

구들과 교집합을 가지는 가장 작은 구

김상섭^o 안희갑
 포항공과대학교 컴퓨터공학과
helmet1981@postech.ac.kr, heekap@postech.ac.kr

The smallest intersecting ball of balls

Sang-Sub Kim^o Hee-Kap Ahn
 Department of Computer Science and Engineering, POSTECH

요 약

구들의 집합이 주어졌을 때, 주어진 구들이 모두 서로 겹치지 않은 경우 이들 모두와 교집합을 가지는 가장 작은 구(SIBB)를 계산하는 문제에 대해 연구한다. 먼저, 점집합에 대한 Welzl의 무작위 알고리즘을 구집합에 적용하였을 때, 계산된 결과가 조건을 만족하는 최소 크기의 구가 아닐 수 있다는 사실을 증명한다. 이와 함께, 선형 기대시간에 SIBB를 계산하는 새로운 무작위 알고리즘을 제시한다. 제시된 알고리즘은 구조가 간단하며 구현이 용이하여 실용적으로 사용될 수 있다.

1. 서 론

SEBP (the Smallest Enclosing Ball of Points)란 n 개의 점을 포함하는 가장 작은 구를 찾는 문제이다. SEBP는 이미 잘 알려진 문제이고 많은 연구결과가 있다. 차원이 고정되어 있을 때 $O(n)$ 시간에 문제를 해결할 수 있는 알고리즘이 Megiddo와 Dyer에 의해 제시되었다 [1,2,3]. 하지만 이 알고리즘은 prune-and-search 패러다임을 기반으로 구현되어 있어서 알고리즘 구조가 복잡하고 시간 복잡도에 나타나지 않는 상수가 커서 실제로 사용하기에는 적합하지 않은 알고리즘이다. Welzl는 무작위 알고리즘을 (randomized algorithms) 사용해서 $O(n)$ 기대시간에 (expected time) 문제를 해결할 수 있는 알고리즘을 제시하였다 [4]. 이 알고리즘은 비록 기대시간 복잡도를 가지지만 알고리즘 구조가 간단해서 구현하기 용이하고 시간 복잡도에 나타나지 않는 상수가 작아서 실제로 수행시간이 빠른 알고리즘이다.

SEBP의 일반화 문제 중 하나로 n 개의 구를 포함하는 가장 작은 구를 찾는 문제를 들 수 있다 (SEBB (the Smallest Enclosing Ball of Balls)라고 하자). 이 문제의 응용으로는 충돌 감지, 렌더링, 가장 먼 이웃(farthest neighbour)의 근사, 생산 자동화, 유사 검색 등을 들 수 있다 [5,6,7,8,9]. Megiddo와 Dyer는 같은 접근 방법을 통해서 SEBB에서도 $O(n)$ 에 문제를 풀 수 있다는 사실을 증명하였다 [10,11]. Fischer는 Welzl의 접근방법이

SEBB에서는 적용할 수 없음을 보였다 [12]. 그러나 앞의 논문들에서 제시한 알고리즘들의 시간 복잡도에 들어 있는 상수가 차원에 지수적으로 증가하기 때문에 고차원에서는 위의 알고리즘들이 효율적이지 않다. 그래서 고차원에서도 효율적인 근사 알고리즘들이 많이 제시되었다 [13-18].

이 논문에서는 n 개의 구들 모두와 교집합을 가지는 가장 작은 구를 찾는 문제를 연구한다 (SIBB라고 하자). SIBB는 SEBP의 또다른 일반화된 문제이다. 구의 크기가 모두 같다면 SIBB문제는 SEBP와 같은 문제가 됨을 쉽게 알 수 있다. SIBB문제의 응용으로는 다음과 같은 예를 들 수 있다. n 개의 지역이 주어졌을 때 이 지역과 통신할 수 있는 가장 작은 안테나의 파워를 구하는 문제를 생각해 볼 수 있다. SIBB문제에 대해서는 이전 연구결과는 현재까지 알려진 내용이 거의 없다.

본 논문에서는 SIBB중에서 구들이 서로 겹치지 않는 경우에 대해서 주로 다룬다. 3.1장에서는 SIBB를 구할 수 있는 일반적인 방법에 대해서 이야기한다. 3.2장에서 Welzl의 접근방법을 사용하여 $O(n)$ 기대시간에 문제를 풀 수 있는 알고리즘을 제시한다. 그러나 기존의 Welzl의 접근방법을 그대로 사용할 경우 잘못된 결과를 만들 수 있다. 이장에서는 그 예를 보이고 이를 해결할 수 있는 알고리즘을 제시한다. 3.3장에서는 구들이 서로 겹칠 수 있는 경우에는 위의 접근방법을 사용하는 것이 비효율적임을 보인다. 5장에서는 SIBB의 일반화된 문제들에 대해서 언급한다.

이 논문은 정보통신산업진흥원의 SW공학 요소기술 연구 개발사업에 의해 지원 되었음을 밝힙니다.

2. 문제 정의

2차원 평면에서 중심이 c 이고 반지름이 r 인 구 $b(c,r)$ 는 다음과 같이 정의 한다. $b(c,r) = \{x \mid \|x-c\|^2 \leq r^2\}$. 주어진 구 B 의 중심과 반지름을 각각 $c(b)$, $r(b)$ 라고 표기한다.

다른 구 b' 에 대해, $b \cap b' \neq \emptyset$ 일때 b 와 b' 이 교집합을 갖는다고 한다. 교집합을 가지는 경우 $b \cap b' = 1$ 이면 b' 는 b 의 경계에 있다고 한다.

평면상에 유한하고 공집합이 아닌 구의 집합 U 에 대해, $md(U)$ 를 U 에 있는 구들 모두와 교집합을 갖는 구들 가운데 가장 작은 구라고 정의한다. 본 논문에서는 $md(U)$ 를 구하고자 한다.

3. 본 론

3.1 모든 경우에 사용할 수 있는 일반적인 방법

구 b 와 점 p 와의 거리 $d(b,p)$ 는 $\|p-c(b)\| - r(b)$ 로 정의 하자. 평면에서 farthest additively-weighted Voronoi diagram이란 구들마다 평면을 분할하는데 다음과 같은 조건을 만족하도록 정의한다. 구들의 집합 $U = \{u_1, u_2, \dots, u_n\}$ 이 주어질때, $i, j \in [1, n]$ 에 대해 u_i 의 영역 안의 점 q 는 i 가 아닌 모든 j 에 대해서 $d(u_i, q) \leq d(u_j, q)$ 를 만족한다. 이때 u_i 를 이 영역의 site라고 한다. 즉 u_i 의 영역안의 모든 점은 다른 구들보다 u_i 가 가장 멀리 떨어져 있는 구이다.

$c(md(U))$ 는 U 의 farthest additively-weighted Voronoi diagram의 정점위에 존재함을 쉽게 알 수 있다. 왜냐하면 다이어그램의 선이나 점으로 이동시키면 $md(U)$ 의 크기가 항상 줄어들기 때문이다. 따라서 $md(U)$ 는 farthest additively-weighted Voronoi diagram을 구한 후 다이어그램의 정점들마다 site와의 거리를 구한 후 가장 작은 거리를 가지는 정점이 $c(md(U))$ 가 되며 site와의 거리는 $r(md(U))$ 가 된다.

3.2 구들이 서로 겹치지 않는 경우

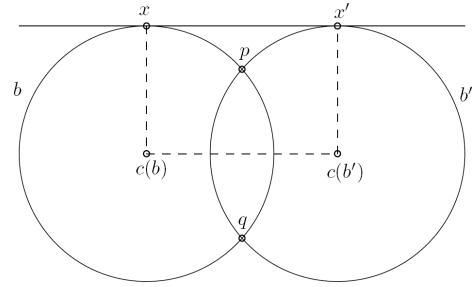


그림 2 유일한 $md(U)$ 의 존재성

우리는 먼저 다음과 같은 사실을 알 수 있다.

소정리 1 구의 집합 U 의 모든 원소들과 교집합을 가지고 반지름이 가장 작은 구는 유일하다.

증명 b 와 b' 이 U 의 모든 원소들과 교집합을 가지고 반지름이 가장 작은 서로 다른 두 개의 구라고 가정하자. 그림 1 처럼 b 가 b' 의 왼편에 있다고 가정할 경우 점 $p(q)$ 를 b 와 b' 의 경계가 만나는 두 점 가운데 위쪽(아래쪽) 점이라고 하자. 점 $x(x')$ 을 구 $b(b')$ 의 가장 위쪽에 있는 점이라고 하고 점 $y(y')$ 을 구 $b(b')$ 의 가장 아래쪽에 있는 점이라고 하자(그림 1 참조). U 의 모든 원소들은 (1) $b \cap b'$ 과 교집합을 가지거나 (2) 두 원호 xp, px' (혹은 yq, yq') 과 동시에 교집합을 가지게 된다. 그러므로 b' 를 왼쪽으로 약간 움직인 후 반지름을 약간 줄여도 여전히 b' 은 U 의 모든 원소들과 교집합을 가진다. 이는 b 와 b' 이 가장 작다는 것에 모순된다. □

또한 다음과 같은 사실을 알 수 있다.

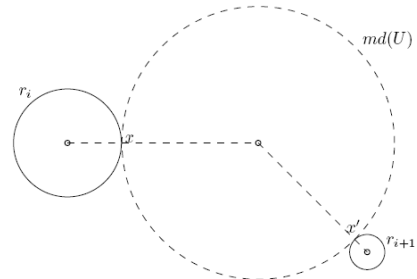


그림 2 소정리 2의 증명

소정리 2 $md(U)$ 의 경계에 있는 구들을 $md(U)$ 의 경계를 따라 시계방향 순서대로 r_1, r_2, \dots, r_k 라고 할 때, 항상 k 는 2이상이며 $\angle c(r_i)c(md(U))c(r_{i+1}) \leq \pi$ ($1 \leq i \leq k$, $i+1$ 의 인덱스는 modulo k)를 만족한다.

증명

먼저, $md(U)$ 의 경계에 단 하나의 구만 존재한다면, 경계조건을 유지하면서 $md(U)$ 의 크기를 줄일 수 있으므로, $md(U)$ 의 정의에 대해 모순된다. 따라서 k 는 항상 2 이상이다. $md(U)$ 의 경계에 있는 구중에 \angle

$c(r_i)c(md(U))c(r_{i+1}) > \pi$ 인 두 구 r_i 와 r_{i+1} 가 존재한다고 가정하자 (그림 2). 경계에 있는 구들은, 시계방향으로 r_i 와 r_{i+1} 사이에는 존재하지 않으며, r_{i+1} 과 r_i 사이에 모두 존재한다. 점 $x = md(U) \cap r_i$, $x' = md(U) \cap r_{i+1}$ 이라고 할 때, 때 그림 2와 같이 $md(U)$ 를 두 벡터 $c(md(U))x$ 와 $c(md(U))x'$ 의 합 방향으로 조금 움직일 경우, $md(U)$ 가 여전히 모든 구들과 교집합을 가지며, 경계에 있는 구들과는 더 많이 겹치게 되어 더 이상 경계에 있는 구들이 존재하지 않게 된다. 따라서 이동한 $md(U)$ 의 반지름을 더 줄여도 모든 구들과 교집합을 가질 수 있기 때문에, $md(U)$ 의 정의와 모순된다. □

소정리 2의 결과로 다음과 같은 사실을 알 수 있다.

따름정리 3 $md(U)$ 는 경계에 있는 최대 3개의 구에 의해 결정된다.

증명

소정리 2에 의해서 적어도 2개의 구가 $md(U)$ 의 경계에 있다는 것을 알고 있다. 경계에 있는 구 가운데 임의의 구 r 를 정하고, 시계방향 순서대로 경계에 있는 구들 가운데 $\angle c(r)c(md(U))c(r') \leq \pi$ 를 만족하는 맨 마지막 구를 r' 이라고 한다. 이 때, 위 각도가 정확히 π 일 경우에는 $md(U)$ 는 r 와 r' 에 의해 결정된다. 다음으로 시계방향 순서대로 $c(r')c(md(U))c(r'') \leq \pi$ 을 만족하는 맨 마지막 구를 r'' 이라고 하자. 이 경우 $c(r'')c(md(U))c(r) \leq \pi$ 가 만족되기 때문에, 이 세 개의 구들이 $md(U)$ 결정하게 된다. □

따름정리 3에서와 같이 $md(U)$ 를 결정하는 최대 3개의 구들의 집합을 S 라고 하고, $md(U)$ 의 determinant라고 부른다. S 는 항상 존재하나 유일하지 않을 수 있다. 평면 상에 유한한 구의 집합인 U 와 R 가 있을 때, $md(U, R)$ 는 U 의 원소들과 교집합을 가지고 R 의 원소들을 반드시 경계에 갖는 구들 중에 가장 작은 구라고 정의한다. 우리는 $md(U, \emptyset) = md(U)$ 임을 알 수 있고, 임의로 주어진 R 에 대해서 $md(U, R)$ 이 존재하지 않을 수 있음을 알 수 있다.

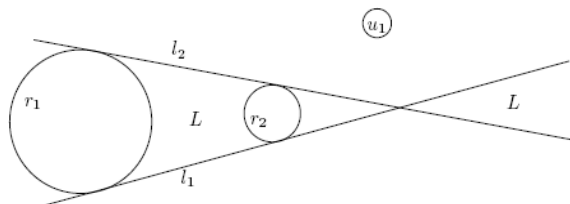


그림 3 $md(U, R)$ 의 개수가 2개가 되는 경우

아래 소정리 4를 증명하기에 앞서 우리는 먼저, $|R|=2$ 일 경우, 서로 다른 2개의 $md(U, R)$ 이 존재할 수 있음을 보일 수 있다(그림 3). 두 구 $r_1, r_2 \in R$ 에 대해 이 두 구와 공통으로 접하는 직선 l_1, l_2 에 의해 생기는 영역중에 r_1, r_2 를 포함하는 영역과 r_1, r_2 에 접하지 않는 영역으로 이루어진 L 이 존재한다고 하자. U 의 구 모두가 L 과 교집합을 가지면 $md(U, R)$ 과 $md(U \setminus \{u\}, R)$ 은 두 개가 될 수 있다(구 r_1, r_2 의 경계에 있으면서 r_1, r_2 의 아래위에 하나씩). 그러나 하나의 구라도 L 과 교집합을 가지지 않으면 $md(U, R)$ 은 유일하다. 예를 들어 그림 3에서와 같이 u_1 이 있으면 $md(U, R)$ 은 r_1, r_2 의 아래쪽에 있을 수 없다.

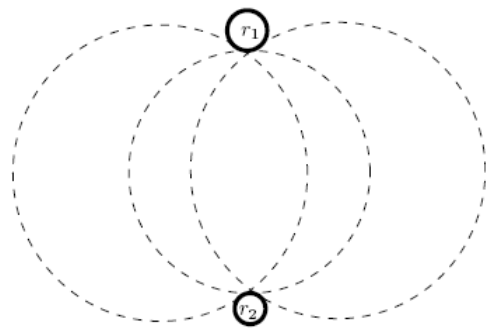


그림 5 소정리 4의 증명

소정리 4 (i) $|R| \leq 1$ 일 경우, $md(U, R)$ 은 항상 유일하며, U 의 각 원소 u 에 대해 $md(U \setminus \{u\}, R)$ 또한 유일하다. (ii) $|R| > 1$ 이고 $md(U, R)$ 이 존재하면, U 의 각 원소 $u \in U$ 에 대해 $md(U \setminus \{u\}, R)$ 또한 존재한다. 이 때 최대 두 개의 $md(U \setminus \{u\}, R)$ 이 존재할 수 있다.

증명

(i) $|R|=0$ 이면 $md(U, R)=md(U)$ 이고 따라서 $md(U \setminus \{u\}, R)=md(U \setminus \{u\})$ 이다. 따라서 소정리 1에 의해 $md(U, R)$ 과 $md(U \setminus \{u\}, R)$ 은 유일하게 정의된다.

$|R|=1$ 일 때, 서로 다른 두 개의 $md(U, R)$ 인 b, b' 이 존재한다고 가정하자. U 의 모든 구들은 $b \cap b'$ 과 교집합을 가지며, $r \in R$ 은 동시에 b, b' 의 경계에 있으므로, 소정리 1의 증명에서와 같이, 항상 더 작은 반지름을 갖는 $md(U, R)$ 이 존재한다. 따라서 b, b' 의 정의에 모순된다. 구 b 를 $md(U, R)$ 이라고 하자. u 가 b 의 경계에 있지 않으면 $b=md(U, R)=md(U \setminus \{u\}, R)$ 이 되며, 따라서 $md(U \setminus \{u\}, R)$ 이 유일하게 정의된다. u 가 b 의 경계에 있는 경우를 고려하자. 만약 b 와 $md(U \setminus \{u\}, R)$ 의 반지름이 같다면, b 는 $md(U \setminus \{u\}, R)$ 이 될 수 있다. 만약 또 다른 $md(U \setminus \{u\}, R)$ 이 존재한다면, 소정리 1의 증명에서와 같이 모순됨을 보일 수 있다. $md(U \setminus \{u\}, R)$ 의 반지름이 b

보다 작은 경우에도, 소정리 1의 증명에서와 같이 $md(U\setminus\{u\}, R)$ 의 유일성을 보일 수 있다.

(ii) $|R| > 2$ 이면 R 이 $md(U, R)$ 을 결정하게 된다. 따라서 이 경우는 $md(U, R) = md(U\setminus\{u\}, R)$ 임을 알 수 있다.

$|R| = 2$ 인 경우를 생각해 보자. u 가 $md(U, R)$ 의 경계에 있지 않다면 $md(U, R) = md(U\setminus\{u\}, R)$ 임을 알 수 있다. u 가 $md(U, R)$ 의 경계에 있다면 $md(U\setminus\{u\}, R)$ 에는 R 의 두 구가 경계에 있어야 한다(그림 4 참조). 그러므로 $md(U\setminus\{u\}, R)$ 는 존재하고 반지름이 같은 구는 2개만 존재함을 알 수 있다. □

소정리 5 $md(U, R)$ 이 존재하고 $u \cap md(U\setminus\{u\}, R) = \emptyset$ 이면 $md(U, R) = md(U\setminus\{u\}, R \cup \{u\})$ 이다.

증명

구 u 가 $md(U, R)$ 의 경계에 있지 않을 경우, 소정리 1에 의해 $md(U\setminus\{u\}, R) = md(U, R)$ 임을 알 수 있으며, $u \cap md(U\setminus\{u\}, R) = \emptyset$ 가 된다. 따라서 $u \cap md(U\setminus\{u\}, R) = \emptyset$ 이면 u 가 항상 $md(U, R)$ 의 경계에 있음을 할 수 있다. 그러므로 $md(U, R) = md(U\setminus\{u\}, R \cup \{u\})$ 이 성립한다. □

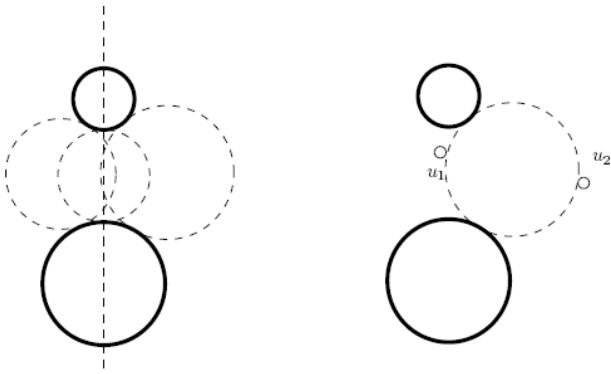


그림 6 $|R|=2$ 인 경우에 $md(U, R)$ 을 구하기

소정리 6 $md(U, R)$ 이 존재하면 크기가 $\max\{0, 3-|R|\}$ 인 determinant의 집합 S 가 존재하며 $md(U, R) \subset \text{det}(\emptyset, SUR)$ 이다.

앞에서 설명한 소정리들을 이용하여 아래와 같이 알고리즘을 설계한다. 기본적인 형태는 Welzl의 알고리즘과 유사하다 [4]. 먼저, 함수 $md(U, \emptyset)$ 를 호출한다. 이 함수는 내부에서 U 의 임의의 원소 u 를 선택하고 $u \cap md(U\setminus\{u\}, \emptyset) = \emptyset$ 가 참인지를 확인한다. 참이라면 $md(U, \emptyset) = md(U\setminus\{u\}, \{u\})$ 이고, 거짓이라면 $md(U,$

$\emptyset) = md(U\setminus\{u\}, \emptyset)$ 이다. $md(U\setminus\{u\}, \{u\})$ 와 $md(U\setminus\{u\}, \emptyset)$ 를 구하기 위해서 앞서 설명한 방법을 재귀적으로 사용한다.

$|R| \geq 2$ 라면 $md(U, R)$ 의 개수는 2개가 될 수 있기 때문에 Welzl의 알고리즘을 그대로 사용할 수는 없다. 따라서 $md(U, R)$ 를 구하는 다른 방법이 필요하다. $|R| \leq 1$ 일 경우, 위에서 재귀적으로 호출된 $md(U, R)$ 은 유일하다는 사실을 알 수 있다. 그러므로 재귀적 호출로 $md(U, R)$ 의 개수가 2개가 될 수 있는 경우는 $|R|=2$ 가 되는 경우뿐이다 ($|R| > 2$ 이면 R 에 의해서 $md(U, R)$ 이 결정되므로 이 경우는 쉽게 해결된다). 따라서 $|R|=2$ 가 될 경우에 $md(U, R)$ 을 모두 구하는 방법이 있다면 우리는 SIBB문제를 풀 수 있다. 그 방법에 대해서 알아 보자. 그림 5의 왼쪽 그림을 보면 굵은 선으로 되어 있는 구 $r, r' \in R$ 를 중심으로 보면, $md(U, R)$ 이 될 수 있는 경우는 2가지가 있다. (i) $md(U, R)$ 이 r, r' 만을 경계에 두는 가장 작은 구와 같은 경우 (ii) $md(U, R)$ 이 r, r' 외에도 다른 구를 경계에 두어서 $C_{md(U, R)}$ 이 직선 $C_r C_{r'}$ 와 만나지 않는 경우이다. 이중 가장 작은 값이 $md(U, R)$ 이 된다. (i)의 경우는 r, r' 을 경계에 두는 가장 작은 구를 구한 다음 U 의 원소들과 교집합을 가지는지 확인해 보면 된다. 만약 교집합을 가지지 않는 구가 있다면 이 경우는 해가 될 수 없다. (ii)의 경우는 ($C_{md(U, R)}$ 이 직선 $C_r C_{r'}$ 의 오른쪽에 있다고 가정한다) 다음과 같이 문제를 해결한다. $u \in U$ 가 $md(U, R)$ 과 교집합을 가지기 위해서는 u 의 위치에 따라서 다음과 같이 볼 수 있다. (그림 5의 오른쪽 그림에서) u_1 과 같은 경우에는 $md(U)$ 는 실선의 구보다 작거나 같아야 한다. u_2 의 경우에는 실선의 구보다 크거나 같아야 한다. 모든 구 $u \in U$ 에서 다음 조건을 만족해야 하므로 u 를 u_1, u_2 의 경우로 가정해서 (만들 수 없는 경우가 생길 수 있으며, 이 경우는 포함시키지 않아도 됨을 쉽게 알 수 있다.) 일차원 부등식을 구한 다음 이들의 해의 영역중에 가장 작은 값을 돌려준다.

알고리즘은 다음과 같다. Welzl의 방식을 $|R|=2$ 일때도 사용하면 $md(U, R)$ 의 개수가 2개가 될 수 있으므로 알고리즘은 구들의 집합 B 를 돌려준다. 하지만 6번째의 줄에서 $|R| < 2$ 이므로 B 는 많아야 하나의 원소를 가짐을 알 수 있다. $|R|=2$ 일때는 1차원 linear programming을 사용하여 구하게 된다.

알고리즘 1 $md(U, R)$

1. if $U = \emptyset$
2. then $B := R$ 을 경계에 두고 있는 가장 작은 구들을 구한다.
3. else if $|R|=2$

4. then $B := R$ 의 원소들의 경계에 있고 U 의 원소들과 교집합을 가지는 가장 작은 구들을 1차원 linear programming을 이용해서 구한다.
5. else 임의로 $u \in U$ 를 구한다.
6. $B := \text{md}(U \setminus \{u\}, R)$
7. if B 의 원소가 정의되고 $u \cap b = \emptyset$
8. then $B := \text{md}(U \setminus \{u\}, R \cup \{u\})$
9. return B

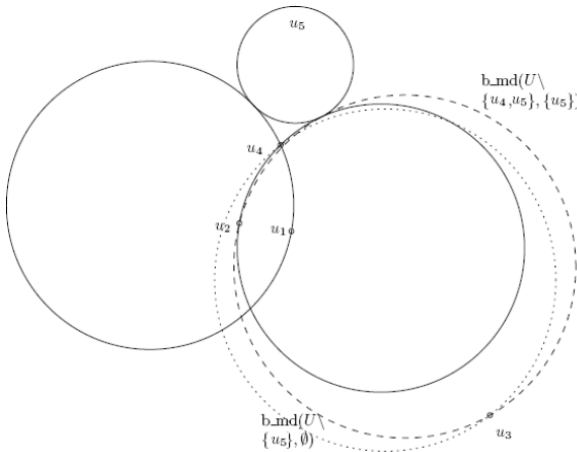


그림 7 Welzl의 알고리즘을 사용했을 때 $\text{md}(U, R)$ 이 2개가 생기는 예

정리 1 알고리즘 1을 사용하여 $\text{md}(U)$ 를 구하면 $O(n)$ 공간을 이용하여 $O(n)$ 기대시간을 사용한다.

증명

알고리즘1에서 4번째 줄을 제외하고는 상수크기의 공간을 사용한다. 4번째 줄에서는 1차원 선형 프로그래밍을 사용하므로 $O(n)$ 공간을 사용한다. 그러나 $\text{md}(U)$ 를 구하는 동안 알고리즘1을 재귀적으로 사용하게 된다. 그러나 재귀호출의 깊이가 많아야 n 이며 그중 하나의 함수에서만 4번째 줄을 실행시키므로 전체적으로 $O(n)$ 의 공간을 사용함을 쉽게 알 수 있다.

알고리즘1의 2번째 줄은 R 의 개수가 많아야 2개이므로 많아야 2개의 구로 $\text{md}(U)$ 를 구하기 때문에 상수시간이 걸린다. 4번째 줄은 일차원 선형 프로그래밍을 사용하기 때문에 $O(n)$ 시간이 걸린다. $T_1(n)$ 을 $|R| = i$ 일때 알고리즘 1의 기대시간 복잡도라고 하자. $T_1(n) = O(1)$ 이고 $T_2(n) = O(n)$ 임을 쉽게 알 수 있다. $T_1(n)$ 알기 위해서는 7번째 줄이 참이 될 확률을 알아야 한다. backward analysis를 이용하면 $\text{md}(U \setminus \{u\}, R) \approx \text{md}(U, R)$ 이기 위해서는 u 가 $\text{md}(U, R)$ 의 경계에 있고 $R \cup \{u\}$ 를 포함한 determinant에 있어야 한다. 따라서 이확률은 $(3-i)/n$

임을 알 수 있다. 따라서 $T_1(n) \leq T_1(n-1) + O(1) + (3-i)/n \cdot T_{i+1}(n-1)$ 임을 알 수 있고 $T_0(n) = O(n)$ 이 됨을 알 수 있다. □

마지막으로 Welzl의 접근방법을 그대로 사용할 경우 $\text{md}(U, R)$ 이 2개가 생기는 예를 보자. 그림 6을 보자. $U = \{u_1, u_2, u_3, u_4, u_5\}$ 이라고 하자. 처음에 u_5 를 선택해서 $\text{md}(U \setminus \{u_5\})$ 를 찾는다. $\text{md}(U \setminus \{u_5\})$ 이 그림 6에서 가는 점선임을 알 수 있다. $u_5 \cap \text{md}(U \setminus \{u_5\}) = \emptyset$ 이므로 $\text{md}(U, R) = \text{md}(U \setminus \{u_5\}, \{u_5\})$ 임을 알 수 있다. 그다음 u_4 을 선택하고 $\text{md}(U \setminus \{u_4, u_5\}, \{u_5\})$ 를 찾는다. $\text{md}(U \setminus \{u_4, u_5\}, \{u_5\})$ 는 긴 점선임을 알 수 있고, $\text{md}(U \setminus \{u_4, u_5\}, \{u_5\})$ 는 u_4 와 교집합을 가지지 않는다. 그러므로 u_4 는 R 에 포함되게 된다. 그 다음에 u_3 를 선택한다. $\text{md}(U \setminus \{u_3, u_4, u_5\}, \{u_4, u_5\})$ 의 개수가 2가 됨을 알 수 있다 (u_5 가 아닌 실선으로 된 구 2개).

3.3 구들이 서로 겹치는 경우

U 의 모든 원소들이 교집합이 존재하는 경우를 생각해 보자. 이 경우 $\text{md}(U)$ 는 모든 구들의 교집합의 영역 중 한 점이 된다. 하지만 이 영역은 3개의 구로 결정되지 않으므로 3.2장에서 설명한 방법을 사용하여 문제를 풀 수 없다. 모든 원소들의 교집합이 존재하지 않는다고 해도 3.2장에서 사용한 함수를 재귀적으로 사용되는 순서에 따라서 모든 구들의 교집합이 존재하는 경우가 생길 수가 있다. 따라서 구가 접하는 경우에 대해서는 새로운 접근 방법을 사용해야 할 것으로 보인다.

4. 결 론

이 논문에서는 구들과 교집합을 가지는 가장 작은 구에 대해서 다루었다. 특히 구들이 서로 교집합을 가지지 않을 경우에 대해서 주로 다루었다. 우리가 알기로는 SIBB에 관한 논문이 알려져 있지 않다. 우리는 이 논문에서 구현하기 쉬운 $O(n)$ 기대시간의 무작위 알고리즘을 제시했다.

앞으로 다룰 문제로는 구들이 교집합을 가질 경우의 문제, 고차원으로 일반화 하는 문제, 2센터 문제 등을 다룰 예정이다. 구들이 교집합을 가질 경우의 문제는 3.2장에서 보았듯이 기존의 방법을 사용할 수 없을 것으로 보이며, 고차원으로 일반화 하는 문제는 d 차원에서 $|R| = d$ 일 경우에 $\text{md}(U, R)$ 을 구하기 위해서는 $\text{md}(U, R)$ 이 될 수 있는 경우가 d 의 지수적으로 증가하기 때문에 쉽지 않을 것으로 보인다.

5. 참고문헌

1. M. E. Dyer. Linear time algorithms for two- and three-variable linear programs. *SIAM J. Comput.*, 13:31-45, 1984.
2. N. Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 12(4):759-776, 1983.
3. N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114-127, 1984.
4. E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes Comput. Sci.*, pages 359-370. Springer-Verlag, 1991.
5. A. Goel, P. Indyk, and K. Varadarajan. Reductions among high dimensional proximity problems. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 769-778. Society for Industrial and Applied Mathematics, 2001.
6. T. H. Hopp and C. P. Reeve. An algorithm for computing the minimum covering sphere in any dimension. Technical Report NISTIR 5831, National Institute of Standards and Technology, 1996.
7. P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179-210, 1996.
8. R. Kurniawati, J. S. Jin, and J. A. Shepherd. The SS^+ -tree: an improved index structure for similarity searches in a high-dimensional feature space. In *Proc. 5th Storage and Retrieval for Image and Video Databases SPIE*, volume 3022, pages 110-120, 1997.
9. H.-M. Will. Computation of additively weighted Voronoi cells for applications in molecular biology. PhD thesis, ETH Zurich, 1998.
10. M. E. Dyer. A class of convex programs with applications to computational geometry. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 9-15, 1992.
11. N. Megiddo. On the ball spanned by balls. *Discrete Comput. Geom.*, 4:605-610, 1989.
12. K. Fischer and B. Gärtner. The smallest enclosing ball of balls: Combinatorial structure and algorithms. In *Proceedings of 19th Annual ACM Symposium on Computational Geometry*, pages 292-301, 2003.
13. M. Bădoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 801-802. Society for Industrial and Applied Mathematics, 2003.
14. M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 250-257. ACM Press, 2002.
15. P. Kumar, J. S. B. Mitchell, and E. A. Yildirim. Approximate minimum enclosing balls in high dimensions using core-sets. *J. Exp. Algorithmics*, 8:1.1, 2003.
16. F. Nielsen and R. Nock. Approximating smallest enclosing balls. In *Proceedings of International Conference on Computational Science and Its Applications (ICCSA)*, volume 3045 of *Lecture Notes in Computer Science*. Springer, 2004.
17. S. Xu, R. Freund, and J. Sun. Solution methodologies for the smallest enclosing circle problem. *Computational Optimization and Applications*, 25:283-292, 2003.
18. G. Zhou, K.-C. Toh, and J. Sun. Efficient algorithms for the smallest enclosing ball problem. To appear in *Computational Optimization and Applications*, see also <http://www.bschool.nus.edu.sg/depart/ds/sunjiehomepage/>, accepted in 2004.