홍콩과기대 전산학과연구원. 2000년~2001년 한국과학기술원 전산학과 연구교수. 2001년~현재 한국외국어대학 전자정보학부 디지털 정보공학 조교수. 관심분야는 계산기하학, GIS 알고리즘, 컴퓨터 그래픽스, 그래프 드로잉

박 상 민

2003.2. 한국과학기술원 전산학과 박사
1997.2. 한국과학기술원 전산학과 석사
1995.2. 연세대학교 컴퓨터과학과 학사
관심분야는 알고리즘, 계산기하학, 그래프 이론