

차선거리유지 경로찾기

(Finding a Second Best Coverage Path)

나 현 숙 [†] 김 정 희 ^{**}
(Hyeon-Suk Na) (Junghee Kim)

요약 사용자가 가능한 센서 가까이에서 이동하는 문제를 최단거리유지 문제(Best Coverage Problem)라 하며, 무선 연결의 안정성을 높이기 위해서 두 개의 센서로부터 최대한 가까이 위치하며 움직이는 이동경로를 차선거리유지 경로(Second Best Coverage Path)라고 한다. 이 논문에서는 센서들의 집합 U 와 시작점 s , 끝점 t 가 주어질 때, s 에서 t 까지 이르는 모든 경로 중에서, 가장 가까운 두 개의 센서까지 거리 중 큰 값이 최소가 되는 차선거리유지 경로를 찾는 $O(n \log n)$ -시간 알고리즘을 제시하며, 차선거리유지 경로를 따라 움직일 때가 최단거리유지 경로를 따라 움직일 때 보다, 네트워크에의 연결이 보다 안정적인을 보여주는 실험결과도 또한 제시한다.

키워드 : 알고리즘, 계산기하학, 보로노이 다이어그램, 네트워크 유지범위

Abstract The best coverage problem is finding a path whose worst distance to the sensor-set is the best among all paths. Instead, if the user stays close to its nearest two sensors, then the stability of the wireless connection may be guaranteed. So we consider the problem of finding a second best coverage path: given the sensor set U , a starting point s and a target point t , find a path whose worst distance to the second closest sensor is minimized. This paper presents an $O(n \log n)$ -time algorithm to find such a path. We also give experimental evidence showing that the connection to the sensor-set along a second best coverage path is more stable than that along the best coverage path.

Key words : Algorithm, Computational Geometry, Voronoi Diagram, Network Coverage

1. 서론

센서 무선 네트워크 환경에서 사용자의 위치와 센서와의 거리는 중요한 요소로 작용한다. 거리와 센서의 전력소모는 비례하여 증가한다고 가정하며, 거리가 가까울수록 센서의 전력소모는 줄어들고, 보다 정확한 정보를 사용자에게 전달할 수 있다. 따라서, 센서의 전력소모를 가능한 줄이기 위해서, 사용자는 가능한 센서 가까이에서 움직이는 것이 필요하며, 센서끼리 정보를 전달할 때

에도 가능한 가까이 있는 센서에게 정보를 전달하여 보낸다면, 전력의 소모를 줄일 뿐 아니라, 전달되는 동안 정보의 손실이 적을 것이다.

이렇게 센서와의 거리를 가능한 최단거리로 유지하면서 네트워크 환경을 이동하는 문제를 최단거리유지 문제(Best Coverage Problem)라 한다. 즉, 가장 가까운 센서까지 거리의 최악의 값을 최소화하는 경로를 찾는 문제이다[1]. 이 문제는 Meguerdichian et al. [1]에서 처음 정의되었으며, 이들은 델로니 삼각화(Delaunay triangulation)의 에지들에 가중치를 주고 다익스트라(Dijkstra) 알고리즘을 이용해 최단거리유지 경로를 찾는 $\Theta(n \log n)$ -시간 알고리즘을 제시하였다. Li et al. [2]은 이 문제를 상대이웃그래프(Relative Neighbor Graph; RNG)를 이용해 분산처리방식으로 $\Theta(n \log n)$ 시간에 해결하였다. RNG는 델로니 삼각형의 부분집합으로 델로니 삼각형보다 더 적은 에지를 가지며, 분산처리방식으로 구성할 수 있다.

Meguerdichian et al.은 [1]에서 최장거리유지 문제(Worst Coverage Problem) 역시 제기했는데, 최단거리유지 문제와 반대로, 센서로부터 가장 가까운 거리를 최

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

[†] 종신회원 : 숭실대학교 컴퓨터학부 교수
hsnaa@ssu.ac.kr

^{**} 학생회원 : 숭실대학교 컴퓨터학부
junghee1013@gmail.com

논문접수 : 2008년 1월 11일

심사완료 : 2008년 5월 14일

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제35권 제8호(2008.8)

대화하는 경로를 찾는 문제이다. 이들은 이 문제를 사용하는 네트워크 환경에서 항상 센서 사이를 이동한다는 가정하에서 보로노이 다이어그램(Voronoi Diagram)을 이용하여 해결했다. 이후 Mehta et al. [3]는 사용자의 이동경로를 무한한 평면에서 찾았는데, 사용자가 반드시 센서 사이를 이동하지 않을 수도 있다. 이 논문에서 최장거리유지 경로는 각 센서의 전파범위를 벗어나는 최장거리를 구한 다음에 시작점과 도착점을 연결하는 최단경로를 찾는다. 따라서 이 때 이동경로는 항상 센서의 사이를 지날 필요가 없으므로, [1]에서 제시된 경로와 같거나 더 나은 최장거리를 갖는다.

이 논문에서는 무선 연결의 안정성을 높이기 위해서 두 개의 센서로부터 가까이 위치하며 움직이는 경로--차선거리유지 경로(Second Best Coverage Path)--를 처음으로 정의하며 이러한 경로를 찾는 $O(n \log n)$ -시간 알고리즘을 제시한다. 이 경로를 따라 움직인다면 연결되어 있는 센서가 갑자기 문제를 일으키더라도 또 다른 센서가 가까이에 있으므로 쉽게 재접속이 가능하게 된다. 또한 센서의 입장에서도 사용자와 연결하기 위해서 무리하게 전력을 올리지 않고 사용자와 접속할 수 있다.

2장에서는 차선거리유지 경로를 찾기 위해서 미리 알아두어야 할 개념들에 대해 소개하고, 3장과 4장에서 이 경로를 찾는 효율적인 알고리즘을 제시하고 합당성을 증명하며, 5장에서는 차선거리유지 경로를 따라 움직일 때가 최단거리유지 경로를 따라 움직일 때에 비해 네트워크의 연결이 보다 안정적일 수 있음을 보여주는 실험결과를 제시한다.

2. 예비지식

2.1 보로노이 다이어그램(Voronoi Diagram)과 델로니 삼각화(Delaunay Triangulation)

이차원 평면을 R^2 , 평면위의 두 점 p, q 간의 유클리드 거리를 $\|pq\|$ 로 표기한다. R^2 위의 점집합 $U = \{u_1, u_2, \dots, u_n\}$ 가 있을 때, u_i 를 사이트(site)라 부른다. 보로노이 다이어그램 $Vor(U)$ 는 사이트 집합 U 에 따라 평면을 나누는 것으로, 평면위의 점들을 그것에 가장 가까운 사이트에게 할당하여 평면을 분할한 경계선들의 집합이며[4], 이를 요약하면 다음과 같다.

- 사이트 u_i 의 보로노이 영역:

$$VR(u_i, U) = \{x \in R^2 \mid \forall j \neq i, \|xu_i\| \leq \|xu_j\|\}$$
- 보로노이 에지 $edge(u_i, u_j)$: 서로 이웃하는 두 보로노이 영역 $VR(u_i, U)$ 와 $VR(u_j, U)$ 의 교집합이며, 선분 $u_i u_j$ 의 수직이등분선이다. 보로노이 에지는 양 끝점이 보로노이 노드인 유한 보로노이 에지와 한 끝점은 보

로노이 노드이고 다른 한 끝점은 무한히 뻗어 나가는 무한 보로노이 에지로 구분된다.

- 보로노이 노드 $node(u_i, u_j, u_k)$: 서로 이웃하는 세 보로노이 영역 $VR(u_i, U)$, $VR(u_j, U)$, $VR(u_k, U)$ 의 교점이다. $node(u_i, u_j, u_k)$ 에서 세 사이트 u_i, u_j, u_k 까지의 거리는 모두 같음을 유의하자.
- $N(u_i)$: 사이트 u_i 에 이웃하는 사이트들의 집합, 즉 $\{u_j \in U \mid VR(u_i, U) \cap VR(u_j, U) \neq \emptyset\}$ 이다.

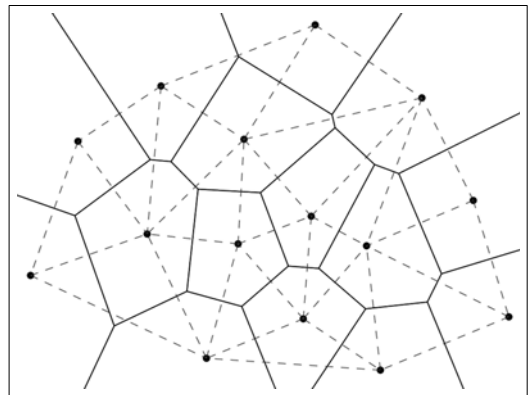


그림 1 보로노이 다이어그램과 델로니 삼각형

사이트의 개수가 n 개일 때, 보로노이 에지는 최대 $3n-6$ 개이며, 보로노이 노드는 최대 $2n-5$ 개이다. 보로노이 다이어그램을 만드는 알고리즘으로는 분할정복법, Fortune's plain sweep 알고리즘 등 여러 가지가 있으며, 최악의 경우 시간복잡도는 $\Theta(n \log n)$ 이다[5]. 보로노이 에지 $edge(u_i, u_j)$ 가 존재하는 u_i 와 u_j 를 직선으로 연결하여 얻는 기하구조를 델로니 삼각화(Delaunay triangulation)라 하며[5], $Del(U)$ 라 표기한다. U 의 어느 네 사이트도 동일한 원 위에 놓여 있지 않음을 가정할 때, $Del(U)$ 는 삼각형들만으로 이루어진 기하구조가 된다.

2.2 다익스트라(Dijkstra) 알고리즘

가중치 그래프(weighted graph)에서 시작노드 s 에서 도착노드 t 까지 에지를 따라 움직일 때, 지나간 에지들 가중치의 합이 최소가 되는 경로를 최단거리경로(Shortest path)라 한다. 최단거리경로를 찾는 알고리즘으로 가장 널리 알려진 다익스트라 알고리즘(Dijkstra Algorithm)[6]은 표 1과 같다.

그래프 노드가 n 개, 에지가 e 개 있을 때 $O((n+e) \log n)$ 의 시간이 필요하다. 다익스트라 알고리즘은 시작점과 연결되어있는 노드의 집합 S 와 아직 연결되어 있지 않

표 1 다익스트라 알고리즘과 RELAX 함수

```

Dijkstra( $G, s, t$ ) {
  INITIALIZE( $G, s$ )
   $d[s] = 0$ ;
   $Q \leftarrow V[G]$ 
  while ( $Q \neq \emptyset$ )
     $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
     $S \leftarrow S \cup \{u\}$ 
    for each vertex  $v \in \text{Adj}[u]$ 
      RELAX( $u, v$ )
  Return  $d[t]$ 
}

RELAX( $u, v$ ) {
  If  $d[v] > d[u] + w(u, v)$ 
     $d[v] \leftarrow d[u] + w(u, v)$ 
}
    
```

는 노드의 집합 T 로 구분한다. T 의 원소 중에서 S 의 원소 v 에 연결되어 있으면서 가중치가 최소인 에지 (v, u) 를 찾아서, u 를 T 에서 제거하고 S 에 추가한다. 그리고 이때, v 의 가중치와 u 의 가중치의 합이 s 에서 u 까지의 최단경로가 된다. 이 논문에서는 다익스트라 알고리즘의 RELAX 함수를 수정하여, 지나간 에지들의 가중치의 “합” 대신 “최대값”을 최소로 하는 경로를 찾는 방법을 이용한다.

3. 차선거리유지 경로 알고리즘

3.1 문제 정의

주어진 점집합 A 와 B 에 대해서, A 에서 B 까지 차선거리 (Second Best Coverage Distance)는 $\text{dist}(A, B) = \max_{a \in A} (\|ab_1\|, \|ab_2\|)$ 로 정의한다. 여기서 $b_1, b_2 \in B$ 이며, b_1 은 a 와 가장 가까운 점, b_2 는 a 와 두 번째로 가까운 점이다. 시작점 s 와 도착점 t , 그리고 센서 혹은 사이트의 집합 U 가 주어졌을 때, s 에서 t 까지 차선거리는 $L(U, s, t) = \min_{\Pi: s \text{에서 } t \text{까지 경로}} \text{dist}(\Pi, U)$ 와 같이 정의될 수 있고, 이 때 $L(U, s, t) = \text{dist}(\Pi, U)$ 인 이동경로 Π 를 차선거리유지 경로 (Second Best Coverage Path)라고 정의한다.

3.2 차선거리유지 경로 알고리즘

```

ConstructGraph( $U$ )

INPUT : 센서집합  $U$ 
OUTPUT: 가중치 비방향 그래프  $G$ 
    
```

```

Step 1: 보로노이 다이어그램  $Vor(U)$ 를 구하고,  $G := Vor(U) \cap CH(U)$ 로 놓는다. 보로노이 에지의 가중치는 에지의 양 끝점과 사이트와의 거리 중 먼 거리로 놓는다.

Step 2:  $U$ 의 모든 사이트  $u_i$ 에 대해서  $N(u_i)$ 를 찾고 다음을 실행한다.
     $\Rightarrow N(u_i)$ 의 임의의 사이트  $v$ 에 대해서,  $\text{edge}(u_i, v) \cap \overline{u_i v} \neq \emptyset$  이 면,  $\text{edge}(u_i, v) = \overline{qq'}$ ,  $c := \overline{qq'} \cap \overline{u_i v}$ 라 표기할 때,  $G := G - \{\overline{qq'}\} \cup \{\overline{qc}, \overline{cq'}\}$ 로 놓고, 에지  $\overline{qc}$ 의 가중치는  $\|qv\|$ , 에지  $\overline{cq'}$ 의 가중치는  $\|q'v\|$ 로 설정한다.
     $\Rightarrow Del(N(u_i))$ 를 구하고,  $Del(N(u_i))$ 의 임의의  $\text{edge } \overline{vw}$ 에 대해  $\overline{vw} \cap VR(u_i, U) \neq \emptyset$  이면,  $\text{edge}(u_i, v)$ 를  $\overline{q_1 q_1'}$ ,  $\text{edge}(u_i, w)$ 를  $\overline{q_2 q_2'}$ , 이들이  $\overline{vw}$ 와 만나는 점을  $c_1$ 과  $c_2$ 라 표기할 때,  $G := G - \{\overline{q_1 q_1'}, \overline{q_2 q_2'}\} \cup \{\overline{q_1 c_1}, \overline{c_1 q_1'}, \overline{q_2 c_2}, \overline{c_2 q_2'}\}$ 로 놓는다. 에지  $\overline{q_1 c_1}$ 의 가중치는  $\max(\|q_1 v\|, \|c_1 v\|)$ , 에지  $\overline{c_1 q_1'}$ 의 가중치는  $\max(\|c_1 v\|, \|q_1' v\|)$ ,  $\overline{q_2 c_2}$ 의 가중치는  $\max(\|q_2 w\|, \|c_2 w\|)$ , 에지  $\overline{c_2 q_2'}$ 의 가중치는  $\max(\|c_2 w\|, \|q_2' w\|)$ , 에지  $\overline{c_1 c_2}$ 의 가중치는  $\frac{1}{2} \|vw\|$ 로 설정한다.

SecondBestCoveragePath( $G, s, t$ )

INPUT : 그래프  $G$ 와  $G$ 의 노트  $s, t$  OUTPUT:  $s$ 로부터  $t$ 까지  $U$ 에 대한 차선거리유지 경로

Return  $\Phi := \text{변형 Dijkstra}(G, s, t)$ 
    
```

3.3 알고리즘 설명

3.3.1 ConstructGraph(U)

Step 1에서는 $G := Vor(U) \cap CH(U)$ 를 계산한다. 즉, 그래프 G 의 노드들은 $Vor(U)$ 의 보로노이 노드들과 무한 보로노이 에지들이 $CH(U)$ 에 의해 잘려져 생긴 노드들로 이루어지며, G 의 에지들은 $Vor(U)$ 의 유한 보로노이 에지들과 무한 보로노이 에지들이 $CH(U)$ 에 의해 잘려지고 남은 부분들로 이루어진다. (그림 2 참고)

Step 2에 의해서 그래프를 더 세분한다. 모든 사이트

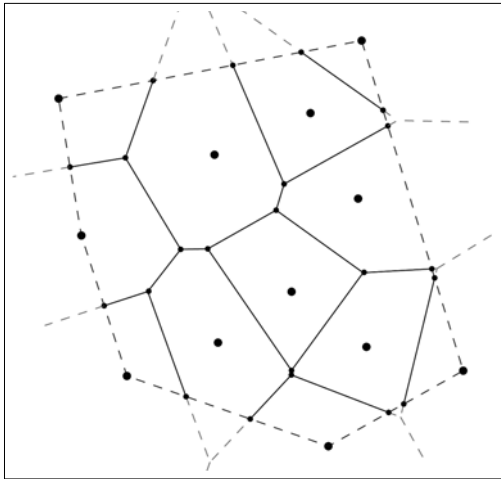


그림 2 Step 1을 마친 후의 그래프 G

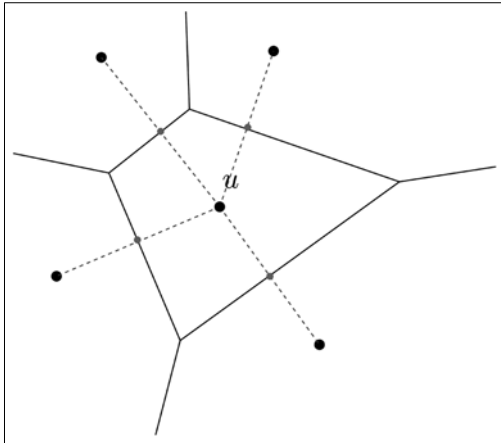


그림 3 u와 N(u)의 연결선의 교점을 추가

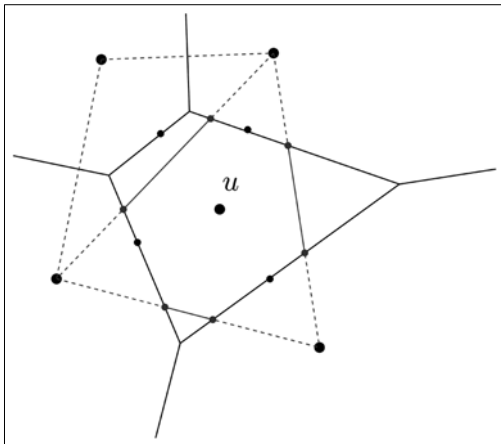


그림 4 보로노이 영역 내부에 유효한 이웃센서 연결선 추가

u_i 에 대해서 이웃 사이트 $N(u_i)$ 를 찾고 u_i 와 $N(u_i)$ 의 원소 v 를 직선으로 연결하여 $edge(u_i, v)$ 와 한 점에서 만난다면, $edge(u_i, v)$ 위의 교점 c 를 G 의 노드로서 추가시키고, 파생되는 에지들의 가중치를 업데이트한다. (그림 3 참고) 또한, $Del(N(u_i))$ 를 구하고, $Del(N(u_i))$ 의 임의의 edge \overline{vw} 에 대해 $\overline{vw} \cap VR(u_i, U) \neq \emptyset$, $\overline{vw} \cap VR(u_i, U)$ 을 그래프 G 에 추가시키고, 이에 의해 분할된 에지들의 가중치를 업데이트한다. (그림 4 참고) 앞으로는, 주어진 가중치 비방향 그래프 G 위의 임의의 경로 Π 에 대하여, Π 가 지나간 에지들의 가중치 최대값을 $\max(\Pi)$ 라고 표기하자.

보조정리 1. G 위의 임의의 경로 Π 에 대해, $dist(\Pi, U) = \max(\Pi)$ 을 만족한다.

(증명) Step 1에서는 센서의 집합 U 의 보로노이 다이어그램을 계산하고, 무한 보로노이 에지는 $CH(U)$ 에 의해 잘라 유한한 에지로 고친 후, 각 에지의 가중치를 에지의 양 끝점과 사이트와의 거리 중 먼 거리로 놓았다. $Vor(U) \cap CH(U)$ 의 임의의 에지 e 에 대해, e 위의 점에서 센서의 집합 U 까지의 차선거리는 e 를 정의한 두 사이트까지의 거리에 의해 결정되므로, $dist(e, U)$ 는 e 의 양 끝점에서 그 두 사이트까지의 거리 중 먼 거리이다. Step 1에서는 e 의 가중치를 이 거리 값으로 놓았으므로, $dist(e, U) = \max(e)$ 이다. G 위의 임의의 경로 Π 에 대해, $dist(\Pi, U) = \max_{e \in \Pi} dist(e, U)$ 이고, $dist(e, U) = \max(e)$ 이므로 $dist(\Pi, U) = \max_{e \in \Pi} \max(e) = \max(\Pi)$ 이다. (증명 끝)

3.3.2 SecondBestCoveragePath(G, s, t)

변형 Dijkstra 알고리즘은 표 1에 나타난 다익스트라 알고리즘의 RELAX 함수를 다음과 같이 변형한 것이다. 이에 따라, 변형 Dijkstra(G, s, t)는 s 로부터 t 까지 도달하는 G 위의 경로들 중, 지나가는 에지들의 “최대 가중치 값이 가장 작은” 경로 Φ 를 찾으며, $dist(\Phi, U)$ 는 최적의 값을 알 수 있다.

```

변형 RELAX( $u, v$ )
    If  $d[v] > \max(d[u], w(u, v))$ 
        $d[v] \leftarrow \max(d[u], w(u, v))$ 
    
```

보조정리 2. $G := ConstructGraph(U)$ 위의 임의의 노드 s, t 에 대해서, $\Phi := SecondBestCoveragePath(G, s, t)$ 라 하자. 그러면, s 로부터 t 까지 도달하는 G 위의 임의의 경로 Π 에 대해서 $dist(\Phi, U) \leq dist(\Pi, U)$ 를 만족한다.

(증명) $\Phi := SecondBestCoveragePath(G, s, t)$ 는 변

형 $Dijkstra(G, s, t)$ 가 리턴하는 경로이므로, 당연히 s 로부터 t 까지 도달하는 G 위의 임의의 경로 Π 에 대해서 $\max(\Phi) \leq \max(\Pi)$ 이다. 한편, 보조정리 1에 의하여 G 위의 임의의 경로 Π 에 대해 $dist(\Pi, U) = \max(\Pi)$ 을 만족한다. 따라서 $dist(\Phi, U) \leq dist(\Pi, U)$ 이다. (증명 끝)

4. 알고리즘의 합당성 및 시간복잡도

4.1 알고리즘의 합당성

이 절에서 우리는 알고리즘 $SecondBestCoveragePath(G, s, t)$ 가 찾은 이동경로 Φ 가 $L(U, s, t) = dist(\Phi, U)$ 를 만족함을 보인다. 실제로 우리는 s 에서 t 까지 도달하는 평면위의 임의의 경로 Π 에 대해, $VR(u, U) \cap \Pi \neq \emptyset$ 인 모든 사이트 u 와 $VR(u, U) \cap \Pi$ 에 속하는 모든 컴포넌트 Π_u 에 대해서, $dist(\Pi_u', U) \leq dist(\Pi_u, U)$ 인 Π_u' 이 G 위의 경로로서 존재함을 보일 것이다. 이는 s 에서 t 까지 평면위의 임의의 경로 Π 에 대해서, $dist(\Pi', U) \leq dist(\Pi, U)$ 인 경로 Π' 이 G 안에 경로로서 존재함을 의미하며, 그러면 보조정리 2에 의해 $SecondBestCoveragePath(G, s, t)$ 에서 찾은 경로 Φ 가 $dist(\Phi, U) \leq dist(\Pi', U)$ 을 만족하므로, $dist(\Phi, U) \leq dist(\Pi', U) \leq dist(\Pi, U)$ 를 만족한다. 결론적으로, s 에서 t 까지 도달하는 평면위의 임의의 경로 Π 에 대해, $dist(\Phi, U) \leq dist(\Pi, U)$ 이므로, $L(U, s, t) = dist(\Phi, U)$ 을 만족한다.

정리 1. s 에서 t 까지 도달하는 평면위의 임의의 경로를 Π 라 하자. 그러면 $VR(u, U) \cap \Pi \neq \emptyset$ 인 임의의 u 와 $VR(u, U) \cap \Pi$ 의 임의의 컴포넌트 Π_u 에 대해서, $dist(\Pi_u', U) \leq dist(\Pi_u, U)$ 인 G 위의 경로 Π_u' 이 존재한다.

(증명) $\Pi_u \subseteq VR(u, U) \cap \Pi$ 의 시작점과 도착점을 I 와 F 라 하자. U 를 사이트 집합으로 한 보로노이 다이어그램 $Vor(U)$ 와 $U - \{u\}$ 를 사이트 집합으로 한 보로노이 다이어그램 $Vor(U - \{u\})$ 를 생각해 보자. $VR(u, U)$ 와 $Vor(U - \{u\})$ 을 겹쳐보면, $VR(u, U)$ 은 그림 5와 같이 $Vor(U - \{u\})$ 에 의해 분할된다. 따라서 U 의 사이트들 중, Π_u 에 가장 가까운 사이트는 u 이고, 두 번째로 가까운 사이트는 $N(u)$ 의 원소 중 하나이므로, Π_u 의 U 에 대한 차선거리는 $N(u)$ 의 사이트들 중에서 가장 가까운 사이트에 의해서 결정된다.

(1) 어떤 사이트 $v \in N(u)$ 에 대하여, $\Pi_u \subseteq VR(v, U - \{u\})$ 인 경우 (그림 6): 이 경우, Π_u 의 시작점 I 와 도착점 F 는 같은 에지 $edge(u, v)$ 에 속한다. $\overline{uv} \cap edge(u, v) \neq \emptyset$ 이면, $q_0 = \overline{uv} \cap edge(u, v)$ 이라 놓고, $\overline{uv} \cap edge(u, v) = \emptyset$ 이면, 선분 \overline{uv} 의 중점에 I 나 F 보다

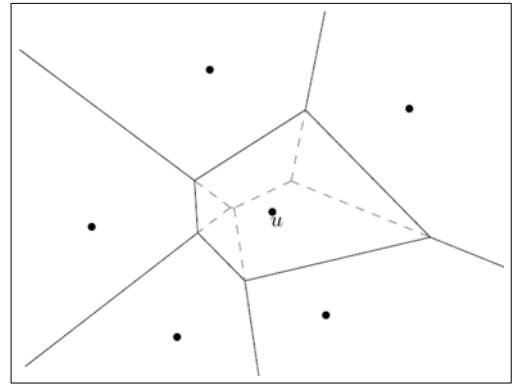


그림 5 $Vor(U - \{u\})$ (점선)에 의해, $VR(u, U)$ (실선)가 $N(u)$ 의 보로노이 영역들 $VR(v, U - \{u\})$ 로 나누어진 모습

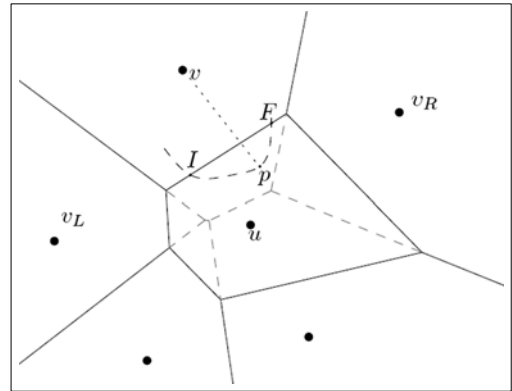


그림 6 $\Pi_u \subseteq VR(v, U - \{u\})$ 인 경우

가까운 $edge(u, v)$ 위의 보로노이 노드 q_0 가 존재한다. 첫 번째 경우는 알고리즘 $ConstructGraph(U)$ 의 Step 2에 의해서, 두 번째 경우는 알고리즘 $ConstructGraph(U)$ 의 Step 1에 의해서, q_0 가 이미 G 안에 노드로서 삽입되어 있음을 알 수 있다. 이제 $\Pi_u' := \{q_0\}$ 로 놓으면 Π_u' 은 한 점만을 포함하는 G 위의 경로이며 $dist(\Pi_u', U) = \|q_0, u\| = \max(\|I, u\|, \|F, u\|) = dist(\Pi_u, U)$ 보다 작거나 같다.

(2) Π_u 가 $N(u)$ 의 여러 원소의 영역을 지나는 경우 (그림 7): I 를 출발한 Π_u 가 교차하는 $Vor(U - \{u\})$ 의 보로노이 에지들을 차례로 $edge(v_i, v_{i+1})$ ($1 \leq i \leq m-1$)와 같이 번호 매기고, 이 교점들을 p_i ($1 \leq i \leq m-1$)라 하자. 교차점(self-intersection)을 갖는 경로는 교차점에서 생긴 루프를 제거하고 만든 단일경로(simple path)보다 U 까지의 차선거리가 더 크거나 같으므로 Π_u 를 단일경로라고 가정해도 무방하며, 각 보로노이 에지를 두

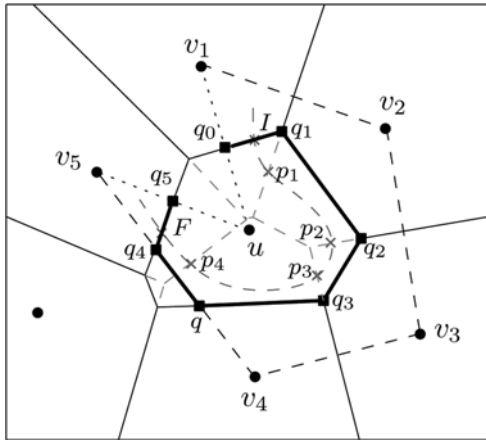


그림 7 Π_i (실선으로 된 곡선)가 $VR(v_1, U - \{u\}), \dots, VR(v_m, U - \{u\})$ 를 지나는 경우, q_0 에서 q_5 까지 도달하는 G 위의 경로 Φ (굵은 선)

번 이상 지나지 않는다고 가정해도 무방하다. (하나의 보로노이 에지를 두 번 이상 지난다면, 가장 끝에 위치한 교차점 사이의 보로노이 에지로 경로를 대체하여 차선거리가 더 작거나 같은 경로로 바꿀 수 있다.) $Vor(U)$ 에서 볼 때, Π_u 의 시작점 $I = p_0$ 와 종점 $F = p_m$ 은 보로노이 에지 $edge(u, v_1)$ 와 $edge(u, v_m)$ 위에 존재하고, p_1, \dots, p_{m-1} 는 보로노이 영역 $VR(u, U)$ 내부에 놓여 있으며, Π_u 의 p_i 에서 p_{i+1} 까지의 경로($0 \leq i \leq m-1$)는 v_{i+1} 까지의 거리에 의해서 차선거리가 결정된다.

(1)에서 I 에 상응하여 정의된 점 q_0 와 비슷하게, F 에 상응하여 정의된 점 q_m 를 생각할 수 있다. (1)에서 이미 언급되었듯이, 두 점 q_0, q_m 은 알고리즘 Construct-Graph(U)에서 G 위의 노드로서 삽입되어 있다. 이제 $\Pi'_u := \text{SecondBestCoveragePath}(G, q_0, q_m)$ 라 놓자. 그러면 보조정리 2에 의하여, 노드 q_0 를 출발해 노드 q_m 에 이르는 G 위의 모든 경로는, 그것에서 U 까지 차선거리가 적어도 $dist(\Pi'_u, U)$ 이다. 만일 우리가 Π_u 의 모든 부분경로 $\Pi_i := \langle p_i \sim p_{i+1} \rangle$ ($0 \leq i \leq m-1$)에 대해서 그것보다 나은 차선거리를 갖는 G 위의 경로 $\Phi_i := \langle q_i \sim q_{i+1} \rangle$ ($0 \leq i \leq m-1$)을 제시한다면, 이들의 합집합 $\Phi := \Phi_0 \cup \Phi_1 \cup \dots \cup \Phi_{m-1}$ 는 노드 q_0 에서 노드 q_m 까지 도달하는 G 위의 경로여서 $dist(\Pi'_u, U) \leq dist(\Phi, U)$ 을 만족하며, $dist(\Phi, U) = \max_i dist(\Phi_i, U) \leq \max_i dist(\Pi_i, U) = dist(\Pi_u, U)$ 도 만족하므로, $dist(\Pi'_u, U) \leq dist(\Pi_u, U)$ 이므로, 증명이 끝난다.

그러면, 이제 0과 $m-1$ 사이 임의의 i 와 경로

$\Pi_i = \langle p_i \sim p_{i+1} \rangle$ 에 대해서, $dist(\Pi_i, U) \leq dist(\Phi_i, U)$ 인 G 위의 경로 $\Phi_i = \langle q_i \sim q_{i+1} \rangle$ 를 찾기로 하자. 여기서 경로 Π_i 는 보로노이 영역 $VR(v_{i+1}, U - \{u\})$ 내부에 존재하며, $dist(\Pi_i, U) = \max_{x \in \Pi_i} (\|p_i v_{i+1}\|, \|x v_{i+1}\|, \|p_{i+1} v_{i+1}\|)$ 이다. 점 q_i 는 언제나 u 와 v_{i+1} 의 보로노이 에지에 놓여 있으면서 G 의 노드이고, $\|q_i v_{i+1}\| \leq \|p_i v_{i+1}\|$ 를 만족한다고 가정한다. ($i=0$ 일 때, 점 q_0 는 u 와 v_1 의 보로노이 에지에 놓여 있으면서 G 의 노드이고, $\|q_0 v_1\| \leq \|I v_1\| = \|p_0 v_1\|$ 를 만족하므로, 이 가정을 만족한다.) 그러면 다음 경우들에 대해 각각 q_{i+1} 을 정의하고 G 위의 경로 Φ_i 를 정의한다.

(i) $\overline{v_{i+1}v_{i+2}} \cap VR(u, U) \neq \emptyset$ 인 경우 (그림 7의 v_1v_5 참고): v_{i+1} 과 v_{i+2} 는 $Vor(U - \{u\}) \cap VR(u, U)$ 에서 이웃한 두 사이트이며 선분 $\overline{v_{i+1}v_{i+2}}$ 가 $VR(u, U)$ 의 내부를 지나므로, $\overline{v_{i+1}v_{i+2}} \cap VR(u, U)$ 가 알고리즘 ConstructGraph(U)의 Step 2에서 G 에 에지로 추가되어 있음을 알 수 있다. 따라서 q_i 를 출발하여 $edge(u, v_{i+1})$ 을 따라가다가 $\overline{v_{i+1}v_{i+2}} \cap VR(u, U)$ 을 따라 $edge(u, v_{i+2})$ 까지 도달하는 G 위의 경로 Φ_i 가 존재한다.

즉, $q := edge(u, v_{i+1}) \cap \overline{v_{i+1}v_{i+2}}$ 와 $q_{i+1} := edge(u, v_{i+2}) \cap \overline{v_{i+1}v_{i+2}}$ 로 놓고, $\Phi_i := \overline{q_i q} \cup \overline{q q_{i+1}}$ 로 정의한다. (그림 7의 경로 $q_3q \cup q_4$ 참고) $dist(\Phi_i, U) = \max(\|q_i v_{i+1}\|, \|q v_{i+1}\|, \frac{1}{2} \|v_{i+1}v_{i+2}\|)$ 이다. $dist(\Pi_i, U)$ 와 $dist(\Phi_i, U)$ 를 비교하면, 앞 단계의 가정에 의해 $\|p_i v_{i+1}\| \geq \|q_i v_{i+1}\|$ 이며, p_{i+1} 은 v_{i+1} 와 v_{i+2} 의 수직이등분선 위에 있어서 $\|p_{i+1} v_{i+1}\| \geq \frac{1}{2} \|v_{i+1}v_{i+2}\| \geq \|q v_{i+1}\|$ 이므로, $dist(\Phi_i, U) \leq dist(\Pi_i, U)$ 이다.

(ii) $\overline{v_{i+1}v_{i+2}} \cap VR(u, U) = \emptyset$ 이고 v_{i+1} 과 v_{i+2} 가 $Vor(U)$ 의 이웃 사이트인 경우 (그림 7의 v_1v_2 참고): 이 경우에는, $q_{i+1} := \text{node}(u, v_{i+1}, v_{i+2})$, $\Phi_i := \overline{q_i q_{i+1}}$ 이라 놓자. 그러면, Φ_i 는 노드 q_i 에서 노드 q_{i+1} 에 이르는 G 위의 경로이며, $dist(\Phi_i, U) = \max(\|q_i v_{i+1}\|, \|q_{i+1} v_{i+1}\|)$ 는 $\max(\|q_i v_{i+1}\|, \|\text{node}(u, v_{i+1}, v_{i+2}) v_{i+1}\|)$ 와 같다. 이제, $dist(\Pi_i, U)$ 와 $dist(\Phi_i, U)$ 를 비교하면, 앞 단계의 가정에 의해 $\|q_i v_{i+1}\| \leq \|p_i v_{i+1}\|$ 이며, p_{i+1} 과 $\text{node}(u, v_{i+1}, v_{i+2})$ 는 선분 $\overline{v_{i+1}v_{i+2}}$ 의 수직이등분선 위에 있고 $\text{node}(u, v_{i+1}, v_{i+2})$ 이 p_{i+1} 보다 $\overline{v_{i+1}v_{i+2}}$ 의 중점에 가까

우므로 $\|node(u, v_{i+1}, v_{i+2})v_{i+1}\| \leq \|p_{i+1}v_{i+1}\|$ 이다. 따라서 $dist(\Phi_i, U) \leq dist(\Pi_i, U)$ 이다.

(iii) $\overline{v_{i+1}v_{i+2}} \cap VR(u, U) = \emptyset$ 이고 v_{i+1} 과 v_{i+2} 가 $Vor(U)$ 의 이웃 사이트가 아닌 경우 (그림 8 참조): 이 경우에는 v_{i+1} 과 v_{i+2} 사이에 시계방향으로 다른 이웃 사이트 집합들 $X = \{x_1, \dots, x_i\} \subseteq N(u)$ 가 있음을 의미한다. $z_1 := node(u, v_{i+1}, x_1)$, $q_{i+1} := node(u, x_i, v_{i+2})$ 이며, 1부터 $l-1$ 까지 모든 j 에 대해서 $z_{j+1} := node(u, x_j, x_{j+1})$ 로 놓고, $\Phi_i := \overline{q_i z_1} \cup \overline{z_1 z_2} \cup \dots \cup \overline{z_{l+1} q_{i+1}}$ 라 하자. 그러면 Φ_i 는 q_i 를 출발하여 $edge(u, v_{i+1})$, $edge(u, x_j)$ 들을 거쳐 $edge(u, v_{i+2})$ 위의 노드 q_{i+1} 까지 도달하는 G 위의 경로가 된다. $dist(\Phi_i, U)$ 는 $\max(\|q_i v_{i+1}\|, \|z_1 x_1\|, \|z_2 x_2\|, \dots, \|z_l x_l\|, \|q_{i+1} v_{i+2}\|)$ 이다. 1부터 $l-1$ 까지 모든 j 에 대해서 $\|z_j x_j\| \leq \min(\|z_j v_{i+1}\|, \|z_j v_{i+2}\|) \leq \|p_{i+1} v_{i+1}\|$ 이고, $\|q_{i+1} v_{i+2}\| \leq \|p_{i+1} v_{i+2}\| = \|p_{i+1} v_{i+1}\|$ 이며 (그림 8 참조), 앞 단계의 가정에 의해 $\|q_i v_{i+1}\| \leq \|p_i v_{i+1}\|$ 이므로 $dist(\Phi_i, U) \leq dist(\Pi_i, U)$ 이다. (증명 끝)

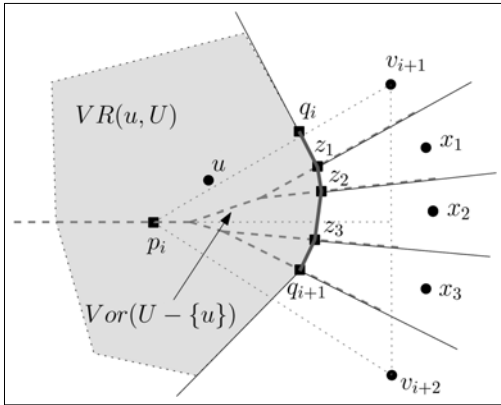


그림 8 q_i 에서 q_{i+1} 까지 도달하는 G 위의 경로 Φ_i (굵은 선)

4.2 최악의 경우 시간복잡도

우리가 3장에서 제시한 차선거리유지 경로 알고리즘은 먼저 센서들의 집합 U 에 대한 $Vor(U) \cap CH(U)$ 를 계산하고 사이트의 보로노이 영역들을 더 세분화하여 만든 가중치 비방향 그래프 G 를 리턴하는 $ConstructGraph(U)$ 알고리즘과, G 위의 임의의 시작노드 s 와 종점노드 t 를 입력으로 받아 변형 Dijkstra(G, s, t)를 통해 차선거리유지 경로를 찾는 $SecondBestCoveragePath(G, s, t)$ 알고리즘으로 구분되어진다.

먼저 $ConstructGraph(U)$ 알고리즘의 시간복잡도와

리턴되는 그래프의 크기를 분석해 보자. 센서의 개수가 n 일 때, R^2 에서 보로노이 다이어그램은 $O(n \log n)$ 의 시간이 필요하다. 이 때의 노드(보로노이 노드)는 $2n-5$ 개이며, 에지(보로노이 에지)은 $3n-6$ 개 이다. 보로노이 영역을 세분화하면서 추가한 에지들의 총 개수는 $\sum_{\{u \in U\}} (3|Vor(N(u))| + |N(u)|) \leq 10 \sum_{\{u \in U\}} |N(u)|$ 이다. 한편, $\sum_{\{u \in U\}} |N(u)|$ 는 $Vor(U)$ 의 보로노이 에지를 정확히 두 번씩 센 값이므로, $2 \times (3n-6)$ 이다. 따라서 $ConstructGraph(U)$ 알고리즘이 리턴하는 가중치 비방향 그래프 G 의 에지들의 총 개수, 즉 $|E(G)|$ 는 $63n$ 보다 작다. 따라서 $ConstructGraph(U)$ 알고리즘의 시간 복잡도는 $O(n \log n)$ 이며, 리턴되는 가중치 비방향 그래프 G 의 에지들의 총 개수, 즉 $|E(G)|$ 는 $63n$ 보다 작다.

$SecondBestCoveragePath(G, s, t)$ 의 시간복잡도와 리턴되는 차선거리유지 경로의 복잡도를 분석해 보자. 변형 Dijkstra(G, s, t) 알고리즘을 통해 차선거리유지 경로를 찾는 시간은 $O(|E(G)| \log |E(G)|) = O(n \log n)$ 이며, 리턴되는 차선거리유지 경로의 에지의 개수는 $63n$ 보다 작다.

5. 실험

5.1 실험 환경 및 조건

본 장에서 소개되는 실험들은 3장에서 소개된 차선거리유지 경로 알고리즘을 MS Compiler 6.0 환경에서 C++로 구현한 프로그램을 이용한 것이다.

실험에 사용한 점집합은 가로 50 세로 50인 이차원 격자 위에서 무작위로 생성시킨 20개의 점들로 이루어져 있으며, 점집합의 4개 이상의 점이 하나의 원위에 존재하지 않고, 3개 이상의 점이 한 직선위에 놓이지 않는다. 그리고 실험의 결과 값은 모두 소수점 넷째 자리까지 표기하였다.

5.2 실험결과 분석

거리유지 경로의 안정성을 검증하기 위해서, 최단거리유지 경로와 비교를 한다. 각 테스트 케이스에서 주어진 시작점과 도착점에 대해 차선거리유지 경로와 최단거리유지 경로를 구한다. 그리고 20개 센서의 전원을 하나씩 켜졌을 때 각 이동경로에서 유효 센서집합까지의 최단거리를 비교한다. 그림 9와 같이 차선거리유지 경로는 최단거리유지 경로에 비해서 전력의 소모는 더 크지만 하나의 센서에 이상이 생겼을 경우 더 안정적인을 알 수 있다. 심지어, 최단거리유지 경로는 1, 2, 9번째 센서가 꺼진 경우 차선거리유지 경로보다 더 나쁜 성능을 보인다.

표 2는 10개의 점집합에 대해서 최단거리유지 경로와 차선거리유지 경로의 최단거리와 한 개의 센서가 꺼졌을 경우 해당 경로의 최단거리의 표준편차를 구하였다.

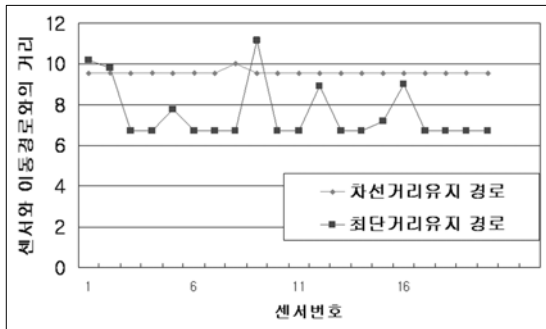


그림 9 센서가 하나씩 꺼졌을 때, 각 이동경로의 센서집합까지 최단거리

표 2 최단거리유지 경로와 차선거리유지 경로의 최단거리값과 한 개의 센서가 꺼졌을 경우

data	최단거리유지 경로		차선거리유지 경로	
	최단거리	표준편차	최단거리	표준편차
1	6.7268	16.3071	9.5382	0.4741
2	6.1033	6.1836	9.4868	0.0
3	6.1033	11.2891	9.6177	0.0
4	8.8459	20.6603	11.8716	0.0
5	5.4083	7.7354	8.2765	0.2382
6	8.1394	3.4849	11.5803	0.0
7	7.5167	3.8331	9.2195	0.1213
8	6.8007	10.5462	8.9022	0.9212
9	9.9247	6.8534	12.618	0.0
10	9.6177	3.6566	10.0896	1.2241

센서에 이상이 없는 경우, 최단거리유지 경로의 최단거리가 차선거리유지 경로의 최단거리보다 당연히 더 작다. 하지만, 하나의 센서에 이상이 생긴 경우는 차선거리유지 경로의 최단거리의 표준편차가 최단거리유지 경로에 비하여 더 작으므로, 차선거리유지 경로가 더 안정적인 모습을 보여준다.

6. 결론 및 향후 연구

본 논문에서 무선네트워크 환경에서 사용자에게 안정적인 서비스를 제공하기 위해서 센서와의 차선거리를 최소로 유지시키는 이동경로를 찾는 알고리즘을 제시하였다. 위 알고리즘으로 얻은 이동경로를 따라 이동한다면 접속하고 있는 가까운 센서에 문제가 생기더라도, 또 다른 센서와 거리를 가깝게 보장하므로 재접속이 용이하다. 이 알고리즘은 $O(n \log n)$ 의 시간을 요구하며, 최적의 경로를 찾아준다. 최적의 차선거리유지 경로를 찾음으로서 사용자는 센서 사이를 이동하면서 무선 네트워크의 잦은 끊김의 불안정성을 피할 수 있다.

향후 연구로는 차선거리유지 경로 뿐만 아니라, 세 번

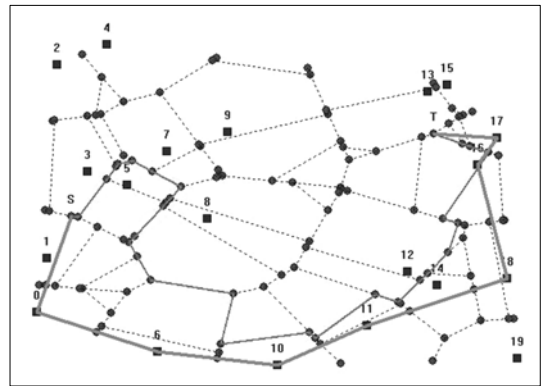


그림 10 최단거리유지 경로(굵은 선)와 차선거리유지 경로(얇은 선)

째로 가까운 센서와의 거리를 최소로 하는 이동경로를 찾는 방법을 연구한다면 보다 안정적인 서비스를 보장할 것이다. 또한 안정성과 전력소모 두 가지 요소를 원하는 비율에 따라 최단거리유지 경로와 차선거리유지 경로의 비율을 맞춰 이동경로를 찾는 방법을 연구할 수 있다.

참고 문헌

[1] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. B. Srivastava. Coverage Problems in Wireless Ad-hoc Sensor Networks. IEEE INFOCOM 2001, pp. 1380-1287.

[2] X. -Y. Li, P. -J. Wan, O. Frieder. Coverage in Wireless Ad Hoc Sensor Networks. IEEE Transactions on Computers, Vol. 52 (6), 2003.

[3] D. P. Mehta, M. A. Lopez, L. Lin. Optimal Coverage Paths in Ad-hoc Sensor Networks. IEEE Intern. Conf. on Comm. 2003, pp. 507-511.

[4] J. O'Rourke. Computational Geometry in C. 2nd eds., Cambridge, 1998.

[5] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. Computational Geometry - Algorithms and Applications, 2nd eds., Springer, 2000.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms, 2nd eds., MIT, 2000.



나 현 숙

1993년 서울대학교 수학과(학사). 1995년 포항공과대학교 수학과(석사). 2002년 포항공과대학교 수학과(박사). 2001년 3월~2002년 8월 프랑스 INRIA Postdoc. 2002년 9월~2003년 2월 홍콩과학기술대학(HKUST) 전산학과 Post Doc. 2003년 3월~현재 숭실대학교 IT대학 컴퓨터학부(조교수). 관심

분야는 알고리즘, 계산기하학, 정보이론



김 정 희

2006년 숭실대학교 미디어학부(학사). 2008
년 숭실대학교 컴퓨터학과(석사). 관심분
야는 알고리즘