내부에 점이 없는 크고 넓은 직사각형을 찾는 근사 알고리즘

박 종 대[1], 신 찬 수[2], 안 희 갑[3], 이 현 섭[4o], 좌 경 룡[5], Otfried Cheong[6]

[1345] 한국과학기술원 전자전산학과 전산학전공

[2] 한국외국어대학교 전자정보공학부

[6] Department of Mathematics and Computer Science, TU Eindhoven

{cdpark[1], heekap[3], haru[4], kychwa[5]}@jupiter.kaist.ac.kr

[2]cssin@hufs.ac.kr

[6] o.cheong@tue.nl

Approximating the Largest Empty and Fat Rectangle

Chong-Dae Park[1], Chan-Su Shin[2], Hee-Kap Ahn[3], Hyunseob Lee[4], Kyung-Yong Chwa[5], Otfried Cheong[6]

[1345]Division of Computer Science, Department of EECS, KAIST

[2] School of Electr. and Inform. Engineering, Hankuk University of Foreign Studies

[6]Department of Mathematics and Computer Science, TU Eindhoven

요 약

n개의 점과 이를 포함하는 직사각형 B가 주어졌을 때 B 내에서 어떤 점도 포함하지 않는 가장 큰 직사각형을 찾는 선형 근사 알고리즘을 제시한다. 찾게 되는 직사각형은 좁고 긴 모양의 것은 피하며 직사각형 B의 변에 평행해야 할 필요는 없다. 모든 방향을 고려하여 가장 큰 직사각형을 근사율 $(1+\varepsilon)$로 $O(n + (\text{area}(B)/(\alpha l)^2)\log^2(\alpha/\varepsilon))$ 시간 내에 구한다.

1 서론

이 논문에서는 n 개의 점과 이 점들을 포함하는 직사각형 B가 주어졌을 때 직사각형 B 내에서 어떤 점도 포함하지 않는 가장 큰 직사각형 R을 찾는 문제를 다룬다. 직사각형 B가 목재나 철판이고 n 개의 점이 어떤 흠집의 종류라고 한다면 목재나 철판을 가공할 때 흠집이 없는 부분을 찾는 곳에 응용할 수 있다.

처음에는 찾고자 하는 직사각형 R의 변이 직사각형 B의 변에 평행해야 한다는 조건 하에서 많은 연구가 진행되었다. 이 문제는 Naamad 등이 소개하면서 $O(n^2)$ 알고리즘을 제시하였다 [1]. 그 후에 $O(n\log^3 n)$ 알고리즘 [2]과 $O(n\log^2 n)$ 알고리즘 [3]이 발표되었다. 최근에는 찾고자 하는 직사각형 R의 변이 직사각형 B의 변에 평행해야 한다는 제약조건 없이 모든 방향의 직사각형에 대하여 내부에 점이 없는 가장 큰 직사각형을 찾는 $O(n^3)$ 알고리즘이 발표되었다 [4].

본 논문에서는 직사각형 R의 방향에 대한 제한을 두지 않는 경우를 다룬다. 그리고 너무 좁고 긴 모양의 직사각형을 찾는 것을 막기 위하여, 직사각형 R의 가로와 세로의 비율 $\alpha(R)$이 주어진 $\alpha \geq 1$에 대하여 $1 \leq \alpha(R) \leq \alpha$를 만족하도록 제약을 두었다. 이는 실제 응용 상황에서도 너무 좁은 재료는 쓸모가 없으므로 의미가 있다. 직사각형 B 내에서 점을 포함하지 않으면서 가로와 세로의 비율 조건을 만족하는 임의의 방향을 가진 가장 큰 직사각형을 R^*라고 할 때 $\frac{\text{area}(R^*)}{\text{area}(R)} \leq (1+\varepsilon)$를 만족하는 직사각형 R을 찾는 근사 알고리즘을 제시한다.

2 본론

전체적인 알고리즘은 다음 순서를 따른다. 우선 직사각형 B 내에서 어떤 점도 포함하지 않는 가장 큰 원을 근사치로 구한다. 이렇게 구한 원을 이용해서 R^*의 크기를 짐작할 수 있다. 이를 이용해서 직사각형 B를 블럭으로 나누어서 R^*가 항상 어떤 2×2 블럭 내에 존재하도록 할 수 있다. 그 다음에는 2×2 블럭을 적당한 크기의 상수 개의 격자로 나누어 R을 찾는다. 하나의 격자 내에 있는 모든 점을 격자 가운데의 한 점으로 바꾸어 놓음으로써 점들을 상수 개의 점으로 근사할 수 있다. 마지막으로, 직사각형 B를 적당한 상수 개의 방향에 대해서만 근사치로 구한다.

2.1 R^*

R^*를 가로 세로의 비율이 α를 넘지 않으면서 내부에 점이 없는 가장 큰 직사각형이라고 하고, 내부에 점이 없는 가장 큰 원 C^*의 지름을 l이라고 하자. 그리고 R^*의 가로, 세로, 대각선의 길이를 각각 w^*, h^*, m^*라고 하자($h^* \le w^*$).

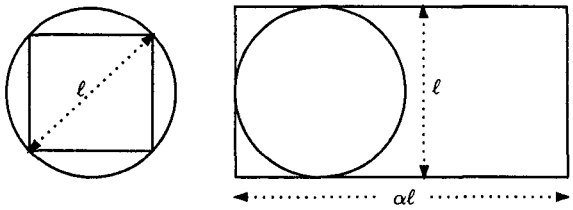

그림 1: R^*의 범위

보조정리 1. *주어진 $\alpha \ge 1$에 대하여, 다음 식이 성립한다.*

$$\frac{l^2}{2} \le \text{area}(R^2) \le \alpha l^2 \quad (1)$$

$$\frac{l}{\sqrt{2\alpha}} \le h^* \le l \quad (2)$$

$$l \le m^* \le \sqrt{1+\alpha^2}\, l \quad (3)$$

증명. R^*는 최소한 C^*에 내접하는 정사각형 보다는 크고 C^*에 외접하는 가로 세로 비율이 α인 직사각형보다는 작다. □

2.2 직사각형 B의 블럭화

직사각형 B를 간격 $d = \sqrt{1+\alpha^2}\, l$인 블럭으로 나눈다.

보조정리 2. *R^*는 임의의 2×2 블럭 안에 존재한다.*

증명. R^*가 가로 방향으로 세 개 이상의 블럭을 차지한다고 가정해보면 R^* 내부에는 d 보다 긴 선분이 존재한다는 것을 말하는데 이는 보조정리 1-(3)에 어긋난다. 세로 방향도 마찬가지이다. □

보조정리 3. *직사각형 B에는 2×2 블럭이 $O(n)$개 존재한다.*

증명. 직사각형 B를 2×2 블럭으로 채운다고 할 때 하나의 블럭에는 최소한 하나의 점을 포함하여야 한다. 그렇지 않으면 그 블럭 내에 아무 점도 포함하지 않는 원을 지름이 d 이상인 것을 찾을 수 있다는 것을 의미하는데 이는 보조정리 1-(3)에 어긋난다. □

2.3 점의 근사치

내부에 점이 없는 가장 큰 직사각형을 $(1+\varepsilon)$ 근사율로 구하기 위해서 n개의 점을 $O(\frac{\alpha^3}{\varepsilon^2})$개의 근사치 점으로 구한다.

2×2 블럭을 다시 상수 δ인 간격의 격자로 만든다. 그러면 어떤 하나의 격자 눈 안에는 여러 개의 점을 갖고 있고, 또 어떤 것들은 갖고 있지 않을 것이다. 여러 개의 점을 갖고 있있는 격자에 대해서는 그 안에 있는 모든 점들을 그 격자 눈의 한가운데로 옮긴다. 이렇게 만들어진 점들의 집합을 P_ε이라고 하고 P_ε에 속하는 어떤 점도 포함하지 않으면서 가로 세로의 비율이 α를 넘지 않는 가장 큰 직사각형을 R_ε이라고 하자. 그리고 R^*와 R_ε의 가로 세로의 비율을 유지하면서 세로의 길이를 $\sqrt{2}\delta$ 만큼 줄인 직사각형을 각각 R', R'_ε이라고 하자. P_ε을 구할 때 점이 최대 $\frac{\delta}{\sqrt{2}}$ 만큼 움직이게 되므로 R_ε의 넓이는 R'의 넓이 보다는 크다.

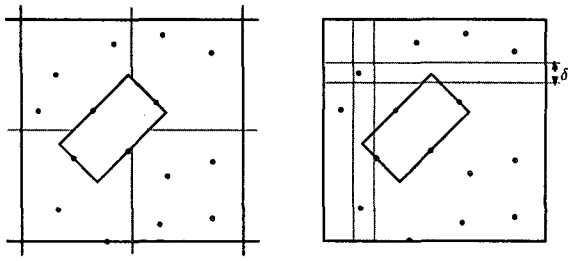

그림 2: 2×2 블럭과 블럭의 격자

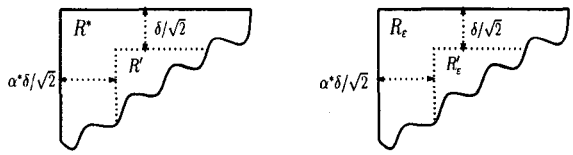

그림 3: $\text{area}(R^*) \le (1+\varepsilon)\text{area}(R'_\varepsilon)$

보조정리 4. *P_i를 블럭 B_i 안에 있는 n_i 개의 점들의 집합이라고 하면, 각각의 블럭 B_i에 대하여 $\text{area}(R^*) \le (1+\varepsilon)\text{area}(R'_\varepsilon)$ 을 만족하도록 $O(\frac{\alpha^3}{\varepsilon^2})$ 개의 점들의 집합 P'_i을 $O(n_i + \frac{\alpha^3}{\varepsilon^2})$ 시간 내에 구할 수 있다.*

증명. $\text{area}(R^*) \le (1+\varepsilon)\text{area}(R'_\varepsilon)$을 만족하는 δ를 결정하면 된다.

R^*의 가로와 세로의 비를 유지하면서 세로의 길이를 $2\sqrt{2}\delta$ 만큼 줄인 직사각형을 R''이라고 하자. 정해진 R^*에 대해서 $\text{area}(R'')$는 $\alpha(R^*) = \alpha$가 될 때 최소가 된다. 또한 정해진 δ에 대해서 $\frac{\text{area}(R'')}{\text{area}(R^*)}$는 $\text{area}(R^*)$가 최소이고 $\alpha(R^*) = \alpha$가 될 때 최소가 된다. 보조정리 1로부터 $\text{area}(R^*) \ge \frac{l^2}{2}$이다. 따라서 $\text{area}(R_\varepsilon) \ge \text{area}(R') \ge (\frac{l}{\sqrt{2\alpha}} - \sqrt{2}\delta)(\frac{\sqrt{\alpha}l}{\sqrt{2}} - \sqrt{2}\alpha\delta)$ 이고 $\frac{\text{area}(R'_\varepsilon)}{\text{area}(R_\varepsilon)}$은 $\text{area}(R_\varepsilon)$가 최소이고 $\alpha(R_\varepsilon) = \alpha$일 때 최소가 된다. $\alpha(R^*) = \alpha(R_\varepsilon) = \alpha$이고 $\text{area}(R^*)$와 $\text{area}(R_\varepsilon)$가 최소일 때 δ가 최소가 된다는 것을 알 수 있다. 따라서 $\text{area}(R^*) = \frac{l^2}{2}$, $\alpha(R^*) = \alpha$으로 두고 $\text{area}(R^*) \le (1+\varepsilon)\text{area}(R'')$을 만족하는 δ를 계산하면 $\delta \le \frac{\varepsilon l}{16\sqrt{\alpha}}$가 되는데 이를 만족하는 δ를 결정하면 된다. □

2.4 방향의 근사치

찾으려는 직사각형을 모든 방향에 대해서 구해야 하지만, $\frac{\pi}{2}$의 각을 적당한 ϕ 간격에서만 구한 직사각형의 넓이가 모든 방향에 대해서 구한 직사각형의 넓이의 $(1+\varepsilon)$ 근사율을 갖도록 구한다.

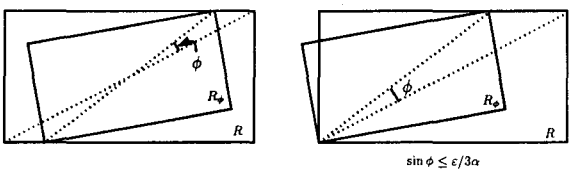

그림 4: R과 R' : $\sin\phi \le \frac{\varepsilon}{3\alpha}$를 만족하는 ϕ를 고르면 된다.

보조정리 5. *ϕ 보다 작은 각을 이루는 두 개의 방향 a, a'에 대하여, 그 방향에서 내부에 점이 없는 가장 큰 직사각형을 각각 R, R'이라고 하면, $\mathrm{area}(R) \le (1+\varepsilon)\mathrm{area}(R')$ 이다.*

증명. 방향 a를 갖는 직사각형 R_a의 가로와 세로의 비율을 $\alpha_a = \alpha(R_a)$, R의 세로 길이를 h라고 하자. 그림 4에서 보는 것처럼 R_a에 완전히 포함되고 a' 방향을 가지면서 가로와 세로의 비율이 α_a인 가장 큰 직사각형을 R_ϕ, R_ϕ의 높이를 h_ϕ, R의 대각선과 R의 세로변이 이루는 각을 θ라고 하자. 두 직사각형의 가로와 세로의 비율이 같으므로,

$$\mathrm{area}(R) \le (1+\varepsilon)\mathrm{area}(R_\phi) \quad (4)$$

$$h \le \sqrt{1+\varepsilon}h_\phi \quad (5)$$

R_ϕ의 대각선 길이는 $\frac{h}{\cos(\theta-\phi)}$가 되고, h_ϕ는 $\frac{h}{(\sqrt{1+\alpha_a^2}\cos(\theta-\phi))}$이 된다. 위의 식에 대입하면, $\cos(\theta-\phi) \le \frac{\sqrt{1+\varepsilon}}{\sqrt{1+\alpha_a^2}}$ 가 된다. 삼각법을 이용하면 ϕ가 $\cos\phi + \alpha_d \sin\phi \le \sqrt{1+\varepsilon}$을 만족하면 위의 식이 성립한다. $\cos\phi \le 1$이고 $0 < \varepsilon \le 1$이므로 $\sin\phi \le \frac{\varepsilon}{3\alpha}$이면 성립한다. □

2.5 내부에 점이 없는 근사치 원 구하기

직사각형 B를 적당한 간격의 격자로 나누면 격자의 개수가 n보다 크면서 $O(n)$이 되도록 할 수 있다. 그러면 최소한 하나의 격자에는 점이 없다. (i,j) 번째 격자에 $v(i,j)$를 다음과 같이 정의하고 가장 큰 $v(i,j)$ 값을 갖는 격자로부터 내부에 점이 없는 정사각형 영역 S를 구할 수 있다.

$$v(i,j) = \begin{cases} 0, \text{if some points in it} \\ 1 + \min\{v(i,j-1), v(i-1,j), v(i-1,j-1)\} \end{cases}$$

이렇게 구한 정사각형 S는 내부에 점이 없으면서 가장 큰 정사각형 S^*의 넓이의 $\frac{1}{3}$ 보다는 항상 크다는 것을 보일 수 있다. 그리고 S에 내접하는 원 C에도 역시 내부에 점이 없으며, 내부에 점이 없으면서 가장 큰 원 C^*는 S^*의 외접원보다는 작다는 것을 이용해서 우리가 근사치로 구한 원 C의 지름이 C^* 지름 l의 $\frac{1}{3\sqrt{2}}$보다 크다는 것을 보일 수 있다. 이로써 내부에 점이 없는 가장 큰 원에 대해서 근사율 $3\sqrt{2}$으로 근사치 원을 구했다.

2.6 알고리즘

n 개의 점을 포함하는 직사각형 B 내에서 아무 점도 포함하지 않는 가장 큰 원의 지름을 근사율 $3\sqrt{2}$로 $O(n)$ 시간 내에 구할 수 있다. 그리고 B를 간격 $\frac{\sqrt{1+\alpha^2}}{3}$로 격자화시키면 $O(\frac{\mathrm{area}(B)}{(\alpha l)^2})$ 개의 2×2 블럭이 생긴다. 각각의 2×2 블럭 내부에 있는 점들의 개수를 최대 $\frac{\alpha^3}{\varepsilon^2}$이 되도록 $O(n_i + \frac{\alpha^3}{\varepsilon^2})$ 시간 내에 근사치 점을 구할 수 있다. 마지막으로 이렇게 얻은 블럭 내의 근사치 점들에 대해서 적당한 간격의 방향에 대해서만 가장 큰 직사각형을 찾아주면 되는데 여기에는 알려져 있는 알고리즘 중 가장 결과가 좋은 알고리즘 [3]을 적용하여 전체적으로 $(1+\varepsilon)$ 근사치를 갖는 내부에 점이 없고 가로 세로의 비율 조건을 만족하는 가장 큰 직사각형을 찾을 수 있다.

정리 6. *n 개의 점을 포함하는 직사각형을 B라고 할 때, $0 < \varepsilon \le 1$과 $\alpha \ge 1$이 주어지면 가로 세로의 비율이 α를 넘지 않으면서 내부에 아무 점도 포함하지 않고 모든 방향에 대해서 고려한 가장 큰 직사각형에 대하여 근사율 $(1+\varepsilon)$을 갖는 직사각형을 $O(n + (\frac{g\alpha^4}{\varepsilon^3})\log^2(\frac{\alpha}{\varepsilon}))$ 시간 내에 구할 수 있다. 이때, $g = O(\frac{\mathrm{area}(B)}{(\alpha l)^2})$이다.*

3 결론

내부에 점이 없는 가장 큰 직사각형을 찾는 문제에 대하여 점들을 포함하는 직사각형의 변과 평행한 것을 찾는 방면으로는 많은 연구 [1-3]가 있어왔다. 그리고 최근에야 모든 방향을 고려한 결과 [4]가 발표되었는데 이의 시간 복잡도 $O(n^3)$가 높기 때문에 가로 세로의 비율을 제약하고서 최적의 값에 $(1+\varepsilon)$ 근사율을 갖는 직사각형을 $O(n + (\frac{g\alpha^4}{\varepsilon^3})\log^2(\frac{\alpha}{\varepsilon}))$ 시간 내에 구하는 선형 시간 근사 알고리즘을 제안하였다.

이를 바탕으로 점 대신 선분 등이 있는 경우에 대해서도 근사 알고리즘을 구하는 것이 향후 연구과제로 남아있다.

참고 문헌

[1] A. Naamad, D.T. Lee, W.L. Hsu. *On the maximum empty rectangle problem*, Discrete Appl. Math. 8, 1984

[2] B. Chazelle, R.L. Drysdale, D.T. Lee. *Computing the largest empty rectangle*, SIAM J. Comput. 15, 1986

[3] A. Aggarwal, S. Suri. *Fast algorithm for computing the largest empty rectangle*, in: Proc. 3rd Annual ACM Symp. on Computational Geometry, 1987

[4] J. Chaudhuri, S.C. Nandy, S. Das. *Largest empty rectangle among a point set*, J. of Algorithms 46, 2003