

비 균등 노드 분포환경에서 부분 PTAS를 이용한 효과적인 유클리드 최소신장트리 생성

김인범*, 김수인**

Efficient Construction of Euclidean Minimum Spanning Tree Using Partial Polynomial-Time Approximation Scheme in Unequality Node Distribution

In-bum Kim *, Soo-In Kim**

요 약

균등하게 분포된 많은 입력노드들을 최소비용으로 연결하는 최소신장트리 생성문제에 PTAS를 사용하면 실행 시간에 있어서 효과적으로 결과를 얻을 수 있다. 그러나 비 균등 분포의 경우에는 PTAS 적용이 오히려 성능을 저하시킬 수 있다. 본 논문에서는 특정 영역에 노드들이 밀집된 경우 해당 영역에만 PTAS를 적용한 부분 PTAS를 제안한다. 이 방법은 50,000개 입력노드들의 90%가 특정 영역에 밀집된 환경에서 기존의 PTAS 방식에 비해서 생성시간은 88.49%, 트리길이는 0.86% 감소를 보였고, Prim의 Naive 최소신장트리 생성 방법에 비해서 생성시간은 87.57% 감소, 트리길이는 1.18% 증가를 보였다. 따라서 본 연구의 제안방법은 많은 노드들이 특정영역에 밀집된 환경에서 이 노드들을 빠른 시간 내에 연결해야 하는 응용 등에 잘 적용될 수 있을 것이다.

▶ Keywords : Polynomial-Time Approximation Scheme, 최소신장 트리, Prim의 알고리즘, 비 균등 노드 분포

Abstract

Employing PTAS to building minimum spanning tree for a large number of equal distribution input terminal nodes can be a effective way in execution time. But applying PTAS to building

•제1저자 : 김인범 •교신저자 : 김인범

•투고일 : 2014. 5. 19, 심사일 : 2014. 6. 2, 게재확정일 : 2014. 6. 13.

* 김포대학교 인터넷정보과(Dept. of Internet Information, Kimpo College)

** 김포대학교 항공전자전자과(Dept. of Avionics, Kimpo College)

※ 본 논문은 2012년 한국멀티미디어학회 춘계학술발표대회에서 발표한 논문 "비 균등 노드 분포 환경에서의 효과적인 대규모 멀티미디어 통신네트워크 구축을 위한 PTAS"를 확장한 것임

※ 본 논문은 2014학년도 김포대학교의 연구비 지원에 의해 연구되었음

minimum spanning tree for tremendous unequal distribution node may lead to performance degradation. In this paper, a partial PTAS reflecting the scheme into specific node dense area is presented. In the environment where 90% of 50,000 input terminal nodes stand close together in specific area, approximate minimum spanning tree by our proposed scheme can show about 88.49% execution time less and 0.86% tree length less than by existing PTAS, and about 87.57% execution time less and 1.18% tree length more than by Prim's naive scheme. Therefore our scheme can go well to many useful applications where a multitude of nodes gathered around specific area should be connected efficiently as soon as possible.

▶ Keywords : Polynomial-Time Approximation Scheme, Minimum Spanning Tree, Prim's Algorithm, Unequality Node Distribution

1. 서론

다양한 스마트 디바이스의 보급 및 확산은 이를 지원할 수 있는 정적·동적 통신망의 효율적이고 빠른 구성을 요구한다. 다항 적(Polynomial) 문제 영역에서 통신망 구성의 최적의 방법은 최소신장트리(Minimum Spanning Tree)를 활용하는 것이다[1]. 그러나 입력단말 노드의 수가 매우 많을 경우, Prim의 알고리즘과 같은 기존의 순수 최소신장트리 생성방법은 효과적이지 않을 수 있다. 이 경우 비 다항적 NP-hard 문제를 해결하기 위해 고안되었던 PTAS(Polynomial-Time Approximation Scheme)를 활용하여, 문제영역을 균등하게 분할하여 각 부분 최소신장트리를 생성 한 후 이를 계층적으로 병합하여 처리하면, 수많은 입력노드를 연결하는 문제를 효과적으로 처리할 수 있다. 그러나 현실세계의 통신의 단말노드들의 분포는 균등하지 않은 것이 일반적이다. 표 1은 대한민국의 2012년 4사분기 산업별 경제활동인구에 관한 것이다. 출판, 영상, 방송통신, 정보서비스업은 전체 경제활동 인구의 약 78.1%가 수도권에 밀집되어있다. 이는 통신망을 구성하는 단말노드의 분포가 특정영역에 집중할 수 있음을 보여주고 있는 한 예가 될 수 있다. 교통 혼잡, 인구과밀 및 집중과 관련된 문제 등 이러한 사례는 여러 응용에서도 빈번하게 발생할 수 있다. 기존의 일반 PTAS 기법은 입력단말 노드의 분포가 거의 균등한 환경에서 좋은 성능을 보였지만, 다수의 입력단말노드들이 비 균등하게 분포되어있는 환경에서는 새로운 변형된 형태의 PTAS가 고안되어 적용되어야 할 것이

다.

본 논문에서는 다수의 입력단말노드들이 전체 문제 영역의 특정구역에 밀집되어 있는 환경에서 부분 PTAS 방법을 활용하여 신속하게 대형 근사 최소신장트리를 생성하는 방법을 제안한다. 이 방법은 대규모 스마트 기기의 보급 및 확산 등으로 통신망의 신속한 구성, 접속 및 서비스 제공 요청 등이 특정 구역에 밀집되어 있는 응용 등에 잘 활용될 수 있을 것이다. 또한 밀집된 공간과 인구와 관련된 응용에도 잘 적용될 수 있다.

본 논문 2장에서는 관련된 연구들의 내용, 3장에서는 본 논문에서 제안하는 비 균등 대규모 노드 분포 환경에서의 부분 PTAS를 이용한 근사 최소신장트리 생성방법, 4장에서 제안 방법의 실제 구현 및 실험분석 등을 기술하고, 5장에서 결론을 맺는다.

표 1. 산업별 경제활동인구(통계청, 2012년 4/4분기, 단위 천명)
Table 1. Industrial economically active population(National Statistical Office, 2012 fourth quarte, unit: thousand people)

산업	전국	수도권	비율
농업, 임업 및 어업	1,194	101	8.5%
광공업	4,197	1,907	45.4%
건설업	1,750	845	48.3%
도소매·음식숙박업	5,597	2,904	51.9%
전기·운수·금융	2,309	1,272	55.1%
출판, 영상, 방송통신, 정보서비스업	691	540	78.1%
사업·개인·공공서비스, 기타	8,663	4,684	54.1%
합계	24,402	12,256	50.2%

II. 관련 연구

본 논문에서 제안하는 방법은 2차원 평면상의 수많은 입력 단말노드들이 비 균등하게 분포되어 특정구역에 밀집된 환경에서 최소신장트리를 생성하여 이 노드들을 빠른 시간 내에 연결하는 방법을 찾기 위함이다. 주어진 입력노드와 입력 연결선이 있을 때 입력 연결선의 부분집합을 이용하여 입력노드들을 모두 연결하는 최소 길이의 트리가 최소신장트리이다. 다항 적 (Polynomial) 시간문제 환경에서 입력 노드들을 최소 비용으로 연결하는 최소신장트리 생성방법은 이미 많이 연구되어 발표되었는데, Kruskal과 Prim의 알고리즘이 대표적이다[1].

컴퓨터 분야에서 PTAS(Polynomial Time Approximation Scheme)는 대부분 NP-hard 최적화문제와 같은, 최적화 문제에 대한 근사 알고리즘이다[2]. PTAS는 최적화 문제의 한 인스턴스(Instance)와 파라미터 $\epsilon > 0$ 을 사용, 최적 해의 $(1 + \epsilon)$ 배의 해를 찾는다. 예를 들어 Euclidean Traveling Salesman Problem에 대해서 PTAS는, L 은 최적 여행경로의 길이라 가정할 때, 길이가 최대 $(1 + \epsilon) \times L$ 인 경로를 찾는다. PTAS의 실행시간은 모든 고정된 n 의 다항식이 요구되지만 다른 ϵ 에 대해서는 다를 수 있다. 따라서 실행시간이 $O(n^{1/\epsilon})$ 이거나 $O(n^{\exp(1/\epsilon)})$ 인 알고리즘도 PTAS가 될 수 있다.

PTAS 관련 연구들은 다양한 분야에서 수행되었다. 고속도로 문제(Highway Problem)에 PTAS를 적용한 연구가 있다[3]. 이 문제에서는 n -edge line graph (highway)와 각각의 budget을 갖는 경로(drivers)의 집합이 주어진다. 각각의 edge에는 가중치(tolls)가 주어지고, 고속도로 소유자는 운전자의 budget을 초과하지 않는 범위에서 운전자로부터 경로 사용비용을 거둬들인다고 가정할 때, 최대의 수익을 얻을 수 있는 가중치를 선택하는 것이다. 이 연구에서 무작위적 절단 기법(Randomized Dissection Approach)을 기반으로 고속도로 문제에 대한 PTAS를 제시하였다. 이 연구에서 고속도로를 bounding path 안에 둘러싸고, bounding path의 크기와 고속도로 위치는 랜덤 변수로 설정한 후, 동일한 최적 가중치(Uniform Optimal Weight)의 sub-path 안에서, bounding path의 재귀적 $O(1)$ -ary dissection을 고려하였다. 최적의 가중치는 알려져 있지 않기 때문에, 근사해를 계산하면서 동시에 동적 프로그래밍을 통해 상향식으로 dissection을 구축하였다. 또한 이 연구의 결과를 활용, 고속도로 요금소 문제(Tollbooth Problem)와 간격 매트릭스

(Interval Matrices) 상에서 최대 실행 서브시스템(Maximum Feasibility Subsystem)에 대한 PTAS도 제공할 수 있다고 주장하였다. 병렬 배치 스케줄링 모델에 PTAS를 적용한 연구가 있다[4]. 이 모델에서, 배치 처리 시간이 배치 멤버 중에서 가장 큰 처리시간인 경우, job 그룹은 배치에서 함께 스케줄링 될 수 있다. rejection을 허용하는 스케줄링 모델에서는 어떠한 job도 목적 값에 부가되는 적절한 penalty cost와 함께 reject될 수 있다. 이 연구에서는 dynamic job arrival과 reject 허용 스케줄링 모델을 결합한 PTAS를 제시하였다. 이 PTAS는 dynamic programming과 rounding 기법을 사용하였으며 접수된 job의 총소요시간(Makespan)과 rejection된 job들의 전체 penalty를 최소화하였다. 또한 최소-가중치 커버 집합(Minimum-Weight Set Cover)에 대한 PTAS 적용 연구가 있다[5]. 주어진 평면상의 객체집합의 최소-가중치 부분집합으로, 주어진 점들의 집합을 커버해야 하는 최소-가중치 커버 집합(Minimum-Weight Set Cover)을 연구하여 평면 객체에서 기하학적 가중치 커버 집합(Weighted Geometric Set Cover)에 대한 PTAS를 제안하였다. 이 연구에서는 단위 정사각형의 평면상에서 알고리즘을 확장하여 단위 정사각형의 교차그래프에서 최소 가중치 지배집합(Minimum Weight Dominating Set)에 대한 PTAS와 단위 정사각형의 기하학적 예산 최대 커버리지(Geometric Budgeted Maximum Coverage)에 대한 PTAS를 제시하였다. 3차원 무선 센서네트워크에 활용하기 위한 PTAS에 대한 연구가 있다[6]. 동종의 센서들이 평면이 아닌 3차원 공간에 배치되었을 때, 센서네트워크의 수학적 모델은 단위 디스크 그래프(Unit Disk Graph)가 아닌 단위 볼 그래프(Unit Ball Graph)이다. 단위 디스크 그래프에서 최소 연결 지배 집합(Minimum Connected Dominating Set)에 대한 PTAS는 존재하지만 이 구조를 단위 볼 그래프에 대한 PTAS로 확장시킬 수 없다. 이 연구에서는 단위 볼 그래프에서의 최소 연결 지배 집합 문제를 위한 PTAS를 제시하고 단위 디스크 그래프에서의 PTAS 실행시간을 개선한 새로운 구조를 제안하였다. 또한 주어진 노드들을 상호 연결하기 위한 PTAS에 관한 연구가 있다[7]. 이것은 ad-hoc 네트워크 또는 센서 네트워크상에서, 중간노드들을 추가 배치시키는 형태의 입력 노드들을 상호 연결하는 문제를 해결하려 하였다. 네트워크의 노드들은 응용시스템 또는 지형적인 요인에 의해 일부지역에서는 밀집하여 분포되고, 그 외의 지역에서는 희박하게 분포될 수 있을 때, 노드들이 밀집한 지역의 상호연결을 무시함으로써, 짧은 실행시간 안에 추가노드들의 최적배치에 근접하

도록 하는 방법을 동적프로그래밍으로 구현하는 PTAS를 제안하였다. 이 연구에서는 목적함수를 노드 밀집지역을 추상화 시킴에 의해 발생하게 되는 문제점에 대처할 수 있도록 적절하게 정의함으로 실행결과와 정확한 분석이 가능하다고 주장하였다. GOSST(Grade of Services Steiner Tree) 생성 문제에 PTAS를 활용한 연구가 있다[8]. GOSST 생성 문제는 NP-Complete 영역에 속하므로, 이를 위한 휴리스틱들은, 다수의 입력에 대해서 막대한 계산량이 요구된다. 이 연구에서는 weight를 가지는 많은 입력 노드들을 대상으로, 공간 지역성을 반영한 PTAS를 개발하여 이를 GOSST 생성에 효과적으로 적용하는 방법을 제안하였다.

III. 비 균등 노드 분포 환경에서 근사 최소신장트리 생성을 위한 부분 PTAS

PTAS 기법은 문제영역이 매우 크거나 복잡할 경우 이것을 여러 개의 부분문제로 나누어 처리하는 일종의 분할정복(divide and conquer) 방법이다. 연결해야 할 노드가 매우 많은 환경에서 전체를 대상으로 Prim의 알고리즘을 활용하여 최적의 최소신장트리를 생성할 경우 막대한 실행시간을 필요로 한다. 노드의 수가 많을 경우, 인접하게 연결할 노드들은 지역성(Locality)을 보이게 된다. 따라서 전체 문제영역을 몇 개의 구역으로 나누어 처리한 후 이를 계층적으로 통합해서 근사 최소 신장트리를 생성해도 그 오차는 크지 않을 것으로 예상된다. 그림 1은 입력단말노드들이 위치한 전체 문제영역을 16개의 부분 구역으로 분할한 최하위 계층을 나타낸다. 각 부분 구역에는 그림 2에 표시된 포털(Portal)이라는 가상 노드들이 경계선에 위치하게 되는데, 이 포털들은 각 구역에서 부분 최소신장트리인 Local MST를 생성할 때, 입력단말 노드들과 함께 사용된다. 또한 포털을 통해서 두 구역의 부분 최소신장트리인 Local MST를 연결하게 되는데, 그림 2와 같이 연결 길이가 작은 Link₁과 Link₂가 선택되고, 이 과정에서 인접한 두 개의 부분 구역의 최소신장트리들이 병합되어 상위 계층의 구역의 최소신장트리를 형성하게 된다. 두 개의 포털을 통해서 최하위 계층의 상하로 인접한 두 구역, 즉 P₁과 P₈이 연결되고, 이것은 최종단계에서의 최적화 단계에서 Link₅로 연결되게 된다. 그림 3에는 이러한 과정을 거쳐 병합된 8개의 구역이 나타나 있고, 이 구역들도 다시 좌우로 인접한 부분영역들이 병합되어 그림 4와 같은 구역으로 병합된다. 그림 5는 그림 4의 4개의 부분구역이 병합되어 생성된 모습이고, 그림 6은 최종 병합된 문제영역을 나타낸다. 각 병합

과정에서는 인접한 영역의 경계선에 사용될 포털들을 미리 정의하여 그림 1의 최하위 영역에서 부분최소신장트리를 생성할 시기에 이 포털도 입력단말 노드들과 함께 참여하여 생성하게 되고 각 병합과정에서 이 포털을 이용하여 상하 또는 좌우 인접한 두 구역의 연결과정에서 활용한다.

그림 7은 500개의 임의로 생성된 균등분포의 입력단말노드의 위치를 나타낸다. 이 입력 환경에 대하여 그림 8에는 일반 PTAS를 이용하여 생성된 근사최소신장트리의 연결모습이 나타나 있다. 그림 9는 Prim의 알고리즘을 이용한 최적의 최소신장트리의 모습이다.

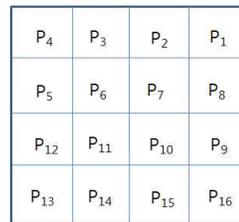


그림 1 최하위(1차) 계층의 16 구역 분할
Fig. 1 Sixteen partitioning of bottom(1st) layer

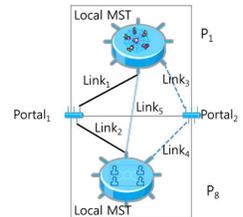


그림 2 portal을 통한 인접 영역의 연결
Fig. 2 Connecting adjacent two partitions by portals

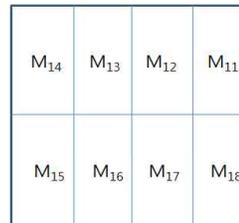


그림 3 1차 계층의 병합으로 생성된 2차 계층
Fig. 3 2nd layer by merging 1st layer

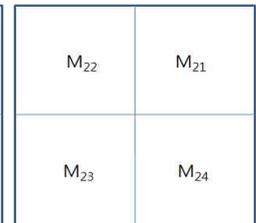


그림 4 2차 계층의 병합으로 생성된 3차 계층
Fig. 4 3rd layer by merging 2nd layer

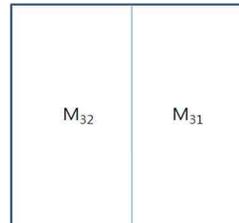


그림 5 3차 계층의 병합으로 생성된 4차 계층
Fig. 5 4th layer by merging 3rd layer



그림 6 4차 계층의 병합으로 생성된 최상위(5차) 계층
Fig. 6 Top(5th) layer by merging 4th layer

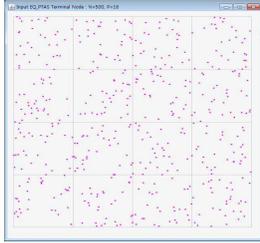


그림 7 균등분포의 500개의 입력단말노드
Fig. 7 Equality distribution 500 input terminal nodes

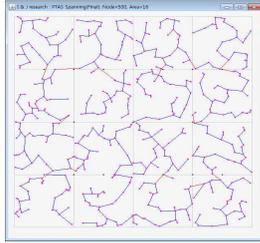


그림 8 일반 PTAS를 사용한 근사 최소신장트리
Fig. 8 Approximate minimum spanning tree by general PTAS for fig 7

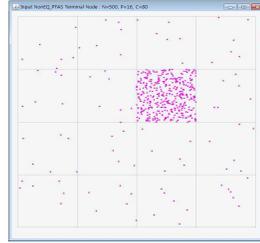


그림 10 비 균등분포의 500개의 입력단말노드
Fig. 10 Unequality distribution 500 input terminal nodes

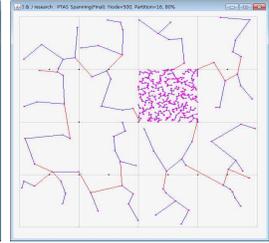


그림 11 일반 PTAS를 사용한 근사 최소신장트리
Fig. 11 Approximate minimum spanning tree by general PTAS for fig.10

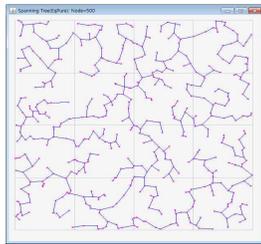


그림 9 Prim의 최적 최소신장트리
Fig. 9 Prim's optimal minimum spanning tree for fig 7

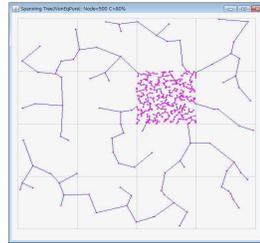


그림 12 Prim의 최적 최소신장트리
Fig. 12 Prim's optimal minimum spanning tree for fig 10

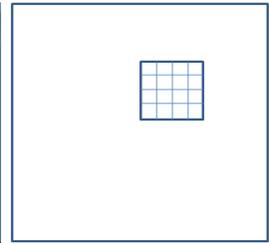


그림 13 밀집구역의 선택 및 분할
Fig. 13 Locating and partitioning of dense node area

그러나 그림 10과 같이 입력단말노드의 분포가 특정 구역에 밀집된 경우에는 상황이 다르다. 그림 11은 밀집된 구역을 기존의 일반 PTAS 방식으로 처리 생성했을 때의 모습이다. 그림 12는 Prim의 방법으로 최적의 최소신장트리를 생성한 모습이다. 이러한 Prim의 방법은 입력단말노드의 분포가 균등하거나, 균등하지 않거나 큰 차이가 없다. 그러나 기존의 일반 PTAS방식은 노드 분포가 균등했을 경우, 실행시간에서 우수한 성능을 보이지만, 특정영역에 밀집되어 있는 환경에서는 그 실행시간과 트리길이 측면에서 성능이 급격하게 떨어지게 된다. 따라서 이러한 경우 그림 13과 같이 입력노드들이 밀집되어 있는 구역에 대해서만 PTAS를 적용하는 부분 PTAS를 본 논문에서는 제안한다. 즉, 그림 10의 입력노드환경에서 구역 P_7 에 입력터미널노드들이 밀집되어 있는 경우에, P_7 을 16개의 세부구역으로 나누어 PTAS를 적용한다.

표 2. 부분 PTAS 근사 최소신장트리 생성 알고리즘
Table 2. Algorithm for Partial PTAS Approximate Minimum Spanning Tree

	Process
1	<pre> mst_congestedArea=NULL; for(i=0; i<(partitionNum; i++){ mst_EachPartition(i)=buildMst_EachPartition(nodes_partition(i)) mst_congestedArea += mst_EachPartition(i); } </pre>
2	<pre> 4nodes = selectRepNode(node_partition(0..partitionNum)) </pre>
3	<pre> nodes_SparseAdeed = nodes_SparseArea+4nodes; mst_SparseArea=buildMst_SparsesArea(nodes_SparseAdeed) </pre>
4	<pre> mergedTree=merge(mst_congestedArea, mst_SparseArea); </pre>
5	<pre> finalTree=cuttingUselessConnection(mergedTree); </pre>

수많은 입력노드들의 분포가 비 균등하게 분포된 환경에서 부분 PTAS를 이용하여 근사 최소신장트리를 신속하게 생성하는 알고리즘이 표 2에 있다. 단계 1에서 우선 입력노드들이 밀집해 있는 구역에 대하여 16개의 영역으로 최하위 계층으로 분리하고 각각의 영역에서 미리 정의된 포털을 이용하여

최소신장트리를 생성한다. 이 과정은 그림 14부터 그림 18에 순차적으로 나타나있다. 그림 14는 밀집구역이 그림 1의 P_7 에 해당하는 구역으로 이것을 16개의 서브구역으로 나누고 이것의 첫 번째 영역인 P_{71} 에 위치한 입력단말노드들의 최소신장트리를 나타낸다. 구역 P_{71} 의 최소신장트리와 아래 인접한 구역의 최소신장트리를 포털을 이용해서 병합한 상위계층의 최소신장트리가 그림 15에 있다. 그림 16, 그림 17, 그림 18은 같은 방법으로 하위 계층의 트리를 병합하여 생성된 상위 계층 트리의 한 예를 보이고 있다. 그림 18의 트리는 노드 밀집 구역에서의 근사 최소신장트리로, 하위 계층의 병합을 위해 사용되었던 포털들을 경유하는 불필요한 연결을 제거하고, 포털과 연결된 입력단말노드들을 바로 연결하는 후처리 작업이 시행된 결과이다. 단계 2에서는 노드 밀집구역을 대표하는 4개의 노드들을 선정하는데, 본 논문에서는 해당 영역의 상하좌우에 위치한 노드들을 선정하였다. 단계 3에서는 그림 19와 같이 밀집구역 이외의 영역에 위치한 노드들과 단계 2에서 선정된 4개의 노드들을 입력으로 최소신장트리를 생성한다. 단계 4에서는 단계 1과 단계 3에서 생성된 근사최소신장트리를 단계 3에서 선정된 4개의 노드들 이용하여 병합하게 된다. 단계 5에서는 존재하는 불필요 연결들을 제거한다. 예를 들면 단계 3에서 생성된 트리과 단계 4에서 선정된 4개 노드들의 연결들 중에서 최단의 한 개의 연결을 선택하고 나머지는 제거한다. 그림 20은 500개의 불균등 분포의 입력단말노드 환경에서 부분 PTAS를 활용한 최종 근사 최소신장트리의 모습을 나타낸다.

기존의 유클리드 최소신장트리 생성을 위한 완전연결을 위한 연결 수는 입력노드 수 N 에 대하여 $O(N^2)$ 이다. 이것을 입력연결로 하여 기존의 Prim의 알고리즘을 적용하면 최소신장트리를 생성하는 실행시간은 $O(N^3)$ 이다. 그러나 들로네 삼각망을 이용하면 완전연결을 하지 않고도 유클리드최소신장트리를 생성할 수 있다. 이 경우 유클리드최소신장트리 생성을 위한 시간은 $O(N \log N)$ 이다. 이것들을 PTAS에 적용했을 때, ρ 개의 구역으로 구분하면, 각 구역의 최소신장트리를 생성하는 시간은 $O(\eta \log \eta)$ 이다. 여기서 입력노드가 균등하게 분포되어있을 경우, $\eta \approx N/\rho$ 이고 비 균등하게 분포되었을 경우는 $\eta \approx N$ 이다. 따라서 비 균등 분포인 경우에 일반적인 PTAS는 일반 Prim의 알고리즘을 이용하는 것과 시간은 동일하지만, 트리의 길이에서는 오히려 오버헤드가 발생하는 현상이 나타난다. 따라서 특정구역에 입력단말노드들이 밀집된 구역을 찾고, 이 구역에 대해서만 일반 PTAS의 장점인 입력노드들을 균등하게 처리하는 방법을 적용한다면, $\eta \approx N/\rho$ 인 빠른 속도의 근사최소신장트리를 생성할 수 있다. 본 논문

에서 제안하는 방법의 실행시간은 들로네삼각망을 적용하였을 경우, 입력노드 수 N 에 대하여 $O(N \log N)$ 이다.

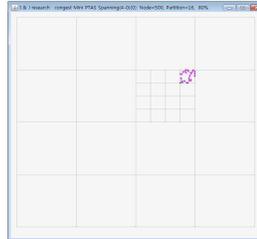


그림 14 밀집 구역의 최하위(1차) 계층에서의 근사 최소신장트리 예
Fig. 14 An approximate minimum spanning tree on bottom(1st) layer of dense node area

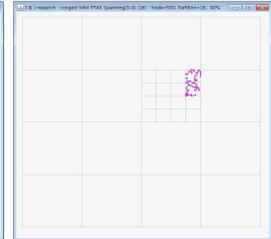


그림 15 밀집 구역의 1차 계층 병합으로 생성된 2차 계층 근사 최소신장트리 예
Fig. 15 An approximate minimum spanning tree on 2nd layer by merging 1st layer of dense node area

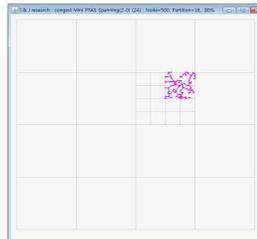


그림 16 밀집 구역의 2차 계층 병합으로 생성된 3차 계층 근사 최소신장트리 예
Fig. 16 An approximate minimum spanning tree on 3rd layer by merging 2nd layer of dense node area



그림 17 밀집 구역의 3차 계층 병합으로 생성된 4차 계층 근사 최소신장트리 예
Fig. 17 An approximate minimum spanning tree on 4th layer by merging 3rd layer of dense node area

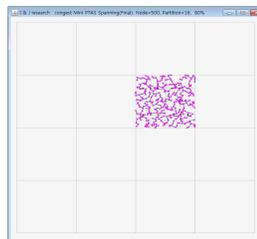


그림 18 밀집 구역의 4차 계층 병합으로 생성된 최상위 계층 근사 최소신장트리
Fig. 18 An approximate minimum spanning tree on Top(5th) layer by merging 4th layer of dense node area

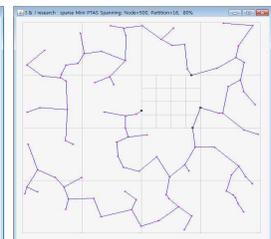


그림 19 비 밀집 구역의 노드와 밀집 대표 노드들에 대한 최소신장트리
Fig. 19 Minimum spanning tree for sparse area nodes and dense node representatives

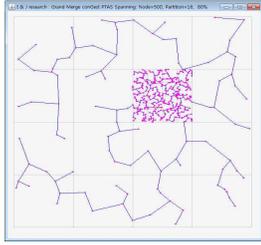


그림 20 최종 생성된 근사 최소신장트리
Fig. 20 Final approximate minimum spanning tree

IV. 실험 및 분석

1. 실험방법

본 논문에서 제안된 방법의 다수의 비 균등 노드분포 환경에서의 효율성을 보이기 위해 입력 노드 50,000개에 대해 입력 노드의 집중도를 변화시켜 실행시간과 생성 트리 구성선분의 길이를 조사하였다. 각기 다른 입력노드들의 집중분포에 대해 제안된 방법의 비교대상은 같은 입력노드환경에 대한 Prim의 순수 최소신장트리와 균일분포 일반 PTAS 근사 최소신장이다.

본 논문에서 제안된 방법과 이를 비교하기 위한 방법들은 모두 8 기가바이트 램, Intel 프로세서, Microsoft 윈도우즈 7.0 환경의 랩톱 컴퓨터에서 Java로 직접 구현하여 비교 분석되었다. 본 논문의 실험에 사용된 50,000개의 2차원 입력 노드들의 가상적인 문제 영역을 16개로 구분하였고, 이 중 그림 1의 P₇ 구역에, 전체 노드 수의 10%, 30%, 50%, 70%, 그리고 90%의 노드들이 분포하게 하였다. 이러한 비 균등 분포의 입력노드들을 랜덤하게 2차원 노드 좌표를 생성하여 사용하였다.

2. 실험 결과 및 분석

첫 번째 실험은 노드 집중도의 변화에 따른 트리들의 생성 시간에 관한 것이다. 밀집구역에 집중된 노드의 수를 전체 노드 수 50,000개의 10%, 30%, 50%, 70%, 그리고 90%로 변화시키면서 입력노드를 임의로 생성한 후, 각각의 입력노드들에 대하여 트리를 생성하는 시간을 측정하였다. 이 실험을 통해 본 논문에서 제안하는 방법이, 특수한 상황이지만 현실적으로 빈번할 수 있는 특정 영역에 연결해야 할 노드가 집중된 환경에서 기존의 방법들에 비해 실행시간을 획기적으로 개선할 수 있음은 물론 기존의 균등분포에서의 실행시간을 개선

한 일반 PTAS 방법에 비해서도 생성시간과 트리길이를 효과적으로 감소시킬 수 있음을 나타내었다. 실험 결과는 밀집구역에 10%의 입력노드가 집중된 환경에서, 본 논문에서 제안한 방법인 부분 PTAS(conPTAS Span)가 가장 성능이 안 좋은 47,065의 시간이 소요되었고, 균등 노드분포에 적합한 일반 PTAS 방법 genPtas Span이 가장 좋은 5,523을 보였다. 그러나 밀집노드가 50% 집중된 환경부터는 상황이 역전되었으며, 70%, 90%의 경우에는 약 66.9%, 88.49%의 시간 절감을 보였다. 순수 Prim의 알고리즘을 이용해서 최적 최소신장트리를 생성하는 naive Span 방법에 비해서 밀집노드 10% 집중 환경에서는 본 논문에서 제안하는 방법이 거의 비슷했으나, 30% 집중 환경부터는 실행시간이 개선되기 시작하여, 70%, 90%의 노드집중 환경에서는 78.9%, 87.6%의 생성시간의 감소를 보였다.

이것은 본 논문에서 제안하는 방법이 밀집노드들이 특정구역에 작은 집중도를 보이는 환경에서는 균등 PTAS 방법에 비해 실행시간의 장점이 보이지 않고, 순수 최소신장트리 생성방법과 유사하지만 밀집노드의 집중도가 높아지면 높을수록 실행시간의 단축비율이 높아짐을 확인할 수 있었다. 참고로 일반 PTAS의 경우에는 순수 최소신장트리와 비교해서 밀집노드의 특정구역으로의 집중도가 작은 환경, 즉 균등한 노드 분포에서는 시간단축의 장점을 보이지만 집중도가 높아지면 높아질수록 오히려 시간능력이 저하됨을 확인할 수 있다.

표 3 집중도 변화에 따른 실행시간 및 비교
Table 3 Execution time and comparison by changing node congestion

congestion Tree	10%	30%	50%	70%	90%
conPTAS Span	47065	32012	15397	10140	6038
genPTAS Span	5523	8221	17175	30639	52463
naive Span	45973	48469	47736	48095	48563
conPTAS vs naive	2.38%	-3.9530%	-67.75%	-78.92%	-87.57%
genPTAS vs naive	-87.99%	-83.04%	-64.02%	-36.29%	8.03%

표 4 부분 PTAS와 일반 PTAS의 실행시간 비교
Table 4 Execution time comparison between partial PTAS (conPTAS) and general PTAS(genPTAS)

congestion Tree	10%	30%	50%	70%	90%
conPTAS Span	47065	32012	15397	10140	6038
genPTAS Span	5523	8221	17175	30639	52463
conPTAS vs genPTAS	752.16%	289.39%	-10.35%	-66.90%	-88.49%

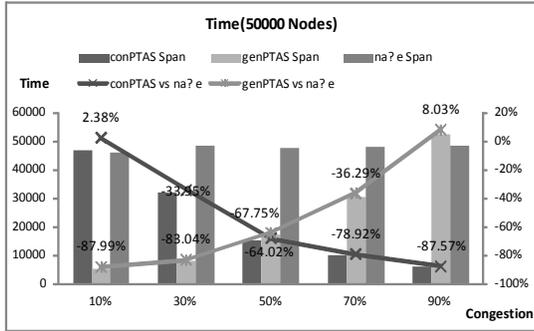


그림 21 집중도 변화에 따른 실행시간
Fig. 21 Execution time by changing node congestion

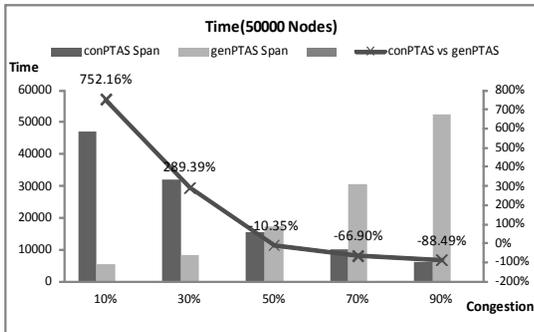


그림 22 집중도 변화에 따른 부분 PTAS와 일반 PTAS의 실행시간 비교
Fig. 22 Execution time comparison between partial PTAS and general PTAS by changing node congestion

두 번째 실험은 밀집노드의 집중도의 변화에 따른 생성 트리 길이에 관한 것이다. 밀집노드의 수를 전체 노드 수 50,000개의 10%, 30%, 50%, 70%, 그리고 90%로 변화시키면서 입력노드를 임의로 생성한 후, 각각의 입력노드들에 대하여 각각의 방법에 따라 트리를 생성한 후 길이를 측정하였다. 실험 결과는 제안된 부분 PTAS인 conPTAS Span은 근사 해이므로 최적 해인 Prim의 순수 최소신장트리 생성방법에 비해 밀집노드의 모든 집중도 환경에서 길이의 증가를 나타내어, 집중도 10%에서 0.54% 길이 증가를 보였고, 집중도의 증가됨에 따라 길이의 증가가 조금씩 늘어나 90% 집중도에서는 1.18%의 증가를 보였다. 그러나 일반 PTAS인 genPTAS에 비교해서는 모든 분포도에서 트리 길이의 절감을 보였는데, 10% 집중도에서 0.24%의 길이 절감을 보였고, 노드 집중도가 증가함에 따라 트리 길이의 절감비율이 높아져 90% 집중도에서는 0.86%의 트리 길이 절감을 보였다. 이것은 본 논문에서 제안하는 방법이 비밀집 구역에 속한 노드들의 연결은 대부분 연결 길이가 길고, 이것들의 연결은 최적 해인 Prim의 알고리즘을 사용하였기 때문에, 길이 증감이

일반 PTAS인 genPTAS 방법에 비해 적은 것으로 분석된다.

두 실험에서 본 논문에서 제안하는 방법은 밀집노드들의 작은 집중도를 보이는 환경에서는 일반 PTAS 방법에 비해 실행시간의 장점이 보이지 않지만, 트리의 길이가 약간 감소하는 장점을 보였으며, 노드 집중도가 커질수록 일반 PTAS에 비해 많은 시간절감과 트리길이 절감을 보였다. 또한 최적 해인 순수 최소신장트리방법에 비해서, 모든 집중도에서 생성된 트리의 길이는 약간 증가하지만, 실행시간 측면에서는 밀집노드의 노드 집중도가 작은 환경에서는 거의 비슷한 시간이 소요되고, 높은 집중도에서는 대폭적인 시간절감을 보였다. 따라서 본 논문에서 제안하는 방법은 밀집노드의 분포가 특정 구역에 집중된 환경에서 좋은 성능을 보일 것으로 분석된다.

표 5 집중도 변화에 따른 트리 길이 및 비교
Table 5 Tree length and comparison by changing node congestion

Tree \ congestion	10%	30%	50%	70%	90%
conPTAS Span	1747.34	1656.93	1510.2	1295.7	948.5
genPTAS Span	1751.56	1662	1515.83	1302.42	956.75
naive Span	1737.94	1647.43	1500.06	1284.92	937.44
conPTASvsnaive	0.54%	0.58%	0.68%	0.84%	1.18%
genPTAS vs naive	0.78%	0.88%	1.05%	1.36%	2.06%

표 6 부분 PTAS와 일반 PTAS 방법의 트리 길이 비교
Table 6 Tree length comparison between partial PTAS (conPTAS) and general PTAS(genPTAS)

Tree \ congestion	10%	30%	50%	70%	90%
conPTAS Span	1747.34	1656.93	1510.2	1295.7	948.5
genPTAS Span	1751.56	1662	1515.83	1302.42	956.75
conPTAS vs genPTAS	-0.24%	-0.31%	-0.37%	-0.52%	-0.86%

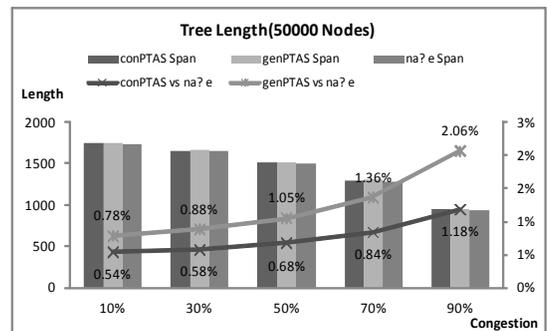


그림 23 집중도 변화에 따른 생성 트리 길이
Fig. 23 Tree length by changing node congestion

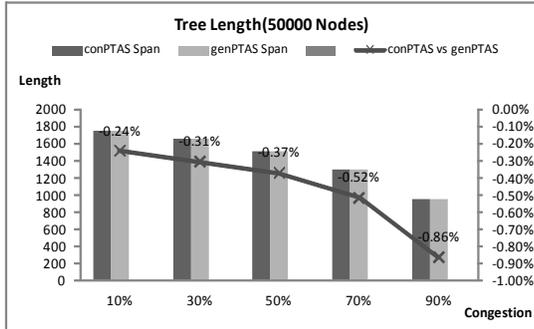


그림 24 집중도 변화에 따른 부분 PTAS와 일반 PTAS의 트리 길이 비교
 Fig. 24 Tree length comparison between partial PTAS and general PTAS by changing node congestion

V. 결론

본 논문에서는 입력노드들의 분포가 균등하지 않고, 어느 특정 영역에 집중되어있는 비 균등 노드 분포 환경에서 부분 PTAS기법을 활용하여 효과적으로 근사 최소신장트리를 생성하는 방법을 제안하였다. 제안된 방법은 50,000개의 입력노드들의 90%가 특정 구역에 집중되어 있는 환경에서 Prim의 순수 최소신장트리생성 방법에 비해 87.57%, 균등분포에서 좋은 성능을 보이는 일반 PTAS 방법에 비해 88.49%의 실행속도의 감소를 보였다. 또한 트리의 길이에서도 같은 환경에서 최적의 순수 최소신장트리에 비해 1.18% 증가했지만 일반 PTAS 방법 보다는 0.86%의 트리 길이 감소를 보였다. 따라서 제안된 방법은 입력노드들이 불균등하게 분포된 환경에서 빠른 시간 내에 최소신장트리를 생성해야 하는 응용에 잘 적용될 수 있을 것으로 판단된다.

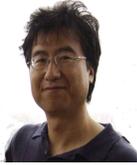
향 후 연구는, 비 균등 노드분포 환경에서 비교적 빠른 속도로 트리의 길이 감소에 관한 것이다. 근사 스타이너트리를 활용하는 등의 연구를 통해 본 논문에서 제안하는 방법보다 개선된 방법을 제시할 수 있다. 또한 특정영역의 노드 밀집 여부를 판단하고 이를 찾을 수 있는 방법의 연구이다. 이것은 들로네 삼각망을 활용하여 영역의 밀도를 계산하여 찾을 수 있을 것이다. 이 연구들을 통해 좀 더 실용적이며 다양한 응용에 본 논문의 제안내용을 적용할 수 있을 것이다.

참고문헌

[1] T.H. Cormen, C.E Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Ed., The

MIT Press, pp.561-579, 2001
 [2] PTAS(Polynomial Time Approximation Scheme), http://en.wikipedia.org/wiki/Polynomial-time_approximation_scheme, May, 2014.
 [3] F.Grandoni and T.RothvoB, "Pricing on paths: a PTAS for the highway problem," SODA '11 Proceedings of the twenty-second annual symposium on Discrete Algorithms, pp. 675-684, 2011
 [4] Z. Cao and X. Yang, "A PTAS for parallel batch scheduling with rejection and dynamic job arrivals", Journal of Theoretical Computer Science archive, Vol.410, No.27-29, pp. 2732-2745. June, 2009
 [5] T. Erlebach and E. Leeuwen, "PTAS for weighted set cover on unit squares," APPROX/RANDOM'10 Proceedings of the 13th international conference on Approximation, and 14 the International conference on Randomization, and combinatorial optimization: algorithms and techniques, pp.166-177, 2010
 [6] Z.Zhang, X.Gao, W.Wu and D.Du, "A PTAS for minimum connected dominating set in 3-dimensional Wireless sensor networks," Journal of Global Optimization archive, Vol.45 No.3, pp.451 - 458, November 2009
 [7] J. Kim, "Interconnection Problem among the Dense Areas of Nodes in Sensor Networks," Journal of the Institute of Electronics Engineers of Korea TC, Vol.48, No.2, pp.6-13, 2011
 [8] I. Kim, "Efficient Construction of Large Scale Grade of Services Steiner Tree Using Space Locality and Polynomial-Time Approximation Scheme," Journal of the Korea Society of Computer and Information, Vol.16, No.11, pp.153-161, 2011

저 자 소 개



김 인 범

1989: 서울대학교
컴퓨터공학과 공학사
1991: 서울대학교
컴퓨터공학과 공학석사
2007: 위스콘신주립대학-밀워키
컴퓨터과학과 공학박사
현 재: 김포대학교
인터넷정보과 교수
관심분야: 네트워크, 데이터베이스
Email : ibkim@kimpo.ac.kr



김 수 인

1984: 광운대학교
전자공학과 공학사
1991: 광운대학교
전자계산기공학과 공학석사
2004: 광운대학교
컴퓨터공학과 공학박사
현 재: 김포대학교
항공전기전자과 부교수
관심분야: 영상인식, 스테레오비전 등
Email : sikim@kimpo.ac.kr