

논문 2010-47TC-11-1

# 센서네트워크 상의 TSP 경로구성 방법에 대한 분석

## ( Analysis for a TSP Construction Scheme over Sensor Networks )

김 준 모\*

( Joonmo Kim )

### 요 약

센서네트워크 등에서 단말 또는 노드들을 한 번씩 모두 방문하는 최적의 라우팅 경로를 동적으로 찾는 문제가 대두된다. 이러한 문제를 근사하게 해결 할 수 있는 일반화된 scheme을 제시하고, 이를 기반으로 구성되는 알고리즘의 실행시간 및 그 결과의 바운드를 수리적으로 정립하면, 주어진 네트워크에서 구축되는 라우팅 경로를 수리적으로 분석 할 수 있게 된다. 본 논문은 이러한 문제를 대표하는 Euclidean TSP(Euclidean Travelling Sales Person) 문제를 대상으로 하여, 근사 Euclidean TSP를 병렬처리 형태로 구성할 수 있는 scheme을 제공하고, 이 scheme에 의해 구해 질 수 있는 근사 Euclidean TSP가 최적의 Euclidean TSP와 어느 정도의 차이를 가지게 되는지 판단할 수 있는 기준을 제시한다.

### Abstract

In Sensor Networks, the problem of finding the optimal routing path dynamically, which passes through all terminals or nodes once per each, may come up. Providing a generalized scheme of approximations that can be applied to the kind of problems, and formulating the bounds of the run time and the results of the algorithm made from the scheme, one may evaluate mathematically the routing path formed in a given network. This paper, dealing with Euclidean TSP(Euclidean Travelling Sales Person) that represents such problems, provides the scheme for constructing the approximated Euclidean TSP by parallel computing, and the ground for determining the difference between the approximated Euclidean TSP produced from the scheme and the optimal Euclidean TSP.

**Keywords** : Sensor Networks, Approximation Algorithm, Euclidean TSP, Rectangular Partition, Routing

## I. 서 론

이동하는 단말을 포함한 센서 네트워크<sup>[1]</sup>에서 단말들을 하나씩 차례로 모두 방문하기 위한 라우팅 경로를 동적으로 찾아야 하는 경우, 단말을 노드로 하고 단말 사이의 거리를 간선으로 하는 그래프를 방문하는 문제를 고려하게 된다. 이러한 문제는 Euclidean 평면상에서 TSP(Travelling Sales Person)를 구축하는 문제로 추상화되며, 이렇게 추상화된 문제 상에서 ETSP(Euclidean

Travelling Sales Person)를 구축하기 위한 모델링을 하게 된다<sup>[2-3]</sup>. 모델을 기반으로 ETSP를 구축하기 위한 방법 및 절차가 구성되고<sup>[4-6]</sup>, 관련한 분석이 진행된다. 본 논문에서는, 센서노드들의 개수가 많아 서로 밀집하여 배치된 경우, 병렬처리를 통하여 ETSP에 근사하는 경로를 생산하는 방법(scheme)을 제안하고 분석하는 것을 주요 내용으로 하고 있다. 지역별로 분산되어 이동하거나 정지해 있는 단말 또는 서버의 위치 정보를 병렬처리하면서 통합함으로써, 방문을 위한 light구조(근사-ETSP)를 구축하는 것이다. 이러한 light구조는 대부분의 경우 최적의 ETSP보다 더 긴 길이를 가지게 되는데, 실제 응용에서 이러한 비용의 증가가 허용될 수 있음을 밝히는 데에 사용될 수 있는 수리적인 기준을 본 논문의 모델링과 분석을 통하여 도출하고 있다.

\* 정회원, 단국대학교 컴퓨터학부  
(Member, Computer Science & Engineering,  
Dankook University)

※ 이 연구는 2009학년도 단국대학교 대학연구비의 지원으로 연구되었음.

접수일자: 2010년4월19일, 수정완료일: 2010년11월10일

이렇게 light구조를 구축하는 근사 알고리즘이 가지는 성능은 그 결과값의 적정성과 실행시간의 측면에서 분석하여야 한다. 오차정리는 light구조 및 이를 도출하기 위한 주요 알고리즘을 분석하기 위한 기준이 된다. 오차정리는 ETSP에 근사하는 light구조의 전체길이가 ETSP 전체길이에 비해 최대로 어느 정도까지 더 길어질 수 있는지를 나타낸다. 주어진 문제 영역을 분할하여 분할된 영역별로 구한 light구조들을 절차에 따라 통합하여 만든 최종적인 light구조가 ETSP에 어느 정도 근사하는지를 오차정리를 통하여 알 수 있다. 따라서 오차의 허용가능 여부를 어느 정도 예측하고 판단할 수 있게 된다. 또한, 오차정리를 수립하는 데에 적용된 기법 등을 일반화하고 활용함으로써 light구조를 구축하기 위한 구체적인 방법 및 알고리즘을 도출할 수 있다. ETSP는 Euclidean 평면상의 Travelling Sales Person 문제이다. 평면에 주어진 점(노드)들을 모두 차례로 한번씩 방문하면서 원위치로 돌아 올 수 있는 단순경로를 찾는 문제이다. 따라서 주어진 problem instance 상의 그래프 구조가 가지는 간선 길이의 합을 비용으로 하여, 이를 일정수준 이하로 제한할 수 있는 scheme을 구성한다. ETSP는 그래프  $G=(V,E)$ ,  $|V|=n$  형태로 표현되며,  $V$ 의 원소인 각 노드는 Euclidean 평면상의 좌표를 가진다. EMST(Euclidean Minimum Spanning Tree)를 사용하여 ETSP에 근사하는 light구조를 구축하게 되는데<sup>[7]</sup>, EMST의 성질과 함께, 평면상의 간선들 사이에 성립하는 triangular inequality를 바탕으로 보조정리1과 오차정리를 도출하게 된다<sup>[8]</sup>. 보조정리1에서는 실수평면인 Euclidean 평면상에 정의된 ETSP 문제를 이산좌표로 이동함으로써 생기게 되는 오차를 수식으로 나타내었다. 현실의 기하학적인 문제는 대부분 실수좌표계에서 추상화되므로, 알고리즘 적용을 위해서는 이산화 작업이 선행되는 것이다. 보조정리1은 이러한 이산화 작업에 의해 발생하는 오차의 범위를 나타내는 것이다. 다음 장에 정의되는 파티션은 알고리즘의 적용을 위해 주어진 평면 영역을 분할하는 방식이다. 일정크기의 직사각형 형태로 분할된 영역에서 light구조를 산출하고, 이들을 단계적으로 접합하면서 전체 영역에 대한 light구조를 구성하게 된다. light구조의 간선이 분할선과 교차하는 경우, 추가적인 오차를 허용함으로써, 분할선 상의 지정된 위치에서만 교차할 수 있도록 한다. 이러한 지정된 교차지점을 포탈이라고 한다. 포탈에 의하여 제한된 복잡도를 가지는 알

고리즘의 구성이 가능하게 되고, light구조와 ETSP의 차이를 분석할 수 있게 된다.

II장에서는 관련연구를 기반으로 본 논문에서 사용되는 기초적인 도구를 정의하고 정리를 나타내었다. III장에서는 light구조를 구축하는 절차와 해당하는 실행시간을 분석하였다. IV장에서는 오차정리를 증명하였다.

## II. 관련연구 및 정의

본 논문은 관련연구<sup>[2-6]</sup>에서 사용하는 증명 기법을 이용하여, 제안된 scheme 하에서 ETSP가 가지는 성질(property)을 찾아내는 것이다. 관련연구에서는, 주로 평면기하 분야의 NP-hard 문제를 대상으로 주어진 문제 영역을 분할한다. 분할된 각 영역에서 최적으로 근사하는 해를 구하고, 구해진 영역별 해를 단계적으로 접합하여 최종적으로 최적으로 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 ETSP 문제는 실수좌표  $R^2$ 에서 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. 보조정리1에서는, 실수좌표  $R^2$  상에 주어진 ETSP 문제를 이산좌표로 옮기는 경우에 발생하는 ETSP 전체간선 합의 최대 변화량을 나타낸다.

보조정리1: 실수좌표  $R^2$ 에서 정의된 ETSP 문제를 이산좌표로 변환하는 경우, 변환에 의한 ETSP 전체간선 합의 변화량은 최대  $n$ 이다.

*proof*: 주어진 실수좌표 상의 ETSP를 이산좌표로 변환하는 경우, 다음 관계가 성립하게 된다.

$$|\Sigma e_{real} - \Sigma e_{discrete}| \leq 2\sqrt{2}n$$

$\Sigma e_{real}$ 은 실수좌표  $R^2$  상에 정의된 ETSP problem instance로부터 구할 수 있는 ETSP 전체 간선들의 합이고,  $\Sigma e_{discrete}$ 는 이산좌표로 변환된 problem instance로부터 구할 수 있는 ETSP 전체 간선들의 합이다. ETSP는  $n$ 개의 간선을 가지며, triangular

inequality에 의해 간선 하나의 한 끝에서 최대  $\sqrt{2}$ 의 오차를 가지기 때문이다. ■

Euclidean 평면위에 주어진 노드(점) 집합을 에워싸는 최소 면적의 직사각형을 박스라 한다. 박스는 정의2에서와 같이 직사각형으로 순환적으로 분할되며, 직사각형의 각 변은 좌표축과 평행하거나 직교한다. 하나의 직사각형을  $R$ 이라할 때, 분할선은  $R$ 을 두 개의 하위-직사각형으로 분할하는 선분이며, 그 길이를  $L$ 로 표시한다.

정의2(파티션): 직사각형  $R$ 에 대한 파티션은 다음과 같은 하위-직사각형들의 바이너리 트리 구조이다. 직사각형  $R$ 은 트리의 루트 노드이다.  $R$ 의 크기가  $\leq 1$ 이면, 트리구조는  $R$  자신이고, 이외의 경우에는 하나의 분할선으로써 파티션 구조를 가지는 두 개의 하위-직사각형으로 분할된다.



그림 1. 분할에 의한 파티션  
Fig. 1. Partitions by separations.

정의3(포털): 포털은 파티션에서 각 분할선에 존재하는 점(point)으로 분할선의 중앙에 위치하는 하나의 점이다.

정의4(light구조): 주어진 문제를 만족하는 연결구조를  $\pi$ 라 하고, 문제에서 정의된 영역인 박스에 대한 파티션을  $S$ 라 하자.  $S$ 의 직사각형 각각에서 분할선이  $\pi$ 의 간선들과 포털에서만 단 한번 교차할 때, 이러한  $\pi$ 를  $S$ 에 대한 light구조라 한다.

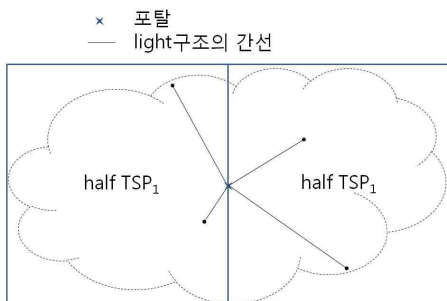


그림 2. Light 구조  
Fig. 2. Light structure.

파티션에 속하는 하나의 직사각형  $R$ 을 두 개의 하위-직사각형  $R_1, R_2$ 으로 분할하고, 각각의 하위-직사각형에서 구축된 두 개의 light구조를 접합하여 통합 light구조를 구성하는 경우; 각각의 하위-직사각형에서, (가상의) 최적 ETSP에 ratio 1.5로 근사하는 ETSP를 구할 수 있다<sup>[7]</sup>.

정리5(오차정리):  $R$ 에서 근사하게 구한 ETSP를 ‘근사TSP’ 라하고,  $R_1$  또는  $R_2$ 에서 구한 것을 ‘half TSP<sub>1</sub>’라고 할 때, 간선들의 길이가 0에 접근하면,  
 $2Length(half TSP_1) \leq Length(근사 TSP) + 2L$ 이다.

다음으로, light구조를 구축하기 위한 동적 프로그래밍을 설명하면서 그 실행시간을 분석한다. 그리고 오차정리를 증명한다.

### III. light구조의 구축과 실행시간 분석

파티션 상에서 light구조를 구축하는 알고리즘을 구성할 수 있다. 본 장에서는 최소비용을 가지는 light구조를 찾아내는 알고리즘의 실행시간이 최악의 경우에  $O(n^4)$ 임을 보인다. 다음 장에서 증명 될 오차정리는, 알고리즘에 의해 구성되는 light구조는 분할선을 기반으로 한단계 통합될 때마다  $2L$ 만큼의 길이가 추가 될 수 있다는 사실을 말한다. 이러한 오차정리와 보조정리1은 알고리즘의 결과인 근사 ETSP를 평가하는 기준이 된다. 알고리즘은 bottom-up으로 진행하지만, 편의상 최종단계를 시작으로 하여 top-down 관점에서 보면 다음과 같다. 알고리즘의 최종단계는 주어진 노드들을 둘러싸는 박스로서, 모든 노드들을 ratio 1.5로 근사하게 TSP 방문하는 light구조를 포함하는 직사각형이다. 최종단계의 직사각형에서 절차가 진행되기 위해서는 파티션에 속한 두 개의 하위-직사각형을 기반으로 한 light구조들을 통합하여야 한다. 이러한 절차를 분할된 하위-직사각형에 대해 재귀적으로 적용하면서 진행한다. 이제 bottom-up으로 진행되는 알고리즘의 절차를 설명한다. light구조를 구성하기 전에 파티션을 먼저 구축한다. 파티션의 깊이는 실제 응용환경을 반영하여 고려하게 되는데, 여기서는 파티션의 깊이를  $d$ 라는 상수로 표현한다.  $d$ 는 파티션에 의해 만들어지는 영역들의 개수에 대해 로그 관계를 가지게 된다. 파티션이 구축되면, 분할선들이 결정되고, 각 분할선의 중심점을 포털로 지정

한다. 따라서 분할된 각 영역은 사각형으로 표현되고, 하나 이상의 변에는 포털이 존재한다. 하나의 분할된 영역에 대해, 그 내부에 존재하는 노드들과 변에 존재하는 포털을 대상으로 ratio가 2 또는 1.5인 근사TSP를 구성 할 수 있다. 더 작은 ratio를 가지는 근사TSP는 실행시간이 크게 증가하므로 사용하지 않는다. 고려대상이 되는 각 부분에 대한 실행시간은 다음과 같다. MST(Minimum Spanning Tree)를 구축하고, 이에 대한 touring을 하는 경우 ratio 2인 근사TSP를 구할 수 있다<sup>[8]</sup>. 이때, MST 구축에는  $O(n \log n)$ 시간이 소요되며, MST에 대한 touring에는  $O(n)$ 시간이 소요된다. 따라서 실행시간은  $O(n \log n)$ 이다. Christofids' scheme<sup>[7]</sup>을 사용하는 경우에는 ratio 1.5인 근사TSP를 구할 수 있다. 이때 실행시간은  $O(n^3)$ 이다. 여기서 ratio가 1.5인 근사TSP를 고려한다. 각 영역별로 light 구조가 구성되면, 인접한 영역들의 쌍을 통합하면서 통합된 light구조를 구성한다. 인접한 영역들은 분할선을 사이에 두고 좌우로 접하게 되는데, 통합된 light구조는 포털을 교점으로 좌우의 light구조가 만나는 모습이 된다. 따라서 통합된 light구조는 TSP 고유의 성질(circle 형성)을 잃게 된다. 하지만, 포털을 인접한 네 개의 점이라고 다시 정의하고, 이들은 정사각형으로 배치되며, 각 변의 거리는 (분석에 영향을 주지 않는) 최소한의 상수인  $\epsilon$ 으로 정의된다고 하자. 이러한 포털에서 간선들을 수정함으로써 TSP 고유의 성질을 만족하도록 할 수 있다(그림 3). 이렇게 구성된 통합 light구조에서 새롭게 정의된 포털 주변의 네 개의 점들을 적절히 이동시키거나 삭제함으로써, 수정된 light구조가 되도록 할 수 있다. 이때, 수정된 light구조는 통합된 light구조보다 길이의 총 합이 줄어드는 방향으로 수정하는 것이다.

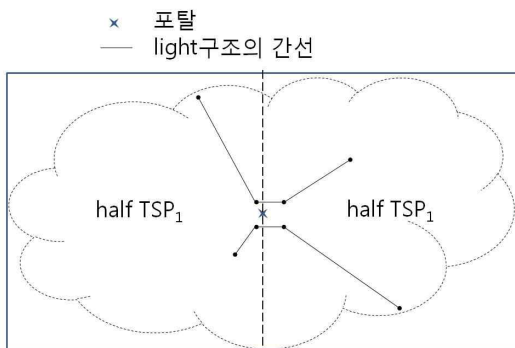


그림 3. 포털 변경  
Fig. 3. Portal change.

점들을 삭제 할 때에는, 삭제 후 인접하게 되는 노드들을 직접 연결함으로써 길이의 총 합이 줄어들게 한다. 지정된 개수의 노드들을 대상으로 임의의 수준으로 길이를 줄이는 작업이므로  $O(1)$ 의 실행시간을 소요하게 된다. 수정된 light구조를 만드는 작업은 실용적인 측면에서 실행 할 수 있으나, 근사 정도를 도출하기 위한 오차정리 부분의 분석 내용과는 무관하다. 통합된 light구조를 구성하는 이상의 동작을 재귀적으로 실행하면서, 최종단계에서 전체 노드와 포털을 대상으로 하는 최종적인 light구조를 구성하게 된다.

전체적인 실행시간은 1) 최하위 파티션에 해당하는 영역들 각각에서의 실행시간과 2)인접한 영역들의 쌍을 통합하는 횟수에 의해 결정된다. 영역들 각각에서의 실행시간은  $O(n^3)$ 을 선택 할 수 있고, 인접한 영역들의 쌍을 통합하는 횟수는 분할선의 개수와 같다. 1) 최하위 파티션에 해당하는 영역에서의 실행시간은 다음과 같다.  $2^d$ 개의 사각형이 최악의 경우 각각  $O(n^3)$  소요하므로, 병렬처리 시 실행시간은  $O(n^3)$ 이다. 순차적으로 처리하는 경우에는  $2^d O(n^3)$ 소요된다. 여기서, 주어진 노드의 개수보다 최하위 분할영역이 많은 경우는 의미가 없으므로  $2^d \leq n$ 라고 할 수 있다. 따라서 순차적으로 처리하는 경우의 실행시간은 최악의 경우  $O(n^4)$ 이다. 2) 인접한 영역들의 쌍을 통합하는 횟수는 다음과 같다.

$$\sum_{i=0}^{d-1} 2^i = 2^d - 1 \text{ 개의 직사각형을 대상으로 통합된}$$

light구조를 구해야 하는데, 통합된 light구조를 구하기 위해서는 상호 인접한 light구조를 확인하기만하면 되므로 각각의 경우  $O(1)$ 시간이 소요된다. 결국 병렬처리 시 최대  $O(d)$ 시간이 소요되며, 순차처리 시에는  $(2^d - 1)O(1)$ 시간이 소요되므로, 실행시간은  $O(1)$ 이라고 볼 수 있다. 그러나 최악의 경우  $2^d \leq n$ 임을 고려한다면,  $O(n)$ 시간이 된다.

결국 전체적인 실행시간은 최악의 경우에 대한 1) 과 2)의 실행시간을 합하여  $O(n^4) + O(n)$ , 즉,  $O(n^4)$ 이 된다.

#### IV. 오차정리의 증명

보조정리6: Euclidean 평면상의 주어진 점들의 집합을  $P$ 라 할 때,  $P$ 를 둘러싸는 최소한의 convex 영역을  $P_A$ 라고 하자.  $P_A$  외부에 임의의 점  $q$ 를 추가 할

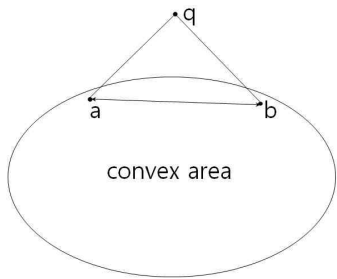


그림 4.  $P_A$  상의 TSP  
Fig. 4. TSP over  $P_A$ .

때,  $PU\{q\}$ 에 대한 TSP는  $P$ 에 대한 TSP보다 긴 길이를 가진다.

*proof*:  $PU\{q\}$ 에 대한 TSP를 구축하기 위하여,  $q$ 는  $P$ 의 임의의 두 개의 점  $a, b$ 와 연결되는 선분  $\overline{qa}, \overline{qb}$ 를 추가한다(그림 4). 이러한 추가선분 길이의 합이,  $P$ 에 대한 TSP상의  $a, b$ 사이의 경로의 길이보다 작거나 같은 경우,  $PU\{q\}$ 에 대한 TSP는  $P$ 에 대한 TSP보다 작거나 같은 거리를 가질 수 있다. 하지만, 이는 Euclidean 평면상의 triangular inequality에 모순된다. ■

그림 5는 하나의 파티션 영역에 생성된 ratio 1.5 근사TSP에 대한 분할선 및 분할선 주변의 교차 상황을 일반화한 그림이다. 분할선 주변에서 다수의 교차가 발생할 수 있는 상황을 나타낸다. 구름표시의 점선 내부에는 ratio 1.5 근사TSP가 구성되어 있으며, 교차되는 간선 및 이에 접한 노드만 표시된 그림이다.

길이  $L$ 인 두 개의 맞닿은 선분  $l, r$ 가 같은 길이의 분할선과 포개어진 채로 근사TSP에 추가되었다고 가정한다. 근사TSP를 분할선 좌우로 나누었다고 할 때,  $l$ 과 분할된 좌측의 근사TSP의 합집합에 해당하는 구조체를 *cut TSP*이라고 한다.

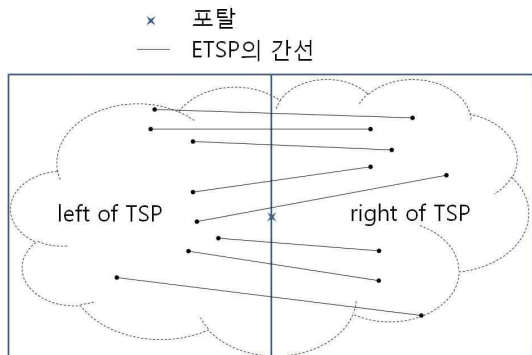


그림 5. 분할선 상의 교차  
Fig. 5. Crossings over separation line.

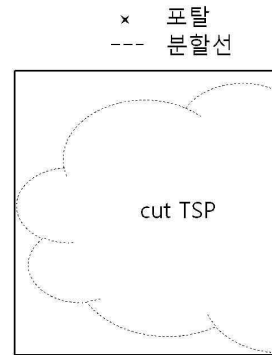


그림 6. 분할된 TSP  
Fig. 6. Cut TSP.

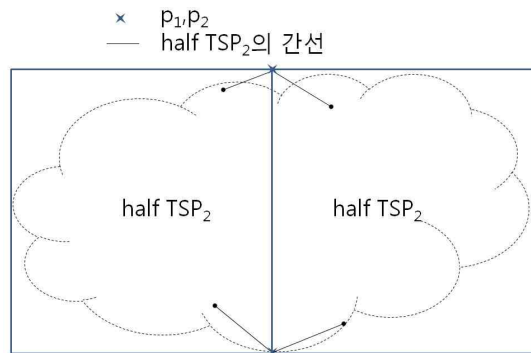


그림 7. 이분된  $TSP_2$   
Fig. 7.  $half TSP_2$ .

분할선의 양끝 점을 각각  $p_1, p_2$ 라 하자. 분할선 좌측 영역의 주어진 노드들에  $p_1, p_2$ 를 추가한 노드들의 집합,  $P_2$ 를 대상으로 구축한 근사TSP를  $half TSP_2$ 라 하자. 그림 7은 구름모습의 점선으로 표현된 두 개의  $half TSP_2$ 를 나타낸다.  $p_1, p_2$ 에 접하는 간선과 노드만 표시하였다.

노드 개수(밀도)가 높아 노드간의 거리가 가까울 때  $Length(half TSP_2)$ 는  $Length(cut TSP)$ 로 근사한다. 노드간의 거리가 가까워지면서,  $half TSP_2$ 에서 분할선과 맞닿은 우측부분이 직선에 가까워지기 때문이다. 분할선 좌측 영역의 주어진 노드들에 분할선 중앙의 포털을 추가하여 만들어진 노드들의 집합,  $P_1$ 을 대상으로 구축한 근사TSP를  $half TSP_1$ 라 하자 (그림2 참조).

*Proof*:(오차정리의 증명)

보조정리6:에 의해,

$$Length(half TSP_1) \leq Length(half TSP_2)$$

이다.  $p_1, p_2$ 는  $P_1$  영역(convex) 외부에 정의된 점들이기 때문이다. 그러므로 노드간의 거리가 0에 접근할 때,

$$\text{Length}(\text{half TSP}_1) \leq \text{Length}(\text{cut TSP})$$

이다. 분할선을 기준으로 좌측의  $\text{cut TSP}$ 와 우측의  $\text{cut TSP}$ 를 만들 수 있는데,  $\text{cut TSP}$ 의 정의에 의해, 그 둘을 합한 전체길이는 근사TSP의 길이에  $2L$ 을 더한 것이다.

$$2\text{Length}(\text{cut TSP}) = \text{Length}(\text{근사 TSP}) + 2L$$

그러므로 좌측과 우측의  $\text{half TSP}_1$ 길이의 총합은 근사TSP의 길이에  $2L$ 을 더한 것보다 작거나 같다.

$$2\text{Length}(\text{half TSP}_1) \leq \text{Length}(\text{근사 TSP}) + 2L$$

따라서 분할선을 기준으로 light구조를 통합할 때 추가될 수 있는 길이는  $2L$ , 분할선 길이의 두 배이다. ■

## V. 결 론

센서네트워크를 구성하고 있는 노드들을 대상으로 Euclidean TSP를 구성하려는 경우, 영역을 나누어 각 파티션에서 ETSP를 구성하고 각각의 ETSP를 접합하면서 전체 영역에 대한 근사-ETSP를 구성하는 방법에 대해 분석하였다. 이러한 방법에 의한 알고리즘은 영역 별로 병렬처리 하도록 변경하여 실행시간을 단축시킬 수 있다. 이렇게 영역별로 구성될 단계별 결과치를 접합하는 과정에서 추가 경로가 발생하는데, 이로 인한 전체적인 길이의 증가는 결국 센서노드 사이의 통신비용으로 작용하게 된다. 오차정리는 이러한 파티션 방법에 의해 발생할 수 있는 오차를 예상할 수 있도록 함으로써, 주어진 네트워크의 상황에 따른 해당 알고리즘 사용에 대한 타당성을 판단 할 수 있게 한다. 다음 단계의 연구는 오차정리 및 관련기법을 기반으로 하여 구성될 수 있는 알고리즘들을 보다 정확히 분석할 수 있도록 하는 것이다. 알고리즘 또는 해당하는 응용이 가지는 기하학적인 특성을 반영하여, 비용증가분이 더 줄어들 수 있는 여지를 찾아 분석에 적용하여야 한다.

## 참 고 문 헌

[1] Atul Verma, Hemjit Sawant, Jindong Tan, "Selection and Navigation of Mobile Sensor

Nodes Using a Sensor Network," Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, p.41-50, March 08-12, 2005.

- [2] 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.
- [3] 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.
- [4] 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.
- [5] J. Park and C. Choi, "Bayes Stopping Rule for MAC Scheme in Wireless Sensor Networks," Journal of IEEK, vol. 45-TC, no. 7, pp.53-61, July 2008.
- [6] J. Park, J. Ha and C. Choi, "Bayesian Cognizance of RFID Tags," Journal of IEEK, vol. 46-TC, no. 5, pp.524-531, May 2009.
- [7] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, Report 388, 1976.
- [8] Robert Sedgewick and Kevin Wayne, "Minimum Spanning Tree lecture notes," Computer Science, Princeton University, 226: Algorithms & Data Structures, Spring 2007.

## 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 컴퓨터학부 부교수

<주관심분야 : Approximations for NP-hard problems>