Journal of The Korea Society of Computer and Information
Vol. 17 No. 7 , July 2012

2012-17-7-3-2

PTAS를 이용한 3차원 스타이너 최소트리의 신속한 구성

김 인 범*

Fast Construction of Three Dimensional Steiner Minimum Tree Using PTAS

Inbum Kim*

요 약

본 논문에서는 3차원 공간상에 존재하는 많은 입력노드를 신속하게 연결하는 PTAS 3차원 스타이너 최소트리를 제안한다. 스타이너 최소트리문제는 비 다항 적 문제 영역에 속하며 적절한 휴리스틱을 도입했을 경우 다항 적 문제 영역에서 최단 길이의 해를 생성하는 최소신장트리 방법과 같은 여러 방법에 비해 우수한 성능을 보이나, 입력 노드의 수가 클 경우 과도한 실행시간을 요구한다. 본 논문에서는 이 문제를 해결하기 위해 PTAS 기법을 도입한 방법을 제안한다. 3차원 공간상에 존재하는 70,000개의 입력 노드에 대한 실험에서, 본 논문에서 제안된 8개 공간 분할 PTAS 방법은, 순수 3차원 스타이너 최소트리방법에 비해 연결 길이는 0.81% 증가했으나, 실행시간은 86.88%의 단축되었다. 이는 제안된 방법이 시간적 제약이 비교적 큰 문제에서 공간상의 많은 노드들을 신속하게 연결하는 응용에 잘 적용될 수 있음을 나타낸다.

▸ Keyword : 다항 적 시간 근사 법, 스타이너 최소트리, 최소신장트리, 3차원 공간, 실행시간

Abstract

In this paper, PTAS three-dimensional Steiner minimum tree connecting numerous input nodes rapidly in 3D space is proposed. Steiner minimum tree problem belongs to NP problem domain, and when properly devised heuristic introduces, it is generally superior to other algorithms as minimum spanning tree affiliated with P problem domain. But when the number of input nodes is very large, the problem requires excessive execution time. In this paper, a method using PTAS is proposed to solve the difficulty. In experiments for 70,000 input nodes in 3D space, the tree

• 제1저자 : 김인범 • 교신저자 : 김인범
• 투고일 : 2012. 04. 10, 심사일 : 2012. 05. 02, 게재확정일 : 2012. 06. 12.
* 김포대학교 인터넷정보과(Dept. of Internet Information, Kimpo College)
※ 본 논문은 2012학년도 김포대학교의 연구비 지원에 의해 연구되었음

produced by the proposed 8 space partitioned PTAS method reduced 86.88% execution time, compared with the tree by naive 3D steiner minimum tree method, though increased 0.81% tree length. This affirms the proposed method can work well for applications that many nodes of three dimensions are need to connect swifty, enduring slight increase of tree length.

▸ Keyword : PTAS, Steiner Minimum Tree, Minimum Spanning Tree, 3-Dimensional Space, Execution Time

I. 서 론

3차원 공간의 구성요소는 길이, 너비, 깊이 또는 높이이다. 3차원 공간에서 특정 노드의 위치는 서로 직각인 x, y, z 좌표축에서의 이격거리를 의미하는 실수 값의 순서쌍으로 표현된다. 3차원 공간 정보는 2차원 평면 정보와는 달리 현실세계의 정보를 좀 더 정확하게 표현한다. IT 기술의 급속한 발전은 3차원 정보의 효과적 수집 및 처리가 가능해지고 이를 통한 다양한 분야의 서비스 제공이 가능해졌다. 또한 교통, 통신 및 관련 기술의 지속적인 발전 및 확대에 따라 과거 2차원 평면 기반의 연구 결과 적용의 한계가 드러나고 이에 따른 대폭적인 개선이 요구되고 있다.

컴퓨터 분야에서 트리(Tree)는 정보를 포함하는 노드와 이들 사이의 연관 관계를 표현하는 링크로 구성된 비선형 자료 구조로서 다양하게 활용된다. 트리의 효율적이고 신속한 구성은 유,무선 네트워크의 라우팅 및 QoS, 전자회로 및 전력선의 설계, 회로설계, 도로, 철로 및 선박이나 항공기의 항로 개발, 의학 및 생화학, 생산 공정 등 수많은 분야에서 유용하게 적용될 수 있다. 특히 통신 네트워크와 IC 설계분야에서 효율적이고 신속한 3차원 트리의 구성은 신속한 통신 서비스 제공, 대용량 데이터 제공, 네트워크 구축 비용 감소, 효과적인 IC 및 정보기기 설계 등에 잘 적용가능하고 이를 통한 효과적인 정보 시스템의 구축 및 정보기기의 개발이 가능해졌다.

일반적으로 다항 적(Polynomial) 문제 영역에서의 최적화 트리인 최소신장트리에 비해 비-다항 적(Non-Polynomial) 문제 영역의 스타이너 최소트리는 구성 노드들을 좀 더 단축된 연결 길이를 갖는다. 그러나 비-다항 적 문제에 포함된 영역에 속하는 스타이너 최소트리의 생성은 적절한 휴리스틱 개발이 필수적이며, 이 때 노드의 수가 매우 큰 경우에는 막대한 시간과 계산양이 요구된다. 이를 해결하기 위해 매우 많은 노드들에 대해 분할 및 합병과정을 이용한 분산처리를 할 수 있다면, 전체 노드를 대상으로 처리하는 것 보다 실행시간을 단축할 수 있다. 그러나 이 경우는 전체 노드 대상의 순수 스타이너 최소트리보다는 연결 길이가 증가함을 피할 수 없을 것이다. 따라서 3차원 공간상에 위치한 전체 노드를 각 단위 문제 영역으로 적절히 분할하여 독립적으로 3차원 스타이너 최소트리를 생성한 후, 차 상위 계층에서 인접한 하위 영역의 결과들을 병합하고, 최종적으로 최상위 스타이너 트리를 생성한다.

본 논문에서는 순수 스타이너 최소트리의 큰 문제점인 과도한 실행시간을 해결하기 위해, PTAS 3차원 스타이너 트리를 활용한 3차원 공간의 노드들을 신속히 연결하는 방법을 제안하고자 한다. 이 방법은 비-다항 적 문제인 Euclidean Traveler Salesman 문제 해결을 위한 PTAS(Polynomial Time Approximation Scheme)[1,2]를 응용하여, 3차원 공간상에 위치하는 많은 입력 노드들에 대하여 역시 비-다항 적 문제인 근사 스타이너 최소트리를 생성하는 것이다. 제안된 방법은 순수 스타이너 최소트리와 비교하여 약간의 연결 길이의 증가를 허용하면서, 신속하게 효율적인 트리를 생성할 수 있다. 이 방법은 분할과 병합이라는 과정을 계층적으로 실행하고, 각 분할 계층의 공간 경계면에 위치하는 포털이라는 가상 연결 노드들을 활용한다. 제안된 방법은 실험을 통해 3차원 공간에서의 순수 스타이너 최소트리 생성시간과 연결 길이를 비교하여 성능의 개선내용을 확인한다. 본 논문의 구성은 2장의 관련된 연구들의 기술, 3장의 제안하는 PTAS 3차원 스타이너 최소트리 생성에 관한 기술, 4장의 실제 구현 및 실험분석, 5장의 결론이다.

II. 관련 연구

본 논문에서 제안하는 방법은 PTAS 기법을 이용하여 신속하게 3차원 스타이너 최소트리를 구현하여 3차원 공간에 위치한 입력노드들을 효과적으로 연결하는 것이다. 따라서 이와 관련된 연구는 PTAS와 3차원 스타이너 트리에 관한 것이다.

PTAS와 관련된 연구들은 다양한 분야에서 수행되었다.

먼저 재고 관리 문제에 PTAS를 적용한 연구가 있다[3].이 연구에서 단일 항목 확률적 재고 관리 문제의 해결을 위해 완전 다항 적 시간 근사 법(Fully Polynomial-time Approximation Scheme)을 설계, 제안하였다. 또한 시차(Time Lags)가 있는 기계 스케줄링 문제에 대한 PTAS 연구가 있다[4]. 이 연구에서는 PTAS를 이용하여 생산 공정의 작업 시작과 종료시간의 차이를 나타내는 makespan을 최소화 하는 방법과 전체 생산종료 시간을 최소화하는 방법을 연구하였다. 이 연구에서 제안된 근사법의 실행시간은 작업 수에는 다항 적이었으나 기계의 수에는 지수 적으로 증가하였다고 주장하였다. n 명의 고객(Customers)들의 위치와 한 개의 창고(Depot), 그리고 적재 용량(Capacity)이 k 인 차량(Vehicle) 1대가 주어졌을 때, 최소 길이의 방문 여정을 찾는 문제인 Capacitated Vehicle 라우팅 문제에 대하여 PTAS를 적용한 연구가 있다[5]. 이 연구에서는 고객들과 창고는 평면에 위치함을 가정하였고, 거리는 Euclidean metric으로 표현하였으며, 유사-다항 적 시간 근사 법(Quasi-polynomial Time Approximation Scheme)을 제안하였다. QoS 라우팅과 관련된 문제를 조사 분석하고 효과적인 QoS 라우팅을 위한 완전 다항적 근사법(Fully Polynomial Time Approximation Scheme, FPTAS)을 제안한 연구가 있다[6]. 이 연구에서 제안된 FPTAS는 개발된 그래프 확대 기반의 동적 프로그래밍 접근 방법들을 적용하였고 이를 확장한 기법을 발표하였다. 단백질, DNA와 같은 고분자 구조 관련 연구들의 주요 문제들 중 하나는 3차원 최소 에너지 환경(Minimum Energy Configurations, MECs)을 결정하는 것이다. 이러한 생화학 및 분자 모델링에 3차원 스타이너 트리를 활용한 연구가 있다[7]. 이 연구에서는 고분자 구조의 모델링에 있어 스타이너 최소트리가 잘 활용될 수 있다고 주장하면서, 제안된 스타이너 최소트리를 물리적으로 의미 있는 방법으로 MEC 에너지와 상관관계가 있는 최소신장트리들과 비교하였고, 또한 스타이너 비율을 계산 및 검증하였다 이 연구에서 단백질을 스타이너 최소트리로 모델링하였을 때, 구성 원소인 탄소와 질소 원자는 스타이너 포인트에 해당될 수 있음을 주장하였다. 동일 계층(Layer) 또는 다른 계층에 숨겨진 스타이너 포인트 할당의 개념을 기반으로 3차원 IC를 위한 연구가 있다[8]. 이 연구에서는 다른 계층 신호 망(Signal Net)의 연결 노드 집합에 대하여, 합병 기반(Merge-based) 방식의 시간 구동(Timing-Driven) 3차원 직선(3D Rectilinear) 스타이너 트리 구성방법을 제안하였다. 또한 3차원 Stacked IC들의 라우팅을 위한 스타이너 트리에 관한 연구가 있다[9]. 이 연구에서 제안된 알고리즘을 트리 구성 단계와 생성 트리의 개선 단계로 구현하였고, 트리 구성 알고리즘은 입력된 온도 프로파일(Thermal Profile) 환경에서 지연-지향(Delay-oriented) 스타이너 트리를 생성하는 것이고, 트리 개선 알고리즘은 성능 제한 상에서의 온도 최적화를 위해 초기 라우팅 토폴로지의 저장 동안 Through-vias를 변경하도록 설계하였다.

III. 3차원 스타이너 최소트리의 신속한 구성을 위한 PTAS

1. 제안방법

본 논문에서 제안하는 공간상에 존재하는 수많은 입력노드들에 대한 PTAS 3차원 스타이너 최소트리를 생성하는 방법은 다음과 같다.

그림 1의 좌측 그림은 본 논문에서 고려하는 3차원 공간에 대한 표현이다. 이것은 PTAS에서의 최상위 계층을 의미한다. 각 입력노드들의 위치는 공간을 구성하는 X, Y, Z의 세 좌표축에서의 이격거리인 실수 값의 순서상으로 표현된다. 그림 1의 우측 그림은 3차원 문제 공간을 최하위 계층의 8개로 분할한 단위 문제 공간들을 표현한 것이다.

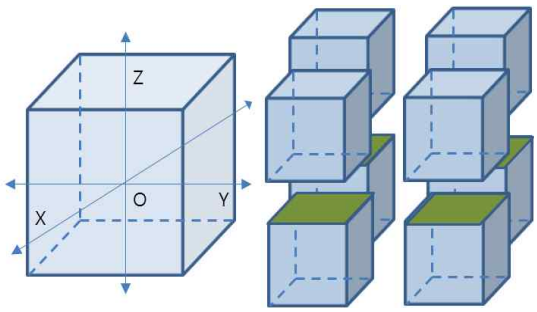

그림 1. 3차원 문제 공간 및 문제 공간의 최종분할
Fig. 1. 3D Problem Space and Final Partitions of the Space

그림 2는 전체 문제 공간을 이등분한 1차분할 방법을 보인다. 그림 3은 1차 분할의 결과로 생성된 두 개의 서브 문제 공간 N, N을 보이며, 또한 경계면에 위치한 5개의 포털들을 표현하고 있다. 포털의 위치와 개수는 필요에 의해 변경 가능하다. 그림 4는 그림 2의 분할 결과로 생성된 서브 문제 공간을 각각 이등분한 2차분할 방법을 보인다. 그림 5는 2차 분할의 결과로 생성된 네 개의 서브 문제 공간 I, J, K, L을 보이

며, 두 개의 경계면에 각각 위치한 5개의 포털들을 표현하고 있다. 그림 6은 그림 4의 분할 결과로 생성된 서브 문제 공간을 각각 이등분한 최종(3차)분할 방법을 보인다. 그림 7는 3차 분할의 결과로 생성된 8개의 서브 문제 공간 A, B, C, D, E, F, G, H를 보이며, 네 개의 경계면에 각각 위치한 5개의 포털들을 표현하고 있다.

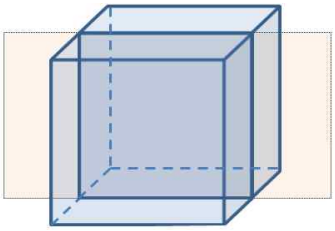

그림 2. 문제공간의 1차분할
Fig. 2. First Partition of Problem Space

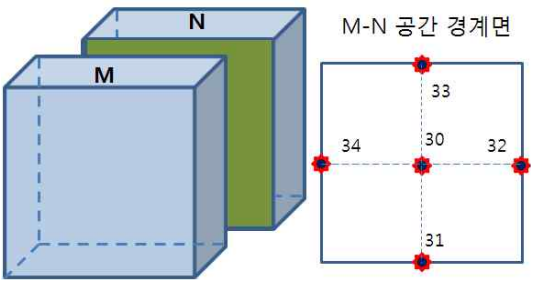

그림 3. 1차분할 공간 및 경계면의 포털
Fig. 3. Spaces of First Partition and Portals on Boundary Surface

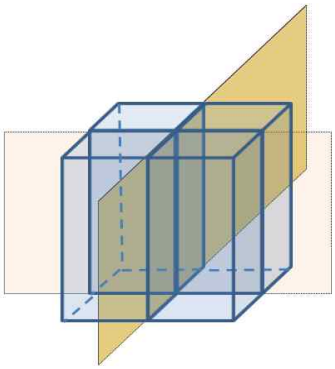

그림 4. 문제공간의 2차분할
Fig. 4. Second Partition of Problem Space

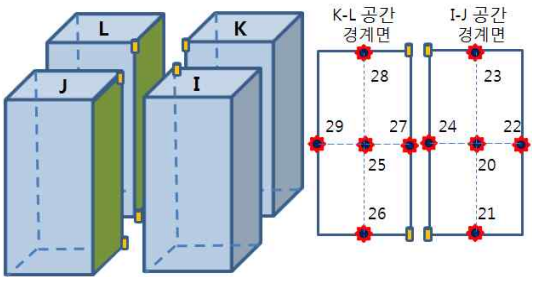

그림 5. 2차분할 공간 및 경계면의 포털
Fig. 5. Spaces of Second Partition and Portals on Boundary Surface

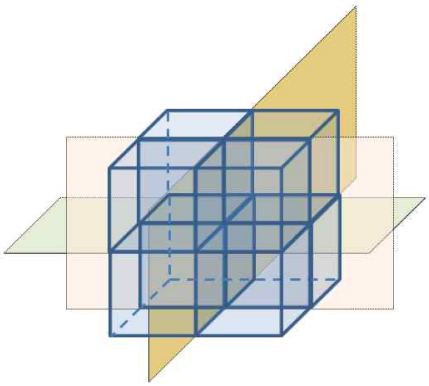

그림 6. 문제공간의 최종분할
Fig. 6. Final Partition of Problem Space

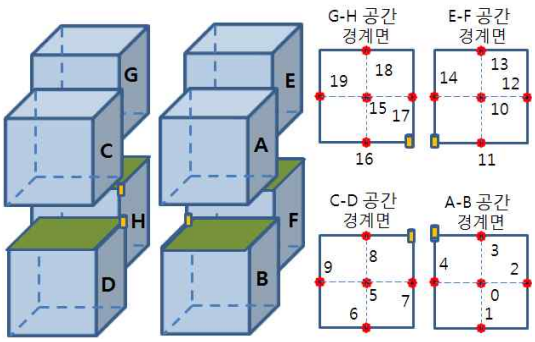

그림 7. 최종분할 공간 및 경계면의 포털
Fig. 7. Spaces of Final Partition and Portals on Boundary Surface

최종 분할로 인해 생성된 8개의 단위 문제 공간들 중 하나에 속한 여러 노드들중에서 임의의 세 개의 입력 노드 A, B, C에 대한 위치가 그림 8에 표현되어 있다. 이 노드들에 대한 3차원 스타이너 포인트를 구하는 개략적인 방법이 그림 9에 표현되어 있다.

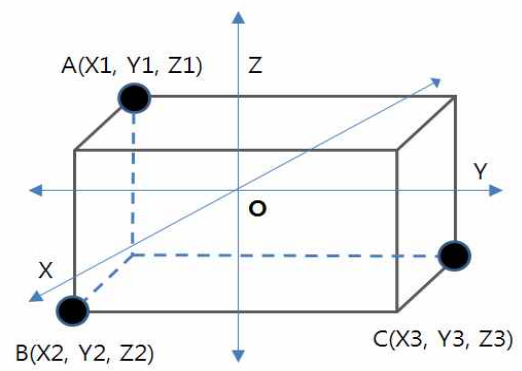

그림 8. 문제공간에 위치한 세 입력노드
Fig. 8. Three Input Nodes in Problem Space

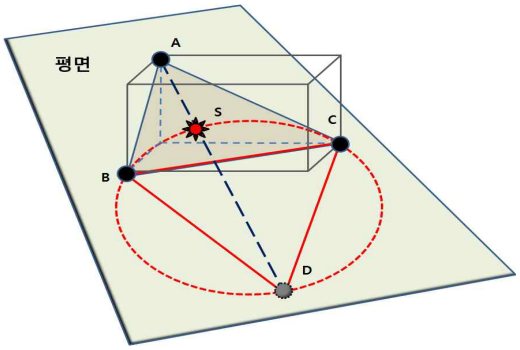

그림 9. 3차원 공간의 세 입력노드에 대한 스타이너 포인트
Fig. 9. Steiner Point for Three Input Nodes in 3D Space

표 1은 최하위 계층의 특정 단위 문제 공간 ϕ_i에서의 3차원 단위 스타이너 최소트리를 생성하는 알고리즘이다. 단계 1에서, 해당 단위 공간에 포함된 노드와 이들 간의 완전 연결 선분에 대하여 Prim 알고리즘을 이용한 3차원 최소신장트리 $3D-MST_i(1)$을 생성한다[10].

단계 2에서는 $3D-MST_i(1)$에 존재하는 임의의 두 연속선분 l_1, l_2을 찾고, 이 선분들을 구성하는 세 노드 p_1, p_2, p_3를 이용하여 평면 P를 구한다. 단계 3에서는 노드 p_1, p_2, p_3의 상호간 거리가 가장 큰 노드의 쌍을 p_1, p_2라고 가정할 때, 선분 $\overline{p_1p_2}$을 한 변으로 하는 정삼각형을 생성하고 또한 생성된 정삼각형의 외접원을 구한다. 정삼각형을 구성하는 노드들 중에서 입력노드가 아닌 새로운 정점을 D라 정의할 때, 외접원과 직선 $\overleftrightarrow{p_1D}$ 교점을 스타이너 포인트 S로 정의한다. $3D-MST_i(1)$에 있는 모든 연속 선분에 대하여 이러한 과정을 반복하여 스타이너 그래프 $3D-SMG_i$를 생성한다. 단계 5에서는 스타이너 그래프 $3D-SMG_i$에 존재하는 노드와 선분에 대한 새로운 최소신장트리 $3D-MST_i(2)$를 구한다. 단계 6에서는 $3D-MST_i(2)$에 존재하는 모든 스타이너 포인트의 연결차수를 조사하여 연결차수가 1인 경우 해당 스타이너 포인트 및 관련 연결을 삭제하고, 연결 차수가 2인 경우에는 스타이너 포인트만 삭제하고 해당 두 연결 노드들을 직접 연결하여 최종 3차원 스타이너 최소트리 $3D-SMT_i$를 생성한다.

표 1. 단위 문제 공간 ϕ_i에서의 3차원 스타이너 최소트리 생성 알고리즘
Table 1. 3D Steiner Minimum Tree Algorithm for Unit Problem Space ϕ_i

단계	작업 내용
1	단위 문제 공간내의 노드와 선분에 대한 Prim의 알고리즘을 이용한 3차원 최소신장트리$3D-MST_i(1)$를 생성한다[10].
2	$3D-MST_i(1)$에 존재하는 임의의 두 연속 선분을 구성하는 노드들을 이용하여 평면 P를 구한다.
3	세 노드의 길이 중 최대 길이를 한 변으로 하는 평면 P상의 정삼각형을 생성하고 이를 지나는 외접원을 이용하여 스타이너 포인트를 생성한다.
4	$3D-MST_i(1)$에 있는 모든 연속 선분에 대하여 과정 2, 3을 반복하여 스타이너 그래프 $3D-SMG_i$를 생성한다.
5	스타이너 그래프 $3D-SMG_i$에 존재하는 노드와 선분을 입력 데이터로 하여 새로운 최소신장트리$3D-MST_i(2)$를 생성한다.
6	$3D-MST_i(2)$의 모든 스타이너 포인트의 연결차수를 조사하여 불필요한 연결선분을 삭제하여 최종 결과물인 3차원 스타이너 최소트리 $3D-SMT_i$를 생성한다.

표 2는 본 논문에서 제안하는 PTAS 3차원 스타이너 최소트리 생성 알고리즘의 실행 과정을 설명한 것이다. 단계 1에서 입력 노드들을 3차원 공간에서의 위치에 따라 미리 정해진 단위 공간인 분할 공간에 할당한다. 본 논문에서는 이 분할 공간의 수를 8로 설정하였다. 단계 2에서는 각 경계면상의 포털에 대하여 경계면에 인접한 두 분할공간의 노드들 중에서 각각 한개씩 선택한 노드들의 쌍 중에서 거리의 합이 가장 작은 것을 선택하고 해당 노드들에 대한 정보를 해당 포털에 저장한다. 단계 3에서는 각 분할된 공간에 할당된 모든 입력 노드들에 대하여 해당 할당 공간 내에서의 완전 연결하는 선분을 생성한다. 단계 4에서는 해당 분할공간에서의 3차원 스타이너 최소트리를 표 1에 있는 3차원 스타이너 최소트리 생성 알고리즘을 이용하여 생성한다. 단계 5에서는 전체 문제 공간에 대한 모든 분할 영역에 대하여 단계 3과 단계 4를 반복하

여 실행한다. 단계 6에서는 단계 2에서 저장한 포털 정보를 이용하여 3차 분할계층의 인접 스타이너 트리들을 병합하고, 단계 7 에서는 2차 분할계층의 인접 스타이너 트리들을 병합하고, 단계 8에서는 1차 분할계층의 인접 스타이너 트리를 병합한다. 단계 9 에서는 단계 8 의 결과 스타이너 트리에서 포털과 연결된 두 노드들을 직접 연결하여 전체의 길이를 최소화한 PTAS 3차원 스타이너 최소트리를 생성한다.

표 2. PTAS 3차원 스타이너 최소트리 생성 알고리즘
Table 2. PTAS 3D Steiner Minimum Tree Algorithm

단계	작업 내용
1	모든 입력 노드들을 공간상의 위치에 따라 미리 정해진 분할공간에 할당한다.
2	각 경계면상의 포털에 대하여 인접한 두 분할공간의 노드들의 쌍 중에서 가장 근접한 것들을 찾는다.
3	각 분할된 공간에 소속된 입력 노드들에 대하여 해당 공간 내에서 완전 연결하는 선분을 구한다.
4	각 분할공간의 노드들과 생성된 연결선분에 대하여 표 1의 3차원 스타이너 최소트리 생성 알고리즘을 활용하여 최하위 계층의 분할공간에서의 3차원 스타이너 최소트리를 생성한다.
5	모든 분할 공간에 대하여 단계 3과 4를 반복한다.
6	최하위 계층의 인접한 두 공간 쌍, 즉 (A,B), (C,D), (E,F), (G,H)의 스타이너 최소트리들을 단계 2에서 찾은 포털정보를 활용하여 병합한 공간 I, J, K, L에서의 스타이너 트리를 생성한다.
7	2차 분할 계층의 인접한 두 공간 쌍, 즉 (I,J), (K,L)의 스타이너 최소트리들을 단계 2의 포털 정보를 이용하여 병합된 공간 M, N에서의 스타이너 트리를 생성한다.
8	1차 분할계층의 인접한 두 공간 쌍, 즉 (M, N)의 스타이너 최소트리들을 단계 2의 포털 정보를 이용하여 병합된 최상위 공간 O에서의 스타이너 트리를 생성한다.
9	단계7의 결과에서 포털을 경유하는 두 노드들을 직접 연결하여 연결 길이를 최소화한, 최종 PTAS 3차원 스타이너 최소트리를 생성한다.

표 1의 단위 문제 공간 ϕ_i의 3차원 최소신장트리 생성 알고리즘의 실행 시간을 분석하면 다음과 같다. 단계 1에서 존재하는 단위 문제 공간 ϕ_i의 노드의 수는 n_i, 선분의 수 e_i는 $n_i \times (n_i - 1)$이다. Prim의 알고리즘을 이용하여 3차원 최소신장트리 $3D-MST_i(1)$을 생성하기 위해서, 각 선분의 길이를 키 값으로 하는 우선순위 큐를 활용한다. 일반적인 이진 힙(Binary Heap)을 이용한 우선순위 큐를 사용하는 경우 최소신장트리의 생성시간은 $O(e_i \times \lg n_i)$이다[10]. 단계 2, 3, 4에서는 세 노드로 구성되는 평면과 이를 이용하여 세 노드에 대한 스타이너 포인트를 찾을 때, $3D-MST_i(1)$에 존재하는 연속적인 두 선분을 대상으로 이루어진다. $3D-MST_i(1)$에는 $n_i - 1$개의 선분이 존재하고 따라서 연속적인 두 선분의 수는 $O(n_i)$이다. 따라서 생성되는 스타이너 포인트의 수도 $O(n_i)$이 된다. 단계 4에서 생성된 스타이너 그래프 $3D-SMG_i$를 구성하는 노드의 수와 선분의 수가 모두 $O(n_i)$이므로 이것을 대상으로 한 단계 5에서 새로운 최소신장트리 $3D-MST_i(2)$를 생성하기 위해서는 $O(e_i \times \lg n_i)$의 시간이 필요하다. 단계 6의 불필요한 연결을 삭제하기 위해 점검되는 대상은 생성된 스타이너 포인트들이므로 이 단계에서는 스타이너 포인트의 수인 $O(n_i)$의 시간이 요구된다. 따라서 표 1의 단위 문제 공간 ϕ_i에서 3차원 스타이너 최소트리 생성을 위해서는 $O(e_i \times \lg n_i) = O(n_i \times \lg n_i)$의 시간이 필요함으로 분석할 수 있다. 만약 특정 단위 문제 공간에 모든 입력 노드 N개가 집중되어 위치되었을 경우에는 이 공간에서 처리되는 시간은 $O(N \times \lg N)$이 된다. 각 단위 문제 공간에 생성된 단위 스타이너 최소트리는 각 공간 경계면에 존재하는 포털들을 이용하여 인접 문제공간과 통합하게 된다. 각 공간 경계면에는 분할의 단계와는 무관하게 포털의 개수가 일정하게 5개이다. 이것은 문제에 따라 그 개수와 각각의 위치가 변동가능하다. 두 공간의 경계면에 존재하는 포털들에는 인접 두 공간의 노드들중에서 가장 근접한 노드의 정보가 저장되어있다. 따라서 두 개의 하위 문제공간을 통합할 경우, 최단 거리로 두 개의 문제공간에 대한 스타이너 트리를 신속하게 연결할 수 있다.

표 2의 단계 1에서 전체 입력 노드 N개를 각 분할공간에 할당하는데 소비되는 시간은 $O(N)$이다. 단계 2에서 각 경계면에 인접한 두 개의 단위 문제 공간 ϕ_i, ϕ_j의 노드의 수를 n_i, n_j라고 했을 때 실행시간은 $O(n_i \times n_j) = O(N^2)$이다. 단계 3에서 단위 문제 공간 ϕ_i의 노드 수를 n_i이라고 가정했을 때, 완전 연결을 생성하는 시간은 $O(n_i^2) = O(N^2)$이다. 단계 4에서 단위 문제 공간 ϕ_i의 노드 수를 n_i이라 했을 때, 이 공간에서 3차원 스타이너 최소트리를 생성하는 시간은 앞에서 기술한 것처럼 $O(n_i \times \lg n_i) = O(N \times \lg N)$이다. 단계 5에서는 단계 3과 4를 분할의 수 k만큼 반복하므로 $k \times O(N^2) = O(N^2)$이다. 단계 6, 7, 8, 9에서는 사용되는 포털의 수 q만큼 실행되므로 입력노드의 수와는 상관없는 $O(q) = O(1)$이다. 따라서 본 논문에서 제안하는 방법의 실행시간은 $O(N^2)$이다. 그러나 입력노드들의 위치가 각 노드

에 거의 균등하게 분포되어있어 그 할당된 노드수가 $n_a \le N$ 라고 가정하면 실행시간은 $O(n_a^2)$ 이므로, 순수 3차원 스타이너 최소트리의 실행시간인 $O(N^2)$ 에 비해 개선될 수 있음을 확인할 수 있다.

2. 실험 및 결과분석

2.1 실험방법

본 논문에서 제안된 방법을 검증하기 위해서, 3차원 입력 노드의 수가 10000, 30000, 50000, 70000인 경우에 대하여 실험하였다. 관찰 결과는 전체 입력노드를 대상으로 한 순수 3차원 스타이너 최소트리를 이용한 방법과 본 논문에서 제안하는 표 2에서 기술된 PTAS 기법을 이용한 3차원 스타이너 최소트리 생성 방법에 의한 트리의 연결 비용, 즉 트리의 길이와 트리 생성시간이다. 본 논문에서는 전체 문제 공간을 최하위 계층의 8개의 단위 문제 공간으로 분할하여 3차원 입력 노드들을 그 위치에 따라 할당하였다. 이 분할 공간의 수는 응용에 따라 다르게 적용할 수 있을 것이다.

본 실험의 주목적은 제안하는 방법이, 순수 3차원 스타이너 최소트리 방법보다 비록 연결 길이는 약간 증가하지만 트리의 생성 시간이 대폭적으로 감소됨을 확인하기 위함이다. 따라서 본 논문에서 제안하는 PTAS 3차원 스타이너 최소트리와 순수 3차원 스타이너 최소트리의 연결 길이의 합과 실행시간이 비교된다. 이 방법들은 4 기가바이트 램, Intel 프로세서, Microsoft 윈도우즈 환경의 랩톱 컴퓨터에서 Java로 구현하여 비교 분석되었다. 실험에 사용된 10000, 30000, 50000, 70000개의 3차원 입력노드들은 8개 문제 공간영역에 균등 분포하도록 랜덤하게 그 좌표를 각각 설정하였다.

2.2 실험 결과 및 분석

그림 10은 입력 노드의 수에 따라 순수 3차원 스타이너 최소트리 알고리즘에 의해 생성된 트리와 본 논문에서 제안된 PTAS 3차원 스타이너 최소트리 방법에 의해 생성된 트리의 생성시간을 분석한 결과이다. 제안된 방법은 입력노드의 수에 따라 순수 3차원 스타이너 최소트리 방법에 비해 85.56%에서 86.88%의 실행시간 감소를 나타낸다. 이 그림에 따르면 입력노드의 수가 증가할수록 실행시간의 감소율이 높아짐을 알 수 있다. 10000개의 3차원 입력노드에 대하여 제안된 PTAS 방법은 85.56%의 시간 감소율을 보였고, 70000개의 입력노드들에 대해 86.88%의 시간 감소율을 나타내었다. 즉 입력 노드의 수가 증가할수록 제안된 방법의 시간 감소율이 높아짐을 확인할 수 있다. 이는 초대형 네트워크에서 효율적인 연결방법을 신속하게 찾는데 본 논문에서 제안한 방법이 효과적으로 사용될 수 있음을 보여준다.

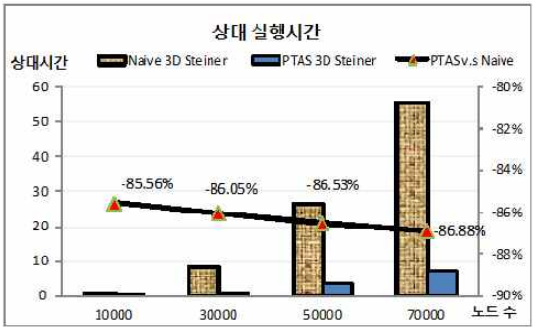

그림 10. 트리 생성을 위한 상대적 시간 비교
Fig. 10. Comparison of Tree Construction Relative Time

그림 11은 입력 노드의 수에 따라 순수 3차원 스타이너 최소트리 생성 알고리즘에 의해 생성된 트리와 본 논문에서 제안한 PTAS 방법에 의하여 생성된 PTAS 3차원 스타이너 최소트리의 길이를 분석한 결과이다. 본 논문에서 제안하는 방법은 순수 3차원 스타이너 트리에 비해 입력 노드 수가 10000개인 경우 1.71%, 30000개인 경우 1.11%, 50000개인 경우 0.94%, 70000개인 경우 0.81%의 트리 길이 증가를 보였다. 특이한 사항은 입력 노드의 수가 증가할수록 트리 길이의 증가율이 둔화된다는 사실이다. 이는 본 논문에서 제안한 방법이 적은 오버헤드로 많은 노드들을 신속하게 연결하는 방법을 찾는데 유리함을 나타내고 있다. 따라서 실험을 통해 제안된 방법은 PTAS 기법을 이용한 방법은 기존의 순수 3차원 스타이너 트리 방법의 가장 큰 문제점이었던 실행시간을 획기적으로 감소시킴을 확인할 수 있었다. 그리고 그 과정에서 발생하는 오버헤드인 길이의 증가도 미비함을 확인할 수 있었다. 이는 제안된 방법이 시간적 제약조건이 큰 환경에서 많은 노드들을 신속하고 효과적으로 연결하려는 응용에서 잘 활용될 수 있음을 확인할 수 있었다.

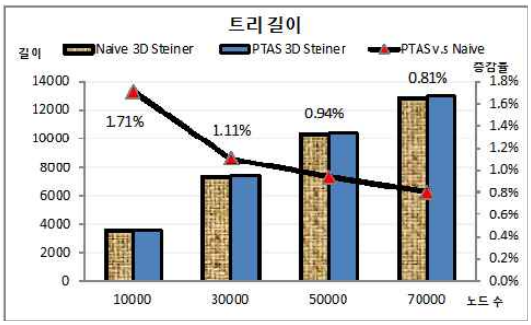

그림 11. 생성된 트리의 길이 비교
Fig. 11. Comparison of Built Tree Length

IV. 결 론

본 논문은 3차원 공간상에 위치한 많은 노드들을 효율적이면서도 신속하게 연결하는 방법을 찾기 위한 PTAS 3차원 스타이너 최소트리 생성에 관한 연구이다. 최적화 문제에 대한 많은 연구들이 2차원 평면을 대상으로 하였으나, 기술이 발전 및 서비스 기대 수준이 높아짐에 따라 교통, 통신 시설 등의 처리 문제에 있어서 3차원적인 접근방법이 요구되고 있다[11]. 다항 적 시간문제에서 길이의 최적화 알고리즘인 최소신장트리보다 더 효과적인 것인 비-다항적 문제로 확장하여 스타이너 최소트리를 활용하는 것이다. 그러나 연결하려는 노드의 수가 매우 많을 경우, 순수 스타이너 최소트리의 생성을 위해서는 매우 많은 실행 시간 및 계산 양을 필요로 하는 문제가 발생한다.

본 논문에서는 다항 적 시간 근사 법(Polynomial Time Approximate Scheme, PTAS)을 이용하여, 3차원 공간에 위치한 많은 노드들을 연결함에 있어서, 문제 영역을 8개의 단위 문제 영역으로 분할하고 이들의 각각 연결방법을 찾은 후에, 포털이라는 연결지점을 활용하여 계층적으로 통합하는 PTAS 3차원 스타이너 최소트리 생성 방법을 제안하였다. 70000개의 공간상의 노드들을 연결함에 있어, 제안된 PTAS 3차원 스타이너 최소트리 연결방법은 순수 PTAS 3차원 스타이너 최소트리 방법에 비해 비록 연결 길이는 0.81% 증가하였으나, 생성시간은 86.88% 감소하였다. 이는 본 논문에서 제안하는 방법이 3차원 공간상의 수많은 노드들에 대하여 약간의 길이 증가를 허용하면서 신속하게 연결방법을 찾고자 하는 응용에 잘 적용될 수 있음을 보인다.

향 후 연구내용은, 기존의 PTAS를 계량하거나 수정 보완하여 제안된 방법의 오버헤드인 트리의 길이 증가율을 감소할 수 있는 방법에 대한 것과 3차원 입력노드들의 공간분포도가 균등하지 않을 때 효과적인 공간 분할 방법에 대한 연구이다. 또한 최하위 계층의 분할된 공간의 크기가 동일한데, 공간상의 노드 분포도에 따른 크기의 변화에 대한 연구도 필요하다. 이러한 연구를 통해 본 논문에서 제안된 방법은 3차원 공간상에 위치한 수많은 노드들에 대하여 더 효율적이고 신속한 연결방법을 찾을 수 있을 것이고, 이와 관련된 응용에 잘 적용될 수 있을 것이다.

참고문헌

[1] G. Xue, G. Lin and D.Z. Du, "Grade of Service Steiner Minimum Trees in Euclidean Plane," Algorithmica, Vol.31, pp.479-500, 2001

[2] http://en.wikipedia.org/wiki/Polynomial-time_approximation_scheme, March 2012

[3] N. Halman, D. Klabjan, M. Mostagir, J. Orlin and D. SimchiLevi, "A Fully Polynomial-Time Approximation Scheme for Single-Item Stochastic Inventory Control with Discrete Demand," Journal of Mathematics of Operations Research Vol.34, No.3, pp.674-685, 2009

[4] X. Zhang, S. Velde, "Polynomial Time Approximation Schemes for Scheduling Problems with Time Lags," Journal of Scheduling, Vol.13, No.5, pp.553-559, 2010

[5] A. Das and C. Mathieu, "A Quasi-polynomial Time Approximation Scheme for Euclidean Capacitated Vehicle Routing," Proceeding of SODA '10 Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms Society for Industrial and Applied Mathematics, pp.390-403, Austin, USA, 2010

[6] J. Huang and Y. Tanak, "QoS routing algorithms using fully polynomial time approximation scheme," Proceeding IWQoS '11 Proceedings of the IEEE 19th International Workshop on Quality of Service, pp.1-3, 2011

[7] C. Stanton and J. Smith, "Steiner Trees and 3-D

Macromolecular Conformation," INFORMS Journal on Computing, Vol.16, No.4 pp.470-485, Fall 2004

[8] J. Yan, Z. Chen and D. Hu, "Timing-driven Steiner tree construction for three-dimensional ICs," Proceeding of Circuits and Systems and TAISA Conference NEWCAS-TAISA, pp.335-338, June, 2008

[9] M. Pathak, and S. Lim, "Thermal-aware Steiner routing for 3D stacked ICs," Proceeding of ICCAD '07 Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design, pp.205-211, 2007

[10] T.H. Cormen, C.E Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Ed., THe MIT Press, pp.561-579, 2001

[11] 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: 김포대학 인터넷정보과 부교수

관심분야: 네트워크, 데이터베이스

Email : ibkim@kimpo.ac.kr