

Geometrijski Algoritmi

Maximum empty rectangle

Ivan Pop-Jovanov

Avgust 2023

Sažetak

Maximum empty rectangle problem (problem maksimalnog praznog pravougaonika) je problem nalaženja pravougaonika najveće površine unutar pravougaonika A takvog da ne sadrži ni jednu od n tačaka iz skupa S . Ovde će biti prikazana dva algoritma, jedan složenosti $O(n^2)$ a drugi $O(n \log^2 n)$.

Sadržaj

1	Opis problema	2
2	Notacija	2
3	Naivni Algoritam	2
4	Optimizacija	2
5	Poređenje efikasnosti	3
6	Zaključak	3

1 Opis problema

Neka je A pravougaonik čije su stranice paralelne koordinatnom sistemu i $S = \{P_1, P_2, \dots, P_n\}$, $n > 1$, skup tačaka u njegovoj unutrašnjosti. Pravougaonik je definisan svojim gornjim, donjim, levim i desnim granicama, u oznakama A_t , A_b , A_l i A_r . Tačke iz skupa S su definisane svojim x i y koordinatama.

MER (maximum empty rectangle) problem je da se nađe pravougaonik najveće površine koji se ceo nalazi unutar pravougaonika A , takav da u svojoj unutrašnjosti ne sadrži ni jednu tačku iz S .

2 Notacija

Kažemo da je pravougaonik M restricted rectangle (RR), ako zadovoljava naredne uslove:

1. M se potpuno sadrži u A
2. M ne sadrži ni jednu tačku iz S .
3. Svaka stranica M sadrži ili bar jednu tačku iz S , ili se sadrži unutar jedne od stranica A .

Rešenje MER problema će očigledno biti neki RR. Broj RR pravougaonika je ograničen sa $O(n^2)$ što nam daje i intuitivnu složenost za algoritam grube sile.

Pretpostavimo dalje da su tačke iz S nasumično i nezavisno odabrane po uniformnoj raspodeli iz unutrašnjosti A . Ispostavlja se da je očekivani broj RR pravougaonika $O(n \log n)$ [1].

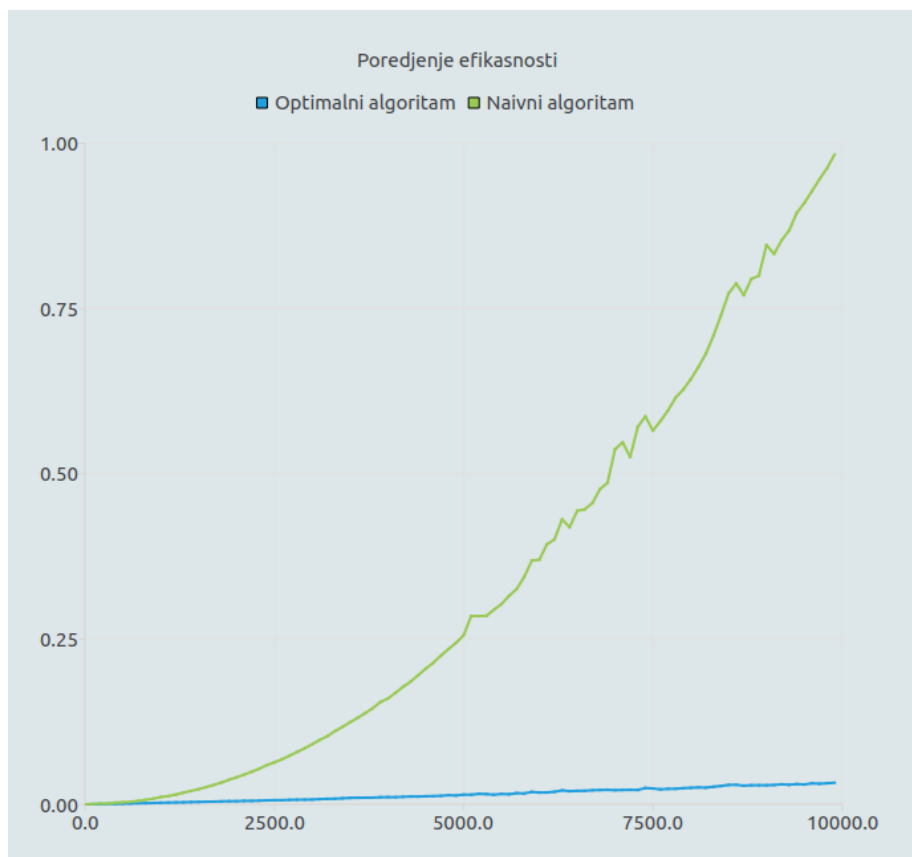
3 Naivni Algoritam

Iz prethodnog direktno sledi algoritam grube sile. Prvo je potrebno da sortiramo sve tačke po y koordinati. U ovaj niz uključujemo i gornju i donju granicu pravougaonika. Sada idemo redom kroz tačke od vrha na dole, nazovimo trenutnu tačku X_i . Za svako i prolazimo kroz sve tačke X_j ispod trenutne i proveravamo da li postoji RR takav da mu je gornja granica u tački X_i a donja u tački X_j . Sve vreme čuvamo najveći pronađeni RR.

4 Optimizacija

Ispostavlja se da nije potrebno da za X_i proveravamo svako X_j koje se nalazi ispod njega. Korišćenjem strukture *semi-dynamic heap* možemo u složenosti $O(\log n)$ u svakom koraku da postepeno odsecamo određene kandidat tačke, za svaki RR na koji naiđemo. Ako označimo broj RR pravougaonika sa s , imamo da je složenost novog algoritma $O(s \log n)$, a pošto znamo da je očekivana vrednost s zapravo $O(n \log n)$ dobijamo konačnu prosečnu složenost $O(n \log^2 n)$.

5 Poređenje efikasnosti



Slika 1: Kao što je i očekivano, optimizovani algoritam je efikasniji od naivnog.

6 Zaključak

Ovaj pregled prati optimizaciju opisanu u originalnom radu Naamad et al. Nakon tog rada izašle su i dalje optimizacije [2, 3].

Literatura

- [1] Naamad, Amnon, D. T. Lee, and W-L. Hsu. "On the maximum empty rectangle problem." *Discrete Applied Mathematics* 8, no. 3 (1984): 267-277.
- [2] Chazelle, Bernard, R. L. Drysdale, and D. T. Lee. "Computing the largest empty rectangle." *SIAM Journal on Computing* 15, no. 1 (1986): 300-315.
- [3] Aggarwal, Alok, and Subhash Suri. "Fast algorithms for computing the largest empty rectangle." In *Proceedings of the third annual symposium on Computational geometry*, pp. 278-290. 1987.