논문 2009-46TC-5-1

근사접근법 분석을 위한 오차허용치의 분배방법

(Sharing Error Allowances for the Analysis of Approximation Schemes)

김 준 모*, 구 은 희**

(Joonmo Kim and Eunhee Goo)

요 약

센서네트워크를 포함한 다양한 모바일 네트워크를 구축하는 경우, 네트워크를 구성하는 단말 또는 노드들을 상호연결 하기 위한 배치 및 그래프를 찾아내는 문제가 대두된다. 이러한 문제를 해결 할 수 있는 공통적인 scheme을 제시하고, 이를 기반으로 구성되는 알고리즘의 실행시간 및 그 결과의 바운드를 수리적으로 정립하면, 관련 시스템 구축의 타당성을 정확하게 평가 할 수 있게 된다. 본 논문은 이러한 문제를 대표하는 EMST(Euclidean Minimum Spanning Tree) 문제를 대상으로 하여 분산 환경 기반에서 EMST를 병렬처리 형태로 구성할 수 있는 scheme을 제공하고, 이 scheme에 의해 구해 질 수 있는 배치 및 그래프가 EMST와 최대로 어느 정도의 차이를 가지게 되는 지를 판단할 수 있는 기준을 제시한다. 그리고 이 scheme에 의해 구성되는 알고리즘의 실행시간 상한을 제시한다.

Abstract

When constructing various mobile networks including sensor networks, the problem of finding the layout or graph to interconnect the terminals or nodes of the network may come up. Providing a common scheme that can be applied to the kind of problems, and formulating the bounds of the run time and the result of the algorithm from the scheme, one may evaluate precisely the plan of constructing analogous network systems. This paper, dealing with EMST(Euclidean Minimum Spanning Tree) that represents such problems, provides the scheme for constructing EMST by parallel processing over distributed environments, and the ground for determining the maximum difference of the layout or the graph produced from the scheme: the difference from EMST. In addition, it provides the upper bound of the run time of the algorithm from the scheme.

Keywords: 네트워크 layout design, 그래프 상호연결, 스패닝 트리, 파티션, 근사 알고리즘

I. 서 론

어떤 지역에 분산되어 있는 유무선 모바일 단말 및 서버를 연결하여 네트워크를 구축하는 경우, 단말 및 서버를 상호연결(Interconnect)하는 문제[1~4]를 고려하게 된다. 이러한 문제는 유클리디언 평면상에서 Minimum Spanning Tree (EMST)를 구축하는 문제로 추상화되며, 이렇게 추상화된 문제를 기반으로 EMST(Euclidean Minimum Spanning Tree)[5~7]를 구축하기 위한 모델링을 하게 된다. 모델을 기반으로 EMST를 구축하기 위한 방법 및 절차가 구성되고, 관련한 분석이 진행된다. 본 논문에서는 분산 환경에서 병렬처리를 통하여 EMST에 근사하는 네트워크 형태의 그래프(light 구조)를 생산하는 방법(Scheme)을 제안하고 분석하는 것을 주요 내용으로 하고 있다. 지역별로 분산되어 있는 단말 또는 서버의 위치 정보를 병렬처리하면서 통합함으로써 상호연결을 위한 light 구조

* 정회원, 단국대학교 전자컴퓨터공학부
(Member, Computer Science & Engineering, Dankook University)
** 학생회원, 단국대학교 대학원 전자컴퓨터공학과
(Dept. of Electronics & Computer Engineering Graduate School, Dankook University)
※ 이 연구는 2008학년도 단국대학교 대학연구비의 지원으로 연구되었음.
접수일자: 2009년2월11일, 수정완료일: 2009년4월15일

를 구축하는 것이다. 이러한 light 구조는 대부분의 경우 EMST보다 더 긴 길이를 가지게 되는데, 실제 응용에서 이러한 비용의 증가가 허용될 수 있음을 밝히는 데에 필요한 수리적인 기준을 본 논문의 모델링과 분석을 통하여 도출하고 있다. 이렇게 light 구조를 구축하는 근사 알고리즘이 가지는 성능은 그 결과 값의 적정성과 실행시간의 측면에서 분석하여야 한다. 오차정리는 light 구조 및 이를 도출하기 위한 주요 알고리즘을 분석하기 위한 기준이 된다. 오차정리는 EMST에 가장 근사하는 light 구조의 모든 간선의 길이의 합이 EMST의 해당 길이의 합에 비해 어느 정도 길어질 수 있는지를 나타낸다. 이를 통하여, 주어진 문제 영역을 분할하여 구한 light 구조들을 절차에 따라 접합하여 만든 그래프가 EMST에 어느 정도 근사하고, 그 비용이 허용 가능한지를 알 수 있게 된다. 또한, 오차정리를 수립하는 데에 적용된 기법 등을 활용하면, 네크워크 경로배정에 관한 다양한 상호연결(Interconnection) 문제에서, 분할된 정보를 통합하여 목적하는 그래프를 효과적으로 구축하는 방법 및 알고리즘을 도출할 수 있다. EMST는 평면상의 Minimum Spanning Tree 문제이다. 평면에 주어진 점(노드)들을 최단거리로 상호연결 하는 문제이다. 따라서 연결구조가 가지는 간선 길이의 합을 비용으로 하여, 이를 최소화하는 Scheme을 구성 한다. EMST는 그래프 $G=(V,E)$, $|V|=n$ 형태로 표현되며, V의 원소인 각 노드는 유클리디언 평면상의 좌표를 가진다. EMST의 모든 간선 길이의 합을 L_{MST}라 하고, 이에 가장 근사하는 light 구조의 모든 간선 길이의 합을 L_{light}이라고 한다. 평면상의 간선들 사이에 성립하는 Triangular Inequality를 바탕으로 보조정리와 오차정리를 도출하게 된다. 보조정리에서는 실수평면인 유클리드 평면상에 정의된 EMST 문제를 이산좌표로 이동함으로써 생기게 되는 오차를 수식으로 나타내었다. 현실의 기하학적인 문제는 대부분 실수좌표계에서 추상화되므로, 알고리즘 적용을 위해서는 이산화 작업이 다양한 형태로 이루어지는 것이다. 보조정리는 이러한 이산화 작업의 결과로 파생되는 오차 여지의 최대값을 나타내는 것이다. 다음 장에 정의되는 파티션은 병렬처리가 가능한 알고리즘을 구성하기 위하여 주어진 평면 영역을 분할하는 방식이다. 일정크기의 직사각형 형태로 분할된 영역에서 light 구조를 산출하고, 이들을 단계적으로 접합하면서 전체 영역에 대한 light 구조를 구하게 된다. light 구조의 간선이 분할선과 교차하는 경우, 오차를 허용함으로써 분할선 상의 지정된 위치에서만 교차할 수 있도록 한다. 이러한 지정된 교차지점을 포탈이라고 한다. 포탈에 의하여 제한된 복잡도를 가지는 알고리즘의 구성이 가능하게 되고, light 구조와 EMST의 차이를 분석 할 수 있게 된다.

Ⅱ장에서는 관련연구를 기반으로 본 논문에서 사용되는 기초적인 도구를 정의하고 정리를 나타내었다. Ⅲ장에서는 동적프로그래밍을 통하여 light 구조를 구축하는 절차와 해당하는 실행시간을 분석하였다. Ⅳ장에서는 오차정리를 증명하였다.

Ⅱ. 관련연구 및 오차 정리

본 논문은 관련연구[8~10]에서 사용하는 증명 기법을 기반으로 주어진 EMST문제의 성질(Property)을 이용하는 것이다. 관련연구에서는, 주로 평면기하 분야의 NP-hard 문제를 대상으로 주어진 문제영역을 분할한다. 분할된 각 영역에서 최적에 근사하는 해를 구하고, 구해진 영역별 해를 단계적으로 접합하여 최종적으로 최적에 근사하는 해를 구한다. 그리고 구해진 해의 최적치에 대한 근사정도와 해당 알고리즘의 실행시간을 분석한다. 이러한 증명기법에는 문제를 접근하는 공통적인 전략(Scheme)이 있으며, 이러한 전략은 공통적인 틀에 의해 구성된다. 아래의 정의 및 보조정리가 이러한 틀을 제공하는 것이다.

주어진 EMST 문제는 실수좌표 R^2에서 정의되므로, 우리는 이를 이산화된 좌표계로 변환하여 알고리즘을 구성한다. 이러한 변환은 주어진 문제를 격자 형태로 표현되는 좌표계로 옮기는 것이며, 각각의 노드는 이산좌표 상의 가장 가까운 격자점으로 옮겨지는 것이다. 이러한 격자간의 단위길이를 1이라 할때, 보조정리1에서는, 실수좌표 R^2 상에 주어진 EMST 문제를 이산좌표로 옮기는 경우에 발생하는 EMST 전체 간선 합의 최대 변화량을 나타낸다.

보조정리1 실수좌표 R^2에서 정의된 EMST 문제를 이산좌표로 변환하는 경우, 변환에 의한 EMST 전체간선 합의 변화량은 최대 $(n-1)$이다.

증명: 주어진 실수좌표 상의 EMST를 이산좌표로 변환하는 경우, 다음 관계가 성립하게 된다.

$$|\Sigma e_{real} - \Sigma e_{discrete}| \leq (n-1) \quad (1)$$

Σe_{real}은 실수좌표 R^2 상에 정의된 EMST 문제로부터 구할 수 있는 EMST 전체 간선들의 합이고, $\Sigma e_{discrete}$는 이산좌표로 변환된 문제로부터 구할 수 있는 EMST 전체 간선들의 합이다. EMST는 $n-1$개의 간선을 가지며, 간선 하나의 양끝에서는 단위길이 이하의 오차를 가지고, R^2 상의 간선들 사이에서는 triangular inequality가 성립하기 때문이다. ■

유클리디언 평면위에 주어진 노드(점) 집합을 에워싸는 최소 면적의 직사각형을 박스라 한다. 박스는 정의1에서와 같이 *직사각형* 형태로 순환적으로 분할되며, 직사각형의 각 변은 좌표축과 평행하거나 직교한다. 하나의 직사각형을 R이라할 때, 분할선은 R을 두 개의 하위-직사각형으로 분할하는 선분이며, 그 길이는 $L_{separator}$로 표시한다.

정의1 (파티션): 직사각형 R에 대한 파티션은 다음과 같은 하위-직사각형들의 바이너리 트리 구조이다. 직사각형 R은 트리의 루트 노드이다. R의 크기가 ≤ 1이면, 트리구조는 R자신이고, 이외의 경우에는 하나의 분할선으로써 파티션 구조를 가지는 두 개의 하위-직사각형으로 분할된다.

정의2 (포털): 포털은 파티션에서 각 직사각형의 변에 존재하는 점(point)으로 각 꼭지점과 각 변의 중앙에 위치한다.

정의3 (*light 구조*): 주어진 문제를 만족하는 연결

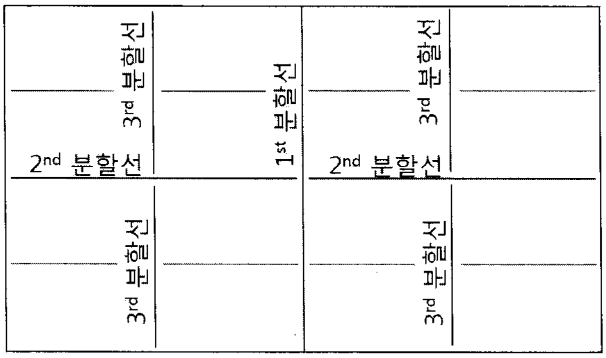


그림 1. 분할에 의한 파티션
Fig. 1. Partition by separation.

그래프를 π라 하자. 문제에서 정의된 영역인 박스에 대한 파티션을 S라 하고, 포털의 집합을 P라고 할 때, π가 다음 두 가지를 만족하면 S에 대한 light 구조라 한다. (1) S의 직사각형 각각에서 분할선은 π의 간선들과 최대 세 번 교차한다. (2) P의 포털에서 교차한다.

정리2 (*오차정리*)

MST의 모든 간선 길이의 합을 L_{MST}라 하고, 이에 가장 근사하는 light 구조의 모든 간선 길이의 합을 L_{light} 이라고 할 때, $L_{light} - L_{MST} \leq \frac{n+1}{4} L_{separator}$인 light 구조가 존재 한다.

다음 장에서는 light 구조를 구축하기 위한 동적 프로그래밍을 설명하면서, 그 실행시간을 분석한다. 그리고 IV 장에서는 오차정리를 증명한다.

III. 동적 프로그래밍의 구성과 실행시간

파티션 상에서 *light 구조*를 구축하는 동적프로그래밍(Dynamic Programming, DP)를 구성하게 된다. 본 장에서는 최소*비용*을 가지는 *light 구조*를 $O(n\log n)$ 시간에 구성하는 DP의 실행시간을 도출한다. 다음 장에서 증명될 오차정리는 EMST 보다 최대 $\frac{n+1}{4} L_{separator}$만큼 더 긴 길이를 갖는 *light 구조*의 존재를 설명한다. 이러한 *오차정리*와 *보조정리1*은 DP의 결과와 EMST와의 오차를 계산하는 기준이 된다. DP의 프로시져는 Bottom-up으로 진행하지만, 편의상 최종단계를 시작으로 하여 Top-down 방식으로 설명하면 다음과 같다. DP의 최종단계는 주어진 노드들을 둘러싸는 *박스*로서, 모든 노드를 최소의 *비용*으로 상호연결 하는 *light 구조*를 포함하는 *직사각형*이다. 최종단계의 *직사각형*이 구축되기 전에 *분할선*에서 선택 가능한 *포털*의 모든 조합을 반영 하여야 한다. 이러한 절차를 분할된 하위-*직사각형*에 대해 재귀적으로 적용하면서 진행한다. 최하위-*직사각형*에 도달하면, MST알고리즘을 사용하여 EMST를 구한다. 이때 주변의 포탈조합을 반영하여 최하위-*직사각형*에서의 EMST 집합을 구하는 것이다. 임의의 *파티션* 단계에서 선택된 하나의 *분할선*은 두 개의 하위-*직사각형*을 결정하게 된다. 분할선 상에서의 가능한 모든 *포털*선택은, *포털*들에 대한 가능한 모든 multi set으로 표현될 수 있다. 이러한 포

털선택 내의 모든 포털과 접하는 양쪽의 하위-*직사각형* 상에서의 *light 구조* 두 개를 접합하여 직사각형 상에서의 *light 구조*를 구축한다. 이렇게 하면 포털선택의 개수와 같은 개수의 *light 구조*들을 산출하게 되는데, 이들 중 최소의 *비용*을 가지는 것을 선택하여 *직사각형* 상에서의 최종적인 *light 구조*로 결정한다. 다음 단계의 하위-*직사각형*에 대해서도 *light 구조*를 구축하기 위한 동일한 절차가 진행되며, 프로시져가 최하위-*직사각형*에 도달할 때까지 이러한 절차는 순환적으로 진행된다. 이상 설명된 모든 조합 각각에 대해 최소 비용의 *light 구조*를 저장하는 Look-up Table의 Entry 개수가 얼마인지, 그리고 각각의 Entry를 채우기 위한 실행시간을 파악함으로써 DP의 전체적인 실행시간을 알 수 있다. Look-up Table에서 하나의 Entry는 *직사각형*, *분할선* 및 포털의 조합으로 이루어지는 경우들 중의 하나에 대응되며, 각 경우에 대한 최소비용 *light 구조*를 저장한다. 여기서 하나의 *light 구조*는 하위의 *light 구조*에 대한 포인터 형태로 Entry에 저장된다.

제시된 각 요소가 가지는 전체적인 경우의 수는, *파티션*의 깊이 d를 지정할 때, 다음과 같이 계산된다. (a) *직사각형*의 개수: 한 단계의 *파티션*에 의해 *직사각형*의 개수가 두 배씩 증가하므로 전체 *직사각형*의 개수는 다음과 같이 정리된다.

$$\sum_{i=0}^{d} 2^i = 2^{d+1} - 1 \qquad (2)$$

(b) 하나의 *직사각형* 둘레에는 최대 8개의 포털이 존재하는데, *light 구조*가 포털을 통과하는 경우의 수는, 전혀 통과하지 않는 경우까지 합하여, $2^8 + 1$이다. 그러므로 Look-up Table 크기의 상한은 다음과 같이 나타내어진다.

$$(2^{d+1} - 1)(2^8 + 1) \qquad (3)$$

Lookup table을 채우기 위한 실행시간은 다음과 같이 설명된다. 최하위 단계의 *직사각형*에서는 EMST를 찾는 알고리즘을 실행한다. $2^{d+1} - 1$개의 *직사각형* 내에서 EMST 알고리즘을 실행하는 시간은 $O(n \log n)$이다. 상위 단계의 *직사각형*은, 맞닿은 두 개의 하위단계 *직사각형*을 대상으로, 각각의 선택된 경우에 있어서의 최소 *비용 light 구조* 두 개를 포인팅 함으로써 구할 수 있다. 여기서 선택 가능한 경우는 제한된 포털의 개수에 의해 한정되므로 $O(1)$ 시간을 소모한다. 그러므로 DP의 실행시간은 $O(n \log n)$라는 상한선을 유지한다. DP는 가능한 모든 *light 구조*를 스캔하도록 구성된 것이다.

Ⅳ. 오차 정리의 증명

EMST는 Delaunay Triangulation의 부분집합인 평면그래프이다[6, 11]. 따라서 EMST에서 교차하는 간선은 존재하지 않는다. 평면위에 주어진 노드를 대상으로 다음의 예와 같이 Delaunay Triangulation을 구성할 수 있으며, 이를 둘러싸는 최소면적의 *직사각형*과 이 *직사각형*을 이등분하는 분할(선)을 표시할 수 있다. 그림에서의 *직사각형*은 표현의 편의상 최소면적의 *직사각형*보다는 크게 그려졌다.

EMST는 Delaunay Triangulation을 구성하는 모든 노드와 일부의 간선으로 이루어진다. 그림 3은 그림 2의 Delaunay Triangulation 상에 존재하는 EMST를 나타낸 것이다.

그림 4에서는 *분할선* 위에 포탈을 설정하여, *분할선* 주변에 *light 구조*를 구성하는 예를 보인다. *분할선*을 교차하는 EMST의 간선들 각각에 대응되는 *light 구조*의 간선들이 구성됨을 보이는 것이다.

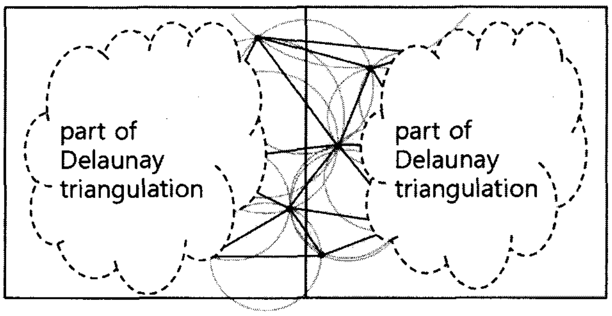


그림 2. triangulation의 파티션
Fig. 2. Partitioning the triangulation.

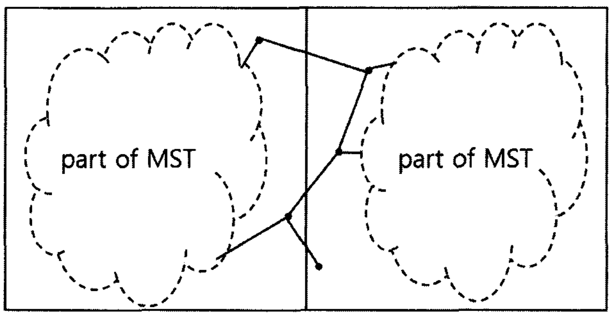


그림 3. MST의 파티션
Fig. 3. Partitioning the MST.

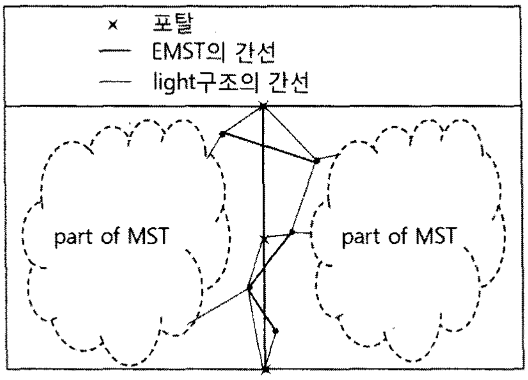


그림 4. light구조의 구성
Fig. 4. Constructing the light structure.

증명: (*오차정리*)

*분할선*을 교차하는 EMST의 간선 하나를 $\overline{ab}$라 하자. 이를 대체하는 *light 구조*의 간선은 $\overline{ap}$와 $\overline{bp}$이다. 여기서 p는, *분할선* 상의 세 개의 포탈 중, $\overline{ap}+\overline{bp}$가 가장 짧아지도록 선정된 하나의 포탈이다.

유클리디언 평면상의 Triangular Inequality에 의하여 다음이 성립한다.

$$\begin{aligned}\overline{ab} &= \overline{ac}+\overline{bc}\\ \overline{ap} &\le \overline{ac}+\overline{pc}\\ \overline{bp} &\le \overline{bc}+\overline{pc}\end{aligned} \tag{4}$$

이를 정리하면,

$$\overline{ap}+\overline{bp} \le \overline{ac}+\overline{bc}+2\overline{pc} = \overline{ab}+2\overline{pc} \tag{5}$$

즉, EMST의 하나의 간선에 해당하는 *light 구조*의 두 개의 간선은 최대 $2\overline{pc}$만큼 더 길어질 수 있다. 포탈간의 거리 $=\frac{1}{2}L_{separator}$이므로, $\overline{pc}\le\frac{1}{4}L_{separator}$이다. $\overline{ap}+\overline{bp}$가 가장 짧아지도록 선택하는 경우, $\overline{pc}$는

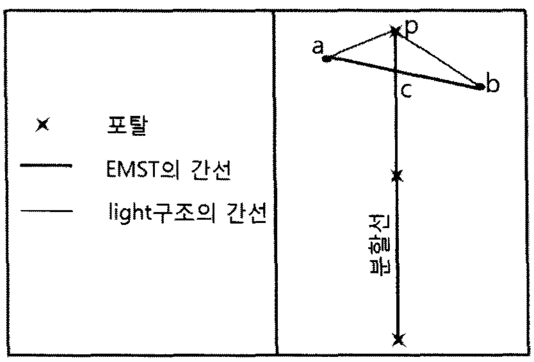


그림 5. 분할선과의 교차
Fig. 5. Crossing with the line separator.

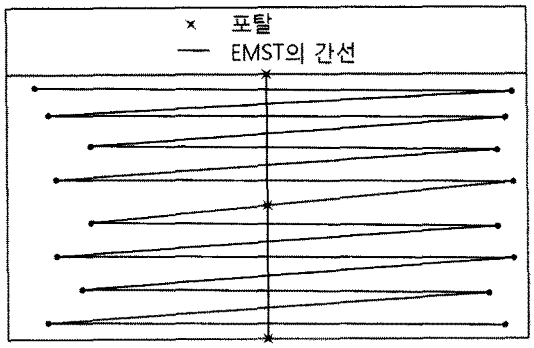


그림 6. EMST 간선과 분할선의 최대 교차의 경우
Fig. 6. The case of maximum crossing between the edges of EMST and the line separator.

포탈간의 거리의 $\frac{1}{2}$이하가 되도록 선택할 수 있기 때문이다.

따라서 하나의 EMST 간선이 *분할선*과 교차하는 경우, *light 구조*는 EMST에 비해 최대 $\frac{1}{2}L_{separator}$만큼 길이가 증가한다. *light 구조*의 전체 길이 증가가 최대가 되는 경우(worst case)는 윗 그림과 같이 $(n-1)$개의 모든 간선이 분할선과 교차하는 경우이다. EMST가 Delaunay Triangulation을 기반으로 하기 때문이다. 이 경우, $(n-1)$개의 간선 중, 첫 번째 간선과 마지막 간선을 제외한 $(n-1)-2$개의 간선은 $\frac{1}{4}L_{separator}$만큼의 길이증가를 유발한다. *light 구조* 간선이 두 번 중복되어, 하나만 사용할 수 있기 때문이다. 따라서 다음 관계가 성립한다.

$$\begin{aligned}&L_{light}-L_{MST}\\ &=2(\frac{1}{2}L_{separator})+(n-3)(\frac{1}{4}L_{separator})\\ &=(\frac{n+1}{4})L_{separator}\end{aligned} \tag{6}$$

■

V. 결 론

넓은 영역에 퍼져있는 모바일 단말 등을 노드로 하여 이들을 상호연결하려는 경우, 영역을 나누어 각 *파티션*에서 EMST를 구축하고, 구축된 각각의 EMST를 접합하면서 전체 영역에 대한 근사-EMST를 구성하는 분산 알고리즘을 구성할 수 있다. 이러한 알고리즘은 영역별로 병렬처리 하도록 구성할 수도 있으며, 따라서 실행시간을 단축시킬 수 있다. 이렇게 영역별로 구축된

단계별 결과치를 접합하는 과정에서 원래의 EMST 보다는 더 긴 전체길이를 가지게 되는데, 이 전체길이의 증가는 결국 단말사이의 통신거리 및 통신비용으로 작용하게 된다. *오차정리*는 위와 같은 분산환경 하의 병렬처리 알고리즘이 얼마나 정확한 결과를 산출하는지를 측정할 수 있도록 함으로써, 이러한 알고리즘 사용의 타당성을 판가름 할 수 있게 하는 기준이 된다. 이러한 알고리즘의 구현 및 실행결과로 산출된 근사-EMST가 주어진 환경 하에서 타당한 비용 증가를 가져온다면, 분산기반의 병렬처리 알고리즘을 이용한 상호연결 방법을 채택하게 되는 것이다. *오차정리*에서는 Worst case의 비용 증가를 고려하였으므로, 이를 감안하여 구축하고자 하는 시스템을 분석하여야 한다. 각각의 접합 단계에서 $\frac{n+1}{4}L_{separator}$이라는 비용이 증가하는 것은 아니다. $\frac{n+1}{4}L_{separator}$는 모든 접합단계의 비용을 합하였을 때의 가능한 최대치이다. *파티션*이 여러 차례 진행되어, 다단계의 접합이 이루어지더라도, 결국 *분할선*과 교차하는 *light 구조*의 간선의 수는 총합 $(n-1)$을 넘지 않으며, *파티션* 단계별 $L_{separator}$의 길이는 지수적으로 감소/증가한다는 사실을 반영하여 비용증가를 계산하여야 한다. 다음 단계의 연구는 *오차정리* 및 관련기법을 기반으로 하여 구성될 수 있는 알고리즘들을 보다 정확히 분석할 수 있도록 하는 것이다. 각각의 알고리즘 또는 해당하는 응용이 가지는 기하학적인 특성을 반영하여, 비용 증가분이 더 줄어들 수 있는 여지를 찾아 분석하고 그 결과를 수식으로 정리함으로써, 네트워크 상호연결 문제에 대한 분산기반 병렬처리 접근방법 상의 성질을 지속적으로 파악할 것이다.

참 고 문 헌

[1] R. C. Prim, "Shortest connection networks and some generalizations," Bell System Technical Journal 36, pp. 1389-1401, 1957.

[2] Joseph. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," In Proceedings of the American Mathematical Society, Vol 7, No. 1, pp. 48 - 50, 1956.

[3] Du X., Wu W., Kelley D.F., "Approximations for subset interconnection designs," Theoretical Computer Science, Vol 207, No. 1, pp. 171-180, 1998.

[4] Fragopoulou P., Akl S.G., "Spanning subgraphs with applications to communication on the multidimensional torus network," Parallel Computing, Vol 22, No 7, pp. 991-1015, 1996.

[5] Robert Sedgewick and Kevin Wayne, "Minimum Spanning Tree lecture notes," Computer Science 226: Algorithms & Data Structures, Princeton University, Spring 2007.

[6] Euclidean Minimum Spanning Tree, http://en.wikipedia.org/wiki/Euclidean_minimum_spanning_tree

[7] A. Yao, "On constructing minimum spanning trees in k-dimensional spaces and related problems," SIAM Journal on Computing, 11(4), pp. 721-736, 1982.

[8] 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.

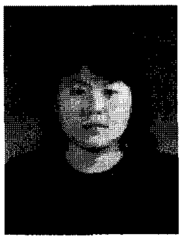
[9] S. Arora, "Nearly linear time approximation schemes for Euclidean TSP and other geometric problems", Proc. 38th IEEE Symp. on Foundations of Computer Science, pp. 554-563, 1997.

[10] 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.

[11] Delaunay Triangulation, http://en.wikipedia.org/wiki/Delaunay_triangulation

저 자 소 개

김 준 모(정회원)
1989년 서울대학교 컴퓨터공학과 학사 졸업.
2001년 University of Minnesota 전산학 박사 졸업.
2002년~2004년 한국정보보호 진흥원 연구원.
2004년~현재 단국대학교 전자컴퓨터공학부 조교수.
<주관심분야 : Approximations for NP-hard problems>



구 은 희(학생회원)
2002년 단국대학교 전자컴퓨터공학부 학사 졸업.
2004년 단국대학교 전자컴퓨터공학과 석사 졸업.
2006년~현재 단국대학교 전자컴퓨터공학과 박사과정.
<주관심분야 : 정보보호, 네트워크, 자바카드, 스마트카드 보안, 암호 알고리즘>