도로망이 설치된 L_1 평면에서의 최단경로 문제

배 상 원*, 좌 경 룡

한국과학기술원 전자전산학과 전산학전공

{swbae,kychwa}@jupiter.kaist.ac.kr

Shortest Paths on the L_1 Plane with a Transportation Network

Sang Won Bae*, Kyung-Yong Chwa

Division of Computer Science, Department of Electrical Engineering and Computer Science

Korea Advanced Institute of Science and Technology

요 약

본 논문에서는 L_1 평면상에 도로망이 주어져 있어서 여행자들이 그 도로들을 이용하여 더욱 빠르게 이동할 수 있는 가정하에서 가장 기초적인 기하 문제중에 하나인 두 점 사이의 최단 경로를 찾는 문제를 다룬다. 이 때, 두 점 사이의 거리는 L_1 거리가 아닌 주어진 도로들을 이용하여 두 점 사이를 이동할 때 필요한 최소시간으로 측정한다. 단순한 평면상에서의 최단경로와는 달리 도로망이 설치되어 있는 경우는 그것을 해결하기가 일반적으로 쉽지 않다. 본 논문에서는 도로망이 있는 평면에 대한 깊은 관찰과 이해를 통해 도로망이 설치되어 있는 L_1 평면상에서의 최단경로 문제를 해결하는 효율적인 알고리즘을 제시한다. 덧붙여, 본 논문에서 제시하는 문제 해결 방법은 L_1 평면 뿐만 아니라 유클리드 평면에도 어렵지 않게 적용할 수 있으며 보로노이 다이어그램으로의 일반화도 간단하다.

1 서론

뉴욕의 맨하탄 혹은 시카고와 같이 계획적으로 발전된 도시를 상상해보자. 높은 빌딩들이 빼곡히 들어차 있고 그 사이로 남북방향 혹은 동서방향의 거리가 놓여 있다. 도시에는 이 거리들을 따라서 버스나 지하철 혹은 택시와 같은 교통망이 설치 되어있다. 이런 도시에서 생활하는 사람들은 보통 시간을 매우 중요하게 생각하기 때문에 다른 곳으로 이동할 때에는 빠른 시간 안에 도착할 수 있는 경로를 통해 이동하기를 원한다.

본 논문에서는 위와 같은 설정에서 두 점 사이의 최단경로를 찾는 효율적인 알고리즘을 제시한다. 관련된 이전 결과로서 Aichholzer 등 [1]은 본 논문과 같은 가정 하에서 시작점을 고정하였을 때에 $O(n^2 \log n)$ 시간 안에 최단경로 맵(shortest path map)을 계산하는 방법을 제시하였다. 또, 배상원과 좌경룡 [2]은 유클리드 평면상에서 더 일반화된 도로망이 주어졌을 때에 $O(n^3 \log n)$ 시간 안에 최단경로 맵을 만드는 방법을 제시하였다. 여기서 n은 주어진 도로망의 복잡도이다. 위의 두 결과는 모두 보로노이 다이어그램(Voronoi diagram)을 계산하는 알고리즘으로부터의 부수적인 결과이다. 본 논문에서는 L_1 평면 위에 속도 v를 가지는 도로망이 주어질 경우, 두 점 사이의 최단경로를 $O(n^2)$ 시간 안에 계산하는 알고리즘을 제시한다.

본론을 논하기 전에 필요한 몇 가지 기호와 용어를 정의하겠다. 먼저 **도로망**은 n개의 교차하지 않는 도로들로 이루어져 있으며, **도로**는 수직 또는 수평 방향의 선분이고 각 도로는 **노드**(node)를 공유할 수 있으며 일정한 속도 v를 제공한다. 따라서 도로망 G는 평면 직선 그래프(PSLG; planar straight line graph)의 형태로 주어진다고 가정한다. 즉, V를 노드의 집합, E를 도로들의 집합이라 할 때에, $G = (V, E)$이다. 두 점 사이를 이 도로망을 이용하여 갈 수 있는 최단 시간을 새로운 거리 d_G로 정의하며 이에 상응하는 경로를 **최단경로**(shortest path)라 한다.

2 최단 경로 찾기

이번 장에서는 도로망 $G = (V, E)$가 설치되어 있는 L_1 평면상에서의 두 점사이의 최단경로의 성질에 대해 알아보며 그 성질들을 이용하여 최단경로를 계산하는 효율적인 알고리즘

을 설계한다.

도로가 여러 개인 일반적인 경우를 고려하기 전에 우선 도로가 한 개 있는 기본적인 경우를 생각해보자. L_1 평면에 무한히 긴 도로가 한 개 있는 경우에 최단경로에 대해서는 Abellanas 등 [4]이 그 성질을 밝힌 바 있다. 그 결과에 덧붙여 약간의 관찰을 통해 우리는 다음의 사실을 쉽게 얻을 수 있다.

고 찰 1. *수평인 도로가 하나 주어졌을 때에 한 점 p로부터 다른 점 q까지 최대한 빠르게 도달하기 위해서 도로를 이용해야 한다면,*

1. *두 점 p, q 모두 도로의 위쪽이나 아래쪽에 위치한다면, p로부터 수직으로 도로에 도달한 후, 적당한 위치까지 도로를 따라간 후 수직으로 q로 이동한다 [4].*
2. *만약, 둘 중에 하나 이상이 위의 조건을 만족하지 못한다면 조건을 만족하지 못하는 점에서부터 가장 가까운 도로의 노드까지의 최단 L_1 경로로 도달한다.*

이로써 우리는 두 점과 하나의 도로가 주어지면, 두 점사이의 최단경로를 쉽게 찾을 수 있게 되었다.

본 논문에서 가장 필수적인 정리를 기술하기 위해 몇 가지를 더 정의한다.

정 의 1. 어떤 경로 P가 다음의 조건을 만족할 때 우리는 그 경로 P를 **기초 경로**라 부르기로 한다.

1. P는 P의 양 끝점을 제외한 나머지 부분에서 V안의 어떤 노드도 포함하지 않는다.
2. P는 최대 한 개의 도로를 지나가며 고찰 1의 규칙을 따른다.

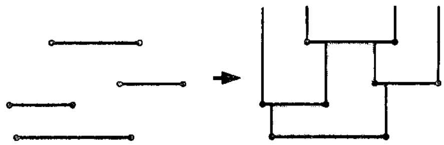
또한, 임의의 두 점 사이를 잇는 기초 경로들 중에 가장 짧은 길이를 갖는 경로를 **최단 기초 경로**라 부르기로 한다.

다음 정리는 본 논문에서 처음으로 밝힌 도로망이 설치된 평면에서의 최단경로에 대한 필수적인 성질이다.

정 리 2. *평면상에 도로망 $G = (V, E)$가 주어질 경우, 임의의 평면상의 두 점 p, q에 대하여 다음 성질을 만족하는 최단 경로 P가 존재한다; P는 끝점이 V에 속한 노드이거나 p 또는 q인 최단 기초 경로의 연속이다.*

위의 정리는 임의의 최단경로를 그 길이를 늘이지 않고 우리가 원하는 성질을 갖도록 변형할 수 있음을 보임으로서 증명이 가능하다.

정리 2 에 의해 우리는 어렵지 않게 최단경로를 찾는 알고리즘을 얻을 수 있다. 끝점이 노드 또는 p 또는 q인 최단 기초 경로의 연속인 최단경로가 항상 존재하기 때문에 모든 최단 기초 경로와 그 길이를 계산하면 도로망이 있는 경우의 최단 경로 문제를 단순한 가중 그래프 상에서의 최단경로 문제로 전환할 수 있다. 여기서 논의되어야 할 부분은 임의의 두 점 사이를 잇는 최단 기초 경로를 계산하는 부분이다.



그 림 1: $\mathcal{M}_1$의 생성

먼저, E_-는 수평 도로들의 집합, $E_|$는 수직 도로들의 집합이라고 정의한다. 물론, $E = E_- \cup E_|$이다. 네 개의 맵(map), $\mathcal{M}_1, \cdots, \mathcal{M}_4$를 다음과 같이 정의한다; 각 맵은 평면의 다각분할(polygonal subdivision)로 각 도로가 일정한 방향으로 움직일 때에 쓸고 지나간 영역을 그 도로의 영역이라고 표시한다. 이때, 움직이는 도로가 다른 도로의 영역에 도달하면 충돌하는 부분만큼만 진행을 멈춘다. $\mathcal{M}_1$과 $\mathcal{M}_2$는 E_-의 도로들을 각각 위쪽 방향과 아래쪽 방향으로 움직여서 만들고, $\mathcal{M}_3$과 $\mathcal{M}_4$는 $E_|$의 도로들을 각각 왼쪽과 오른쪽 방향으로 움직여서 만들어낸다. 그림 1를 참조하라. 참고로 각 맵에서 하나의 영역은 이름이 붙지 않는다. 이 과정은 plane sweep 방법을 이용하여 $O(n \log n)$ 시간 안에 완료할 수 있다.

이제 임의의 점 $x \in V \cup \{p, q\}$에 대해, 각 맵에 point locaiton을 수행하여 대응되는 도로가 무엇인지를 알아낸다. 이때, x가 영역의 경계에 있다면 그 경계를 공유하는 모든 영역과 그에 대응하는 도로를 알아낸다. $1 \leq i \leq 4$에 대하여, $E_i(x)$를 맵 $\mathcal{M}_i$에 대해 점 x를 point location하여 알아낸 도로들의 집합이라고 하자. $x \in V$는 $|E_i(x)| \leq 2$이고 $|E_i(p)|, |E_i(q)| \leq 3$이다.

마지막으로 w를 가중치로 하는 가중 그래프 $G_w = (V_w, E_w)$를 구성한다. 우선, G_w는 $V_w = V \cup \{p, q\}$인 완전 그래프이다. 각 간선에 대한 가중치는 다음과 같다. 각 $(i, j) \in \{(1,1), (1,2), (2,1), (2,2), (3,3), (3,4), (4,3), (4,4)\}$와 $x, y \in V_w$에 대해, $d_{i,j}(x, y)$를 두 점 x, y와 $E_i(x) \cap E_j(y)$안의 도로로 구해지는 기초 경로의 거리라 하자. 참고로 $|E_i(x) \cap E_j(y)| = 3$인 경우는 없으며, $|E_i(x) \cap E_j(y)| = 2$이면 x와 y 사이의 기초 경로는 꺾이지 않는다. $|E_i(x) \cap E_j(y)| = 0$이라면, $d_{i,j}(x, y) = \infty$로 한다. 또, 두 점사이의 L_1 거리를 d_1이라하면, $w(x, y) = \min\{\min_{i,j} d_{i,j}(x, y), d_1(x, y)\}$. 또한, 이때에 최소 거리로 선택된 i, j를 기억하면 최소 기초 경로를 이루는 도로를 알 수 있다.

위에 기술한 알고리즘은 이전 결과 [1]의 4.3절의 내용으로부터 다음의 알고리즘이 정확하게 두 점 사이의 최단 기초 경로를 찾는다는 것을 보일 수 있다. 가중 그래프가 주어질 경우에 최단경로 문제는 Dijkstra의 알고리즘 [3] 등을 이용하여 $O(n^2)$ 시간 안에 해결할 수 있다. 따라서 다음 결론에 이른다.

정 리 3. *L_1평면 상에 도로망 G가 주어졌을 때에, 임의의 두 점 사이의 최단 경로를 $O(n^2)$의 시간 안에 $O(n^2)$의 공간을 이용하여 계산할 수 있다.*

3 L_1 평면에서의 바늘과 최단경로 맵

"바늘(needle)" 은 도로망에서의 보로노이 다이어그램을 분석하기 위한 매우 적합한 도구이다 [2]. 배상원과 좌경룡은 유클리드 평면에서의 도로망을 분석하기 위해 바늘을 정의하였고 유클리드 평면에서 non-piercing한 바늘들의 집합에 대한 보로노이 다이어그램을 최적의 시간 안에 계산할 수 있음을 보였다.

이 장에서는 L_1평면에서 바늘들의 보로노이 다이어그램에 대해 논의한 후 이것을 이용하여 최단경로 맵을 구성할 수 있음을 보인다.

L_1 평면에서는 바늘의 방향이 수직 또는 수평인 것만이 의미가 있으므로 여기서는 그런 바늘들만을 다루기로 한다. 이 가정으로 임의의 평면상의 점으로부터 어떤 바늘까지의 거리를 계산할 수 있게 되며, 이것으로부터 두 바늘 사이의 거리의 이등분선을 계산할 수 있다. 여기에 non-piercing의 조건이 추가되면 바늘들의 보로노이 다이어그램이 abstract Voronoi diagram임을 증명할 수 있고 그것으로부터 다음을 유추할 수 있다.

정 리 4. *n개의 수직 또는 수평방향의 바늘들의 집합 S 안의 모든 바늘들이 서로 non-piercing하다면 그 보로노이 다이어그램은 $O(n \log n)$ 시간 안에 $O(n)$의 공간을 이용하여 계산할 수 있다.*

최단경로 맵은 출발점 p가 고정되어 있는 상태에서 도착점 q를 입력받았을 때에 빠르고 효율적으로 p로부터 q로 가는 최단경로를 제공하는 구조이다. L_1 평면상에 도로망 G와 출발점 p가 주어지면 다음과 같은 과정을 통해 최단경로 맵을 만들 수 있다.

먼저, 2 장에서 설명한 방법을 이용하여 $V_w = V \cup \{p\}$에 대한 가중치 그래프 $G_w = (V_w, E_w)$를 구성한다. 그런 후, 출발점 p에 대해 G_w에 최단 경로 알고리즘(예로 Dijkstra 알고리즘)을 적용한다. 그 결과로 우리는 p를 뿌리로 하는 최단경로 트리(shortest path tree) T를 얻게 된다. 이 최단경로 트리 T는 두 노드 사이의 최단 기초 경로에 대한 정보를 함께 가지고 있으므로 경로가 꺾이는 점들과 그 점들까지의 거리도 모두 알 수 있다. T를 순회하여 적당한 바늘들을 생성하면 그 바늘의 집합에 대한 보로노이 다이어그램이 곧 최단경로 맵이 된다. [1,2] 바늘들을 생성하는 방법은 다음과 같다.

최단경로 트리 T의 각 간선 l와 l에 인접하는 두 정점 w, y에 대하여, $d_G(p, w) < d_G(p, y)$라고 가정하자. 만약 간선 l이 어떤 도로 $e \in E$위에 위치할 때에만 w에서 시작하고 e의 양쪽 노드까지 향하는 바늘 두 개를 생성한다. 바늘 생성 과정이 끝나면 [2]의 5장에서 설명한 방법과 같이 바늘들을 적당히 잘라내어 non-piercing하도록 만든다. 자세한 방법은 [2]를 참조하기를 바란다.

위와 같은 과정에 의해 만들어진 바늘들의 집합을 $\mathcal{S}$라 하자. 기초 경로는 최대 두 번 꺾이므로 T의 복잡도는 $O(n)$이다. 이것은 $|\mathcal{S}| = O(n)$임을 의미한다. 또한, $\mathcal{S}$는 모두 수직 또는 수평 방향의 바늘들이며 non-piercing하므로 정리 4에 의해 L_1 평면에서의 그 보로노이 다이어그램 $\mathcal{V}_1(\mathcal{S})$은 $O(n \log n)$의 시간 안에 계산 가능하다. 여기에 도착점 q를 쿼리(query)하면 $\mathcal{V}_1(\mathcal{S})$에 point location을 하는 것으로 우리는 $p - q$의 최단 경로를 빠르게 보고할 수 있다.

한 가지 더 명확히 하여야 할 점은 $\mathcal{V}_1(\mathcal{S})$이 실제로 최단경로 맵의 기능을 수행하는가에 대한 의문인데 이것은 바늘의 특성으로 쉽게 보일 수 있으며 본 논문에서는 충분히 기술하지 못하지만 앞선 논문들 [1,2]에서 같은 논의가 이미 되어 있다.

따라서 다음의 결론을 도출할 수 있다.

정 리 5. *L_1 평면상에 도로망 G와 임의의 점 p가 주어지면, $O(n^2)$의 시간 안에 $O(n^2)$의 공간을 이용하여 최단경로 맵을 구성할 수 있으며 이 최단경로 맵은 도착점 q를 입력 받았을 때에 $O(\log n + r)$의 시간 안에 p에서 q로 가는 최단경로를 계산한다. 이 때, r은 최단경로의 복잡도 이며 $r \leq 3n$이다.*

4 결론

앞에서 우리는 L_1 평면에 수직 또는 수평인 도로로 이루어진 도로망이 주어졌을 경우에 최단경로 맵을 계산하는 방법을 보였다. 이것은 매우 간단하게 보로노이 다이어그램으로 일반화할 수 있다. 대강을 설명하면 주어진 각 site에 대해 최단경로 트리를 만들고 그것으로부터 바늘들을 생성하여 보로노이 다이어그램을 생성하는 것이다. m개의 site가 주어졌을 때에, 이것은 $O(nm(n+m))$의 시간 안에 가능하며 이전 결과 [1]의 복잡도 $O(n^2 \log n + m \log m)$과는 다른 양상을 보인다.

덧붙여, 본 논문에서 소개한 방법은 유클리드 평면에서도 적용이 가능하다. 그럴 경우, 유클리드 평면에 방향의 제한이 없고 도로의 속도를 개별적으로 줄 수 있는 경우에 $O(n^3)$의 시간과 $O(n^2)$의 공간으로 최단경로 맵을 구성할 수 있다.

참 고 문 헌

[1] O. Aichholzer, F. Aurenhammer, and B. Palop. Quickest paths, straight skeletons, and the city voronoi diagram. In *Proceedings of 18th SoCG*, pages 151–159, 2002.

[2] S. W. Bae and K.-Y. Chwa. Voronoi diagrams with a transportation network on the euclidean plane. In *Proceedings of ISAAC 2004*, to appear.

[3] E. W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math.*, 1:269–271, 1959.

[4] F. Hurtado, B. Palop, and V. Sacristán. Diagramas de voronoi con funciones temporales. *VIII Encuetos en Geometria Computacional*, 1999.