

논문 2008-45TC-7-2

센서 네트워크 구축에서의 Combinatorial 기법 적용

(The application of the combinatorial schemes for the layout design of Sensor Networks)

김 준 모*

(Joonmo Kim)

요 약

센서 네트워크에서의 효율적인 라우팅을 위하여 센서노드들을 최적으로 상호연결 하는 배치를 찾는 문제를 고려하게 된다. 유사한 이론 문제로서 평면상에 주어진 점들을 최적으로 상호연결 하는 트리 구조를 찾는 스타이너 트리 문제가 있는데, 이 문제에 대한 근사 알고리즘을 차용하여 센서노드들을 최적에 근사하게 상호연결 하는 배치를 찾을 수도 있다. 하지만 스타이너 트리 문제는 평면상에서 수학적으로 정의된 점들의 집합을 상호연결 하는 문제로서 센서 네트워크에서는 나타나지 않는 특수한 경우까지 내포하므로, 그 알고리즘을 사용하는 접근은 타당한 분석 방식이 될 수 없다. 센서 네트워크에서 센서들의 분포는 평면상의 수학적 점들의 임의적인 분포와는 달리, 센서들이 일정거리 이상으로 서로 떨어져 있다고 가정 할 수 있다. 이러한 물리적인 성격을 반영하여 센서 네트워크를 위한 근사 알고리즘을 구성함으로써, 센서 네트워크 상호연결이라는 문제에 적합한 실행시간 및 최적치에 대한 근사비율을 도출 할 수 있게 된다.

Abstract

For the efficient routing on a Sensor Network, one may consider a deployment problem to interconnect the sensor nodes optimally. There is an analogous theoretic problem: the Steiner Tree problem of finding the tree that interconnects given points on a plane optimally. One may use the approximation algorithm for the problem to find out the deployment that interconnects the sensor nodes almost optimally. However, the Steiner Tree problem is to interconnect mathematical set of points on a Euclidean plane, and so involves particular cases that do not occur on Sensor Networks. Thus the approach of using the algorithm does not make a proper way of analysis. Differently from the randomly given locations of mathematical points on a Euclidean plane, the locations of sensors on Sensor Networks are assumed to be physically dispersed over some moderate distance with each other. By designing an approximation algorithm for the Sensor Networks in terms of that physical property, one may produce the execution time and the approximation ratio to the optimality that are appropriate for the problem of interconnecting Sensor Networks.

Keywords: 센서 네트워크, 그래프 상호연결, NP-hard, 근사 알고리즘, 스타이너 트리

I. 서 론

센서 네트워크에서 노드간의 상호연결 및 효율적인 경로 구성을 위하여 노드의 배치에 관해 분석하여야 할 필요가 있다. 정보 감지가 필요한 위치에 센서노드를 배치하고, 이를 기반으로 센서노드들을 상호 연결 할

수 있도록 연결노드를 사이사이에 적절히 배치하여야 한다. 본 논문은 센서네트워크에서 노드를 배치하는 문제를 추상화하기 위하여 센서네트워크가 설치되는 공간을 평면(Euclidean plane)이라고 가정한다. 그리고 노드의 위치를 평면상의 점으로, 노드의 전파반경을 점을 중심으로 하는 원으로 대응시킨다. 이와 관련하여 여러 가지 주제가 도출 될 수 있으며, 본 논문과 관련하여 다음 두 가지 주제를 고려한다. 하나는 배치하여야 할 연결노드의 개수를 줄이는 방법에 대한 연구가 이며^[1], 또 하나는 노드간의 연결거리의 총합을 줄이는 방법에 대한 연구이다. 전자는 상대적으로 센서노드가 비교적 좁

* 정회원, 단국대학교 전자컴퓨터공학부
(Computer Science & Engineering, Dankook University)

※ 이 연구는 2006학년도 단국대학교 대학연구비의 지원으로 연구되었음.

접수일자: 2008년6월10일, 수정완료일: 2008년7월21일

은 영역에 퍼져 있는 경우에 적합하며, 사용되는 전체 노드의 수를 줄임으로써 결과적으로 센서네트워크에서 중복 전송되는 메시지의 수가 줄게 한다. 이로 인해 각 노드별 사용 전력량을 줄이는 효과를 얻게 된다. 본 논문은 후자에 대한 내용으로, 센서노드가 넓은 영역에 퍼져 있는 경우에는 무엇보다도 노드간의 연결거리의 총합이 가장 우선적으로 고려되어야 할 사항이 된다. 연결거리의 총합이 줄어들면, 이에 따른 경로배정을 포함한 다양한 관점에서의 통신거리가 짧아지기 때문이다. 짧아지는 통신거리는 외란에 의한 에러의 가능성을 줄이는 효과를 포함하여 안정적인 통신환경의 기반이 된다. 그리고 연결거리가 단축됨으로 해서 전자에서와 같이 연결노드의 개수가 비례하여 줄어드는 효과 역시 얻을 수 있다. 본 논문은 이러한 관점에서의 센서 네트워크 배치방법을 주제로 한다. 이는 주어진 조건하에서 최단거리의 센서 네트워크 상호연결을 구성하는 문제이며, 센서 네트워크 상호연결 문제라고 부르겠다. 배경이 되는 이론적인 설명을 위하여 관련 문제^[2~4]를 소개한다. 유클리드 평면 상에 존재하는 n 개의 점들의 집합을 P 라 하자. P 를 상호연결 하는 최단거리의 네트워크는 트리 형태가 되는데, 이를 최소 스타이너 트리(Steiner Minimum Tree)라고 하며 $SMT(P)$ 로 나타낸다. $SMT(P)$ 는 P 에 속하지 않는 점들을 포함할 수 있는데, 이러한 점들을 스타이너 포인트(Steiner points)라고 하며, P 에 속하는 점들을 터미널이라고 부른다. 이때 $SMT(P)$ 를 찾는 문제는 최단 거리 센서 네트워크를 구축하는 문제가 추상화된 것이라고도 볼 수 있다. 그런데, 이 문제는 NP-hard임이 증명되었고^[5], 한정된 절차에 의해 센서 네트워크 상호연결 문제로 변환 될 수 있으므로, 센서 네트워크 상호연결문제 역시 NP-hard이다. 센서노드의 전파반경은 노드 사이의 거리보다 충분히 작다는 성질을 유지하도록 $SMT(P)$ 를 찾는 문제의 인스턴스에서의 점과 점 사이의 거리를 충분히 띄울 수 있도록 배치하면, $SMT(P)$ 를 찾는 문제의 모든 인스턴스를 센서 네트워크 상호연결 문제의 인스턴스로 변환할 수 있다.

이로써 $SMT(P)$ 를 찾는 문제를 위해 고안된 근사 알고리즘을 센서 네트워크 상호연결 문제에 적용할 수 있음을 알 수 있다. 하지만 $SMT(P)$ 를 찾는 문제는 평면상에 주어진 임의의 점들을 상호연결 하는 수학적 문제이기 때문에, 센서 네트워크가 존재하는 물리적인 공간에서는 일어나지 않는 복잡하거나 특수한 경우까지 포함하고 있다. 이는 센서 네트워크 상호연결 문제

에 적합하지 않는 큰 실행시간을 제시하는 원인이 되기도 한다. 센서 네트워크에서 센서들의 분포는 평면상의 수학적 점들의 임의적인 분포와는 달리, 센서들이 일정거리 이상으로 서로 떨어져 있다고 가정할 수 있다. 그리고 이웃하는 센서간의 거리는 무한하게 긴 경우도 없다고 가정할 수 있다. 센서 네트워크가 가지는 이러한 물리적인 성격을 반영하여 논리적인 제한을 가지고, $SMT(P)$ 를 찾는 문제와 관련된 연구결과^[6~8]를 적용하면, 센서 네트워크에서의 상호연결을 위한 근사 알고리즘을 별도로 구성할 수 있다. NP-hard라는 성질을 가진 문제를 위해 구성된 근사 알고리즘은 실용적인 알고리즘은 아니다. 센서 네트워크 상호연결 문제를 전수조사(brute-force search)가 아닌 방법을 통해 이론적으로 어느 정도까지 접근할 수 있는지에 대한 분석이다. 이 분석결과를 기반으로 관련 연구^[7]를 적용하여 Randomized 알고리즘을 구축하면 실용적인 실행시간을 유도해낼 수도 있다. 최적치에 대한 근사 정도를 희생하면, 휴리스틱 알고리즘과 근접해지는 실행시간을 얻을 수도 있다. 본 논문의 근사 알고리즘은 Dynamic Programming(동적 계획법)을 기반으로 한다. 하지만 Dynamic Programming에 이르기까지, 분석과 증명을 거쳐 문제를 재구성하고 PTAS(Polynomial Time Approximation Scheme) 기법을 적용하면서 정립된 절차를 진행해야 한다^[6~8]. 다른 유사문제와 달리 센서 네트워크 상호연결 문제는 PTAS에 적용시키기 적합한 성질을 가지므로 최적치에 대한 근사라는 목표를 향해 바로 진행할 수 있다. 최소화 되어야 할 대상인 목적값(objective value)은, 관련연구^[8]와는 달리, 문제에서 정의된 비용의 총합으로 쉽게 표현되기 때문이다. 다음 장에서는 PTAS 기법을 적용하는 데에 필요한 정의와 본 논문에서 증명하고자 하는 정리 등을 기술한다.

II. 관련연구 및 근접구조 정리

본 논문은 주어진 문제인 센서 네트워크 상호연결 문제에 대해 PTAS를 적용한다. Euclidean TSP 문제를 해결하기 위해 고안된 대표적인 PTAS^[6~8]의 절차를 센서 네트워크 상호연결 문제에 적용하기 위하여 공통적인 정의를 변경하고 정리한다. 문제에서 찾고자 하는 최단거리의 트리구조를 최적연결구조라 한다. 최단거리가 아닌 경우에는 연결구조라 한다. 주어진 문제의 영역은 직사각형 형태로 분할되며, 각 변은 좌표축과 평

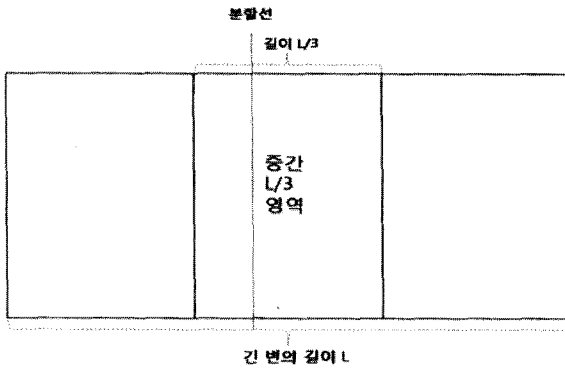


그림 1. 분할선
Fig. 1. Line separator.

행하거나 직교한다. 직사각형의 크기는 긴 변의 길이로 정의한다. **박스**는 주어진 터미널 집합을 에워싸는 최소 면적의 직사각형이다. **비용**은 연결구조가 가지는 간선 길이의 합이다. 직사각형 R 의 분할선은 R 을 두 개의 하위-직사각형으로 분할하는 선분이다. 직사각형 R 의 긴 변의 길이를 L 이라 할 때, 분할선(그림1)은 R 의 중간 $L/3$ 영역에 존재하는 하나의 수직선이 된다. 다음은 직사각형 R 에 대한 재귀적 분할의 정의이며, 이 분할을 기반으로 Dynamic Programming이 실행된다.

정의1 ($\frac{1}{3} : \frac{2}{3}$ -타일링): 직사각형 R 에 대한 $\frac{1}{3} : \frac{2}{3}$ -타일링은 다음과 같은 하위-직사각형들의 바이너리 트리 구조이다. 직사각형 R 은 트리의 루트 노드이다. R 의 크기가 ≤ 1 이면, 트리구조는 R 자신이고, 이외의 경우에는 하나의 분할선으로써 $\frac{1}{3} : \frac{2}{3}$ -타일링 구조를 가지는 두 개의 하위-직사각형으로 분할한다.

타일링 절차에 의해 형성된 직사각형들은, 깊이 d 에서, 루트 직사각형에 대한 파티션(partition)을 형성한다. 깊이 $d+1$ 에 있는 직사각형 집합은 깊이 d 에 있는 직

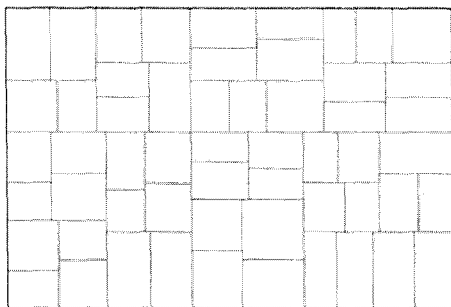


그림 2. $\frac{1}{3} : \frac{2}{3}$ 타일링
Fig. 2. Tiling.

사각형 집합을 세분화 한 것으로 크기가 > 1 인 각각의 직사각형에 대해 분할선을 적용하여 구한다. 따라서 깊이 d 에 있는 직사각형의 면적은 루트 직사각형의 $(\frac{2}{3})^d$ 배 이하이며, 다음의 보조정리1을 도출한다.

보조정리1 직사각형의 길이 L 일 때, $\frac{1}{3} : \frac{2}{3}$ -타일링의 깊이는 $O(\log_{1.5} L)$ 로 표현된다.

정의2 (포털): 포털은 $\frac{1}{3} : \frac{2}{3}$ -타일링에 존재하는 각 직사각형의 변에 존재하는 점(point)으로 각 꼭지점과 각 변의 중앙에 위치한다.

정의3 (light구조): 주어진 문제를 만족하는 연결구조를 π 라 하자. 문제에서 정의된 영역인 박스에 대한 $\frac{1}{3} : \frac{2}{3}$ -타일링을 S 라 하고, 포털의 집합을 P 라고 할 때, π 가 다음을 만족하면 S 에 대한 light 구조라 한다. (i) S 의 직사각형 각각에서 분할선은 π 의 간선들과 최대 세 번 교차한다. (ii) 교차는 P 의 포털에서 일어난다.

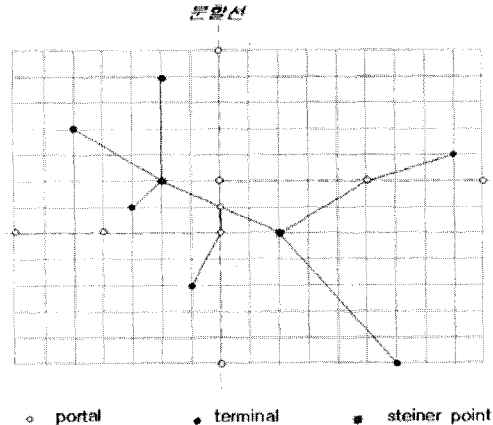


그림 3. Light 구조
Fig. 3. Light structure.

정리 2 (근접구조정리) 각각의 $\epsilon > 0$ 에 대해 다음이 성립한다. 문제에서 주어진 터미널의 집합은 최적연결구조에 $5.5^{O(\log n)}$ -근사하는 light 구조를 가진다.

다음으로 정리2의 증명에 필요한 사항과 실행을 위한 동적 프로그래밍에 대해 먼저 설명하고, 정리 2를 증명한다.

III. 격자좌표로의 변환

주어진 문제는 실수 평면 \mathbb{R}^2 에서 정의되므로, 우리

는 이를 이산화된 좌표계로 변환하여 알고리즘을 구성한다. 이러한 변환은 주어진 문제를 격자 형태로 표현되는 좌표계로 옮기는 것이며, 각각의 *터미널*은 격자평면 상의 가장 가까운 격자점으로 옮겨지는 것이다. 그리고 알고리즘의 실행에 의해 결정되는 각각의 *스타이너* 포인트도 하나의 격자점에 위치한다고 가정한다.

보조정리3에서는, 이렇게 변환된 문제 상에 존재하는 최소 스타이너 트리에 근사하는 그래프를 구하는 것은 실수평면 상의 원래의 문제 상에 존재하는 최소 스타이너 트리에 근사하는 그래프를 구하는 것에 대한 충분조건이 됨을 보인다. 다음의 ϵ 과 $\hat{\epsilon}$ 은 양의 값을 가지는 두 개의 상수이며, 일반적으로 0보다 큰 아주 작은 상수를 의미한다. 최소 스타이너 트리와 오차가 ϵ 인 그래프는 최적에 근사한다는 의미를 가진다.

보조정리3 격자좌표로 변환된 문제에 대한 $(1 + \epsilon)$ 근사는 실수 평면, \mathbb{R}^2 상의 원래 문제에 대한 $(1 + \hat{\epsilon})$ 근사이다.

증명: 변환과 이에 수반되는 가정에 의해 아래의 식이 성립한다. 식에서, $\sum_{\text{최적치}} e$ 는 실수 평면 \mathbb{R}^2 상에 정의된 원래의 문제로부터 구할 수 있는 최소 스타이너 트리의 간선들의 합이고, $\sum_{\text{최적치}} e$ 는 격자평면으로 변환된 문제로부터 구할 수 있는 최소 스타이너 트리의 간선들의 합이다.

$$\left| \sum_{\text{최적치}} e - \sum_{\text{최적치}} e \right| \leq 2(2n-3)$$

최소 스타이너 트리는 $2n-3$ 개의 간선을 가지며, 간선 하나의 양끝에서 단위길이 이하의 오차를 가지기 때

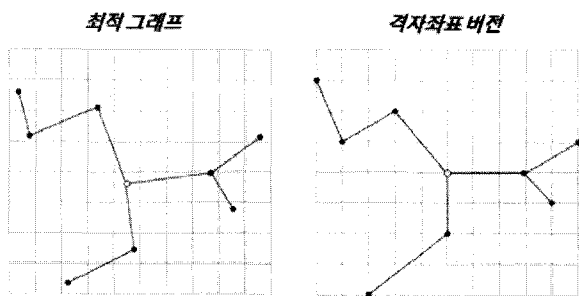


그림 4. 격자좌표 버전
Fig. 4. Grid version.

문이다. 위의 수식을 만족하도록 격자의 단위길이를 조정할 수 있으며, 이러한 조정은 증명의 취지를 벗어나지 않는 범위에서 허용될 수 있다. 참고로, 주어진 문제에서 *터미널*의 개수와 *스타이너* 포인트의 개수를 모두 합하면 최대 $2n-2$ 이 된다. n 개의 *터미널*이 주어질 때, 최대 $n-2$ 개의 *스타이너* 포인트가 존재한다. 마찬가지로 다음이 성립한다.

$$\left| \sum_{\text{근사치}} e - \sum_{\text{근사치}} e \right| \leq 2n^2,$$

이는 최악의 경우 근사치가 complete graph 형태가 될 수 있음을 반영한 것이다. 그리고 다음 식은 충분조건을 나타낸 것이다.

$$\sum_{\text{근사치}} e \leq (1 + \epsilon) \sum_{\text{최적치}} e$$

이제, 위 세 개의 수식을 다음과 같이 종합할 수 있다.

$$\begin{aligned} \sum_{\text{근사치}} e - 2n^2 &\leq (1 + \epsilon) \left(\sum_{\text{최적치}} e + 2(2n-3) \right) \\ \sum_{\text{근사치}} e &\leq (1 + \epsilon) \sum_{\text{최적치}} e + 2(1 + \epsilon)(2n-3) + 2n^2 \\ &\leq (1 + \epsilon + \frac{2(1 + \epsilon)(2n-3) + 2n^2}{\sum_{\text{최적치}} e}) \sum_{\text{최적치}} e \\ &\leq (1 + \epsilon + \frac{2(1 + \epsilon)(2n-3) + 2n^2}{n^3}) \sum_{\text{최적치}} e \\ &\leq (1 + \hat{\epsilon}) \sum_{\text{최적치}} e \quad \blacksquare \end{aligned}$$

위 증명의 수식에서 n^3 의 등장이 중요한 역할을 하는데, 이는 격자평면에서 격자의 단위길이를 충분히 짧게 함으로써 최적치에 근사할 수 있다는 사실을 반영하는 것이다. 즉, 격자의 길이를 조절함으로써 n^3 이상의 차수를 가지는 항을 만들면 된다. 그리고 $\epsilon + \frac{2(1 + \epsilon)(2n-3) + 2n^2}{n^3}$ 부분을 또 하나의 epsilon 인 $\hat{\epsilon}$ 로 나타냄으로써 수식을 정리하였다.

IV. 동적 프로그래밍의 구성과 실행시간

근접구조 정리와 보조정리3를 기반으로 하여 타일링의 최상위 단계까지 *light* 구조를 구축하는 동적프로그래밍(Dynamic Programming, DP)를 구성할 수 있다. 다음 장에서 증명 될 근접구조 정리는 최적치에 $5.5^{O(\log n)}$ -근사하는 *light* 구조와 이에 대응하는 타일링

S의 존재를 보장한다. 이제 S와 최소비용을 가지는 *light* 구조를 $O(n^2)$ 시간에 찾아내는 DP를 설명한다. 단, $O(n^2)$ 는 약 $2^9/3$ 의 큰 계수를 가진다. 이 실행시간에 대한 분석은 후반부에 제시된다. DP의 프로시저는 bottom-up으로 진행하지만, 편의상 최종단계를 시작으로 하여 top-down 방식으로 설명하면 다음과 같다. DP의 최종단계는 주어진 터미널들을 둘러싸는 박스로서, 모든 터미널을 최소의 비용으로 상호연결 하는 *light* 구조를 포함하는 직사각형이다. 최종단계의 직사각형이 구축되기 전에 다음과 같은 조합이 검토되어야 한다. $\frac{1}{3} : \frac{2}{3}$ -타일링에 따라 직사각형을 두 개의 하위-직사각형으로 나누는 모든 분할선이 하나씩 검토되어야 한다. 그리고 각각의 분할선마다 선택 가능한 포털의 모든 조합을 검토하여야 한다. 이러한 절차를 분할된 하위-직사각형에 대해 재귀적으로 적용하면서 진행한다. 하위-직사각형의 크기가 ≤ 1 이 되면, 최하위 level의 직사각형인 *rectangle_B*에 도달한 것이므로 더 이상 분할하지 않고, 이미 구해진 전수조사 결과를 사용한다.

선택된 하나의 분할선은 두 개의 하위-직사각형을 결정하게 된다. 분할선 상에서의 가능한 모든 포털선택은, 포털들에 대한 가능한 모든 multi set으로 표현될 수 있다. 이러한 포털선택 내의 모든 포털과 접하는 양쪽의 하위-직사각형상에서의 *light* 구조 두 개를 접합하여 직사각형 상에서의 *light* 구조를 구축한다. 이렇게 하면 포털선택의 개수와 같은 개수의 *light* 구조들을 산출하게 되는데, 이들 중 최소의 비용을 가지는 것을 선택하여 직사각형 상에서의 최종적인 *light* 구조로 결정한다. 다음 단계의 하위-직사각형에 대해서도 *light* 구조를 구축하기 위한 동일한 절차가 진행되며, 프로시저가 *rectangle_B*에 도달할 때까지 이러한 절차는 순환적으로 진행된다. 그리고 *rectangle_B* 각각은 제한된 개수의 터미널들을 가지게 되므로 전수조사를 통하여 최소 비용을 가지는 *light* 구조를 구축하게 된다. 이상 설명된 모든 조합 각각에 대해 최소 비용의 *light* 구조를 저장하는 lookup table의 entry 개수가 얼마인지, 그리고 각각의 entry를 채우기 위한 실행시간을 파악함으로써 DP의 전체적인 실행시간을 알아본다.

Lookup table에서 하나의 entry는 직사각형, 분할선, 포털의 조합으로 이루어지는 경우들 중의 하나에 대응되며, 최소 비용 *light* 구조를 저장한다. 여기서 하나의

light 구조는 하위의 *light* 구조에 대한 포인터 형태로 entry에 저장된다. 제시된 각 요소가 가지는 전체적인 경우의 수는 다음과 같이 계산된다. (a) 직사각형의 개수: 주어진 problem instance의 직사각형 개수를 구하기 위해 최하위 level의 직사각형의 개수, #*rectangle_B*,를 우선 구한다. *rectangle_B*에서 brute force algorithm을 실행할 수 있도록 *rectangle_B*당 세 개의 터미널을 포함하도록 하는 조건을 적용하면 #*rectangle_B* $\leq n/3$ 가 된다. *rectangle_B*의 상위 level의 직사각형은 *rectangle_B*를 기반으로 두 개씩 합하여 구성되므로 전체 직사각형의 개수는 $< (1 + \frac{1}{2} + \frac{1}{4} + \dots) \cdot \#rectangle_{B} = 2 \cdot \#rectangle_{B} = \frac{2n}{3}$ 이다. (b) 분할선의 개수: 대부분의 직사각형은 분할선으로 이루어진 네 개의 변을 가지게 되며, 각각의 분할선은 $\frac{L}{3}$ 개의 후보 분할선들 중에서 하나가 선택되는 것이다. 따라서 하나의 직사각형은 $\left(\frac{L}{3}\right)^4$ 가지의 조합을 가지게 된다. $\frac{L}{3}$ 은 포털 사이의 거리(오차의 한계치)를 최대로 잡았을 때의 거리이다. 즉, *rectangle_B*가 정사각형이고, 포털이 세 개인 경우를 고려한 길이이다. (c) 하나의 직사각형 둘레에는 최대 8개의 포털이 존재하는데, *light* 구조가 포털을 통과하는 경우의 수는, 전혀 통과하지 않는 경우까지 합하여, $2^8 + 1$ 이다. L은 모든 터미널이 일렬로 늘어선 경우에 최대가 됨을 고려하면 $O(n)$ 으로 나타낼 수 있다. 그러므로 lookup table 크기의 상한은 다음과 같이 나타내어진다.

$$\frac{2n}{3} \cdot \left(\frac{L}{3}\right)^4 \cdot (2^8 + 1) = O(n^2)$$

Lookup table을 채우기 위한 실행시간은 다음과 같이 설명된다. 최하위 단계의 직사각형은 한정된 개수의 터미널과 포털을 가진다. 그러므로 $O(1)$ 시간을 소모하는 전수조사를 통하여 각각의 선택된 경우에 있어서의 최소 비용 *light* 구조를 구할 수 있다. 상위 단계의 직사각형은, 맞닿은 두 개의 하위단계 직사각형을 대상으로, 각각의 선택된 경우에 있어서의 최소 비용 *light* 구조 두 개를 포인팅 함으로써 구할 수 있다. 여기서 선택 가능한 경우는 역시 제한된 포털의 개수에 의해 한정되므로 $O(1)$ 시간을 소모한다. 그러므로 DP의 실행시간은 $O(n^2)$ 라는 상한선을 유지한다. 요약하자면, DP

는 가능한 모든 *light* 구조를 스캔하도록 구성된 것이고, 근접구조정리는 이론상의 최적치에 비해 $5.5^{O(\log n)}$ 배 이내의 비용을 가지는 *light* 구조가 DP에 의해 스캔된 모든 *light* 구조들 중에 존재함을 보이는 것이다.

V. 근접구조 정리의 증명

근접구조 정리는, 일부 *light* 구조들을 이론상의 최적 연결구조와 비용면에서 비교했을 때 그 비율이 타당한 수식으로 표현될 수 있음을 밝힌다. 이러한 *light* 구조가 존재함이 밝혀지면, 모든 *light* 구조를 스캔하는 DP에 의해 구해지는 최소비용의 *light* 구조는 그 비율을 만족하는 것이다. 이론상의 최적연결구조를 현실적으로 찾아낼 수는 없지만, 이 장에서는 최적연결구조와 이에 근접하는 *light* 구조사이의 비용차이를 분석을 통하여 나타낸다. 근접구조정리에 대한 증명의 개요는 다음과 같다. 주어진 문제의 인스턴스에 대해 $\frac{1}{3} : \frac{2}{3}$ -타일링을 적용하여 구한 직사각형 각각에 대해, 최적연결구조와 최소한으로 교차하는 분할선을 선정한다. 그리고 그 교차점들을 목표점들이라 하자. 그 목표점들과 가장 가까운 포털에서 교차하는 *light* 구조가 최적연결구조와 최대 얼마나 차이가 날 수 있는지를 분석하는 것이다. 즉, 목표점들과 그들로부터 가장 가까운 *light* 구조의 부분이 지나는 포털들까지의 거리의 합이 최적연결구조와 다음에 정의되는 근사 *light* 구조의 차이를 나타내는 척도가 되는 것이다. 이러한 길이의 차이를 아래의 보조정리4를 통해 계산되는 최적연결구조의 길이에 대한 최소 추정치와 비교하는 것이다. 높이 L 인 직사각형 내부에서 폭은 단위길이이며 높이는 L 인 하위-직사각형을 단위밴드라 하자.

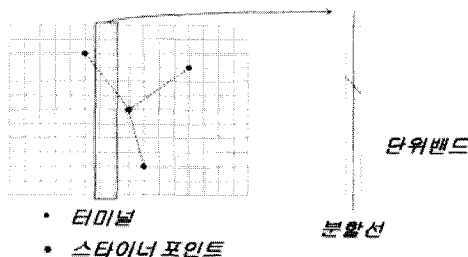


그림 5. 단위밴드
Fig. 5. Unit band.

보조정리4 단위밴드의 중앙을 지나는 분할선이 최적 연결구조와 $k \in N \cup \{0\}$ 회 교차하는 경우, 단위밴드 영

역안의 최적연결구조가 가질 수 있는 비용의 최소값은 $\frac{k}{2}$ 이다.

증명: 단위밴드 내부 영역에서 하나의 교차점을 지나는 최적연결구조의 길이는 가장 짧은 경우에 $\frac{1}{2}$ 이다. 즉, 분할선과 직교하면서 교차지점을 통과하지 않고 닿기만 하는 경우이다. 따라서 단위밴드 영역안의 최적연결구조가 가질 수 있는 비용의 최소값은 $\frac{k}{2}$ 이다. ■

분할선의 길이를 l 이라 할 때, 직사각형 중앙의 분할선을 중심으로 하는 부분-직사각형 영역을 중간 $L/3$ 영역(그림1)이라 하자. $\frac{1}{3} : \frac{2}{3}$ -타일링에 의해 만들어지는 직사각형 각각에 대해, 중간 $L/3$ 영역에서 최적연결구조와 최소로 교차하는 하나의 분할선을 선택하고, 이 교차 회수를 k 라고 하자. 그러면 중간 $L/3$ 영역에서 최적연결구조의 최소비용은 $\frac{k}{2} \cdot \frac{1}{3}L$ 로 표현된다. 우리는 이와 같은 분할선을 바탕으로 최적연결구조에 $5.5^{O(\log n)}$ -근사하는 *light* 구조가 존재함을 밝힐 수 있다. 이러한 *light* 구조를 근사 *light* 구조라고 부른다. 중간 $L/3$ 영역을 고려하는 이유를 살펴보자. 분할선과 교차하는 최적연결구조의 하나의 간선을 a 라하고, 이에 대응하는 근사 *light* 구조의 간선을 b 라 하자. 포털 사이의 거리로 표현되는 a 와 b 의 길이 차이는 삼각형 길이의 부등식 (triangular inequality)에 의해 최대길이가 $\frac{1}{4}l$ 이 된다. 중간 $L/3$ 영역은 이러한 최대 길이를 항상 여유 있게 포함할 수 있도록 $L/3$ 를 폭 길이로 갖는 것이다. ($l \leq L$ 이므로) 근접구조정리의 증명에서는 중간 $L/3$ 영역에서 최적연결구조가 가질 수 있는 최소한의 거리(비용)의 밀도와 근사 *light* 구조가 가질 수 있는 최소한의 거리(비용)의 밀도를 비교함으로써, 중간 $L/3$ 영역에서 간선들이 구체적으로 어떻게 구성되는지를 고려할 필요가 없도록 한다. 중간 $L/3$ 영역은 이러한 밀도를 반영하기 위한 적당한 크기의 영역이며, $\frac{1}{3} : \frac{2}{3}$ -타일링을 고려한 것

이기도 하다. DP는 근사 light 구조 중에서 최소의 비용을 가지는 최소비용 light 구조를 찾아낼 수 있다는 사실을 기반으로 다음 증명을 한다.

증명: (근접구조 정리)

근사 light 구조와 최적연결구조 사이의 비용 차이를 통하여 근사 light 구조가 최적연결구조에 비해 얼마나 더 큰 비용을 가지는지 분석한다. 고려하여야 할 두 가지의 경우가 있는데, 최적연결구조와 분할선의 교점의 수가 포털의 개수인 세 개를 넘느냐 그렇지 않느냐로 구분한다.

경우1) 직사각형을 지나는 분할선이 최적연결구조와 교차하는 회수 k 가 3회 (지정된 포털의 개수) 이하인 경우.

경우2) 직사각형을 지나는 분할선이 최적연결구조와 교차하는 회수 k 가 3회 이상인 경우.

위의 교차횟수는 직사각형의 중간 $L/3$ 영역, 즉 분할선이 존재할 수 있는 곳에서 최적연결구조와 분할선이 최소한으로 교차하게 되는 경우를 기준으로 한다.

경우1)에 해당하는 하나의 직사각형에서 근사 light 구조는 최적연결구조보다 최대 $3k \frac{L}{4}$ 이라는 비용(거리)을 더 갖는다. 근사 light 구조와 포털의 교차점은 최대 3개의 간선을 가지는 스타이너 포인트가 될 수 있는데, 이때 각각의 간선은 최적연결구조의 대응되는 간선보다 포털 간의 거리의 절반인 $\frac{L}{4}$ 만큼 더 길어 질 수 있기 때문이다. 따라서 이러한 경우 하나의 직사각형에서 최적연결구조와의 비용 차이를 비율로 나타내면

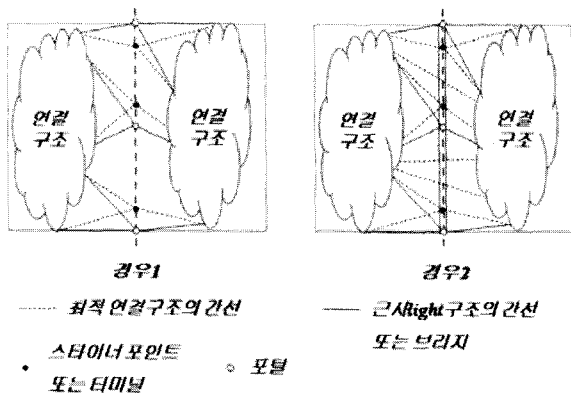


그림 6. 최적연결구조와 근사 light 구조의 비교
Fig. 6. The optimal structure and the approximate light structure.

$\frac{3k \frac{L}{4}}{\frac{k}{2} \cdot \frac{L}{3}} = 4.5$ 가 된다. 경우2)에서, 직사각형의 중간

$L/3$ 영역에서 최적연결구조가 가질 수 있는 최소한의 비

용은 $\frac{3}{2} \cdot \frac{L}{3} = \frac{L}{2}$ 이다. 그 영역에서 최적연결구조와 최소

한으로 교차하게 되는 분할선이 선택되기 때문이다. 경우2)의 직사각형에서는 최적연결구조가 복잡하며 비용이 크기 때문에 대응하는 근사 light 구조를 구성하기위해 길이 L 인 두 개의 브리지를 추가한다. 브리지는 분할선의 양옆에 평행하게 밀착되는 두 개의 선분이다. 밀착이라 함은 둘 사이의 거리가 분석의 결과에 영향을 미치지 않으면서 무시될 수 있음을 뜻한다. 이렇게 함으로써 복잡한 최적연결구조에 근사하는 근사 light 구조를 한정된 시간 내에 구성할 수 있는 것이다. 분할선을 사이에 둔 두 개의 브리지를 임의의 하나의 포털을 통해 연결한다. 근사 light 구조가 포털에서 분할선과 교차하는 대신 브리지와 접함으로써 이웃하는 두 개의 근사 light 구조가 연결되는 셈이다. 이렇게 형성되는 그래프는 스타이너 트리가 아닌 네트워크형태가 되는데, 근사 light 구조 구성의 최종단계에서 네트워크 형태를 스타이너 트리 형태로 바꿀 수도 있지만, 우리의 최종 목표는 스타이너 트리와 비교 가능한 비용을 가지면서 모든 터미널이 상호연결 되는 것이므로 바꾸지 않아도 된다. 결과적으로, 이렇게 형성된 근사 light 구조와 해당하는 최적연결구조 사이의 비용비교는 다음과 같이 나타낼 수 있다. 우선 근사 light 구조는 최적연결구조 보다 최대 $2L$ 이라는 비용을 더 갖는데, 이는 추가된 브리지의 길이이다. 그리고 이 비용을 직사각형의 중간 $L/3$ 영역에서 최적연결구조가 가질 수 있는 최소한

의 비용인 $\frac{3}{2} \cdot \frac{L}{3} = \frac{L}{2}$ 와 비교하면, $\frac{2L}{L/2} = 4$ 가 된다. 결과

적으로, 하나의 직사각형에서 근사 light 구조는 최적연결구조 보다 최대 (1+4.5)배의 비용을 가진다고 볼 수 있다. 그리고 $rectangle_n$ 당 세 개의 터미널을 포함하도록 하는 조건과 보조정리1을 통해, 최상위의 직사각형에 이르기 위해서는 $O(\log n)$ 단계를 거치게 됨을 알 수 있다. 근사 light 구조는 최악의 경우 최적연결구조 보다 $5.5^{O(\log n)}$ 배 큰 비용을 가지게 된다. ■

VI. 결 론

주어진 문제는 NP-hard이므로, 증명의 마지막에 나타난 바와 같이 제안된 근사 알고리즘의 결과치는 최악의 경우에 이론상의 최적치보다 $5.5^{O(\log n)}$ 배만큼 큰 비용을 가지게 된다. 이는 $SMT(P)$ 를 찾는 문제에 대한 근사 알고리즘의 결과치가 $1+\epsilon$ 의 비율로 최적치에 무한하게 접근할 수 있는 것에 비하면 큰 차이를 가지는 것이다. 하지만 $SMT(P)$ 를 찾는 문제에 대한 근사 알고리즘의 실행시간이 $O(n^{1/\epsilon})$ 라는 이론적인 수치로 표현되는 것에 비하면, 제안된 근사 알고리즘에서는 $O(n^2)$ 가 된다. 또한, 관련연구^[6-7]에서 알 수 있듯이 $SMT(P)$ 를 찾는 문제와 유사한 부류의 문제에서는 근사 알고리즘이 일단 주어지면 이를 바탕으로 randomized 알고리즘을 구성하여, 최적치에 대한 근접 비율을 그대로 유지하면서, 실용적인 실행시간을 가지게 할 수도 있다. 실행시간의 관점에 있어서는 근사 알고리즘보다 휴리스틱 알고리즘이 실용적이다. 하지만, 휴리스틱 알고리즘의 결과치와 이론상의 최적치와는 비교되지 않기 때문에 결과치에 대한 평가 또는 분석이 이루어지지 않는다. 휴리스틱 알고리즘의 결과치는 유용한 결과치(feasible solution)로서 실용적인 면에서 허용되는 경우에 의미를 가지는 것이다. 제안된 근사 알고리즘이 가지는 $5.5^{O(\log n)}$ 이라는 비율은 결과치를 평가하고 분석하는 지표로서 의미 있게 사용될 수 있다. 대부분의 관련연구에서처럼, 센서 네트워크가 가지는 다른 물리적인 성질 또는 관리상의 여건을 제안된 분석기법에 추가로 적용하면서 분석을 더 진행하면, 최적치에 대한 비율 및 실행시간을 개선시킬 여지가 있다고 본다. 그리고 제안된 근사 알고리즘은, 센서 네트워크가 구축되는 영역이 넓어서 각각의 센서노드가 평균적으로 두 개의 이웃하는 센서노드와 직접적으로 연결되는 상황에 적합하다. 비교적 좁은 영역 안에서, 직접적으로 이웃하는 센서노드들이 평균 세 개를 넘는 경우에는 본 논문에서와는 다른 관점으로 접근해야 할 것이다^[1]. 이러한 적용범위의 구분을 위한 분석도 계속 연구되어야 할 부분이다.

참 고 문 헌

[1] 김준모, "기하학적 NP-hard문제에 대한 근사적 접근법," 전자공학회 논문지, 제 44권 TC편 제 8월호,

pp.62-67, 2007

- [2] E.N. Gilbert and H.O. Pollak, "Steiner minimal trees," SIAM Journal on Applied Mathematics, Vol. 16, pp.1-29, 1968.
- [3] D. Z. Du and F. K. Hwang, "An approach for proving lower bounds: solution of Gilbert-Pollak conjecture on Steiner ratio," Proceedings of IEEE 31st FOCS, pp.76-85, 1990.
- [4] F. K. Hwang, D.S. Richards and P. Winter., "The Steiner Tree Problem," Annals of Discrete Mathematics, Vol.53, North-Holland, 1992.
- [5] M.R. Garey, R.L. Graham and D.S. Johnson, "The complexity of computing Steiner minimal trees," SIAM J, Appl. Math., 32, pp.835-859, 1977.
- [6] S. Arora, "Polynomial-time approximation schemes for Euclidean TSP and other geometric problems", Proc. 37th IEEE Symp. on Foundations of Computer Science, pp. 2-12, 1996.
- [7] S. Arora, "Nearly linear time approximation schemes for Euclidean TSP and other geometric problems", Proc. 38th IEEE Symp. on Foundations of Computer Science, pp. 554-563, 1997.
- [8] X. Cheng, J.-M. Kim and B. Lu, "A Polynomial Time Approximation Scheme for the Problem of Interconnecting Highways," Journal of Combinatorial Optimization, Vol 5, issue 3, pp. 327-343, 2001.

저 자 소 개



김 준 모(정회원)

1989년 서울대학교 컴퓨터공학과 학사

2001년 University of Minnesota 전산학 박사

2002년~2004년 한국정보보호진흥원 연구원

2004년~현재 단국대학교 전자컴퓨터공학부 조교수

<주관심분야 : Approximations for NP-hard problems>