Algoritham brišuće prave za određivanje hijerarhije između kružnica

Andrija Urošević

Univerzitet u Beogradu, Matematički fakultet
Katedra za računarstvo i informatiku
`andrija.urosevic@matf.bg.ac.rs`

Avgust, 2023.

Opis algoritma i implementacionih detalja algoritma brišuće prave za određivanje hijerarhije između kružnica. Algoritam je zasnovan i implementiran po uzoru na rad: *Deok-Soo Kim, Byunghoon Lee, Kokichi Sugihara: A Sweep-Line Algorithm for the Inclusion Hierarchy among Circles* [1].

1 Opis problema

Neka je dat skup kružnica $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, gde je kružnica C_i definisana sa centrom (x_i, y_i) i poluprečnikom r_i. Pretpostavimo da kružnice mogu da sadrže druge kružnice dok presek između dve kružne linije nije dozvoljen. Za dati skup $\mathcal{C}$, želimo da konstrušemo hijerarhiju između kružnica, tj. da odredimo relaciju sadržanja između svake dve kružnice iz $\mathcal{C}$.

Jedan primer kružnica i odgovarajuće hijerarhije između kružnica je dat na slici 1.

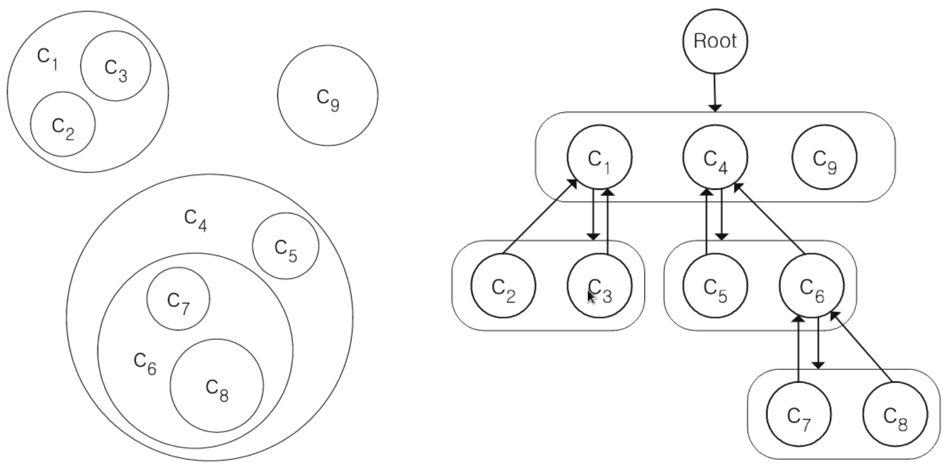
2 Algoritam

Razmotrićemo prvo naivni algoritam grube sile, koji radi u $O(n^2)$ vremenu. Nako toga primetićemo da je algoritam moguće rešiti mnogo efikasnije u $O(n \log n)$ vremenu, ako koristimo tehniku brišuće prave zajedno sa odgovarajućim strukturama podataka.

2.1 Algoritam grube sile

Ideja algoritma grube sile je u proveri svih parova različitih kružnica da li se jedna sadrži u drugoj. Provera da li je neka kružnica sadržana u drugoj kružnici se složenosti $O(1)$, pa je ukupna složenost algoritma $O(n^2)$.

Algoritam grube sile je opisan algoritmom 1.



(a) (b)

Slika 1: Primer kružnica: (a) Grafički prikaz kružnica; (b) Hijerarhija između kružnica.

Algorithm 1: Gruba sila

Data: Skup kružnica $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$

for $C_i \in \mathcal{C}$ **do**

  for $C_j \in \mathcal{C}$ **do**

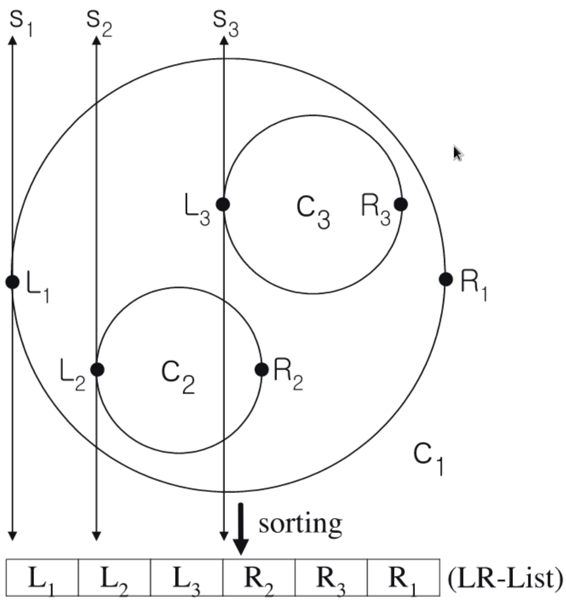
    if $i \neq j$ i C_i *sadrži* C_j **then**

      $C_i.sadrzi \leftarrow C_j$;

    end

  end

end



Slika 2: Lista događaja

2.2 Algoritam brišuće prave

Metod brišuće prave se oslanja na vertikalnu liniju, *brišuću pravu*, koja prolazi sa leva na desno. Na tom putu ona česno ne menja zadate osobine, ali postoje tačke u kojima dolazi do promene zadatih osobina i te tačke se nazivaju *tačke događaja*.

2.2.1 Događaji

Neka su najlevlja i najdešnja ekstremna tačka kružnice C_i obeležene sa L_i i R_i, respektivno. Zaključujemo da ih ima $2n$ za n kružnica. Ekstremne tačke sortirane po x koordinatama čine listu događaja. Ekstremne tačke koje odgovaraju najlevljim ekstremnim tačkama kružnica su događaji *rađanja*, dok su ekstremne tačke koje odgovaraju najdešnjim tačkama kružnica događaju *umiranja*. Slika 2 ilustruje primer tri kružnice i njihovu odgovarajuću listu događaja.

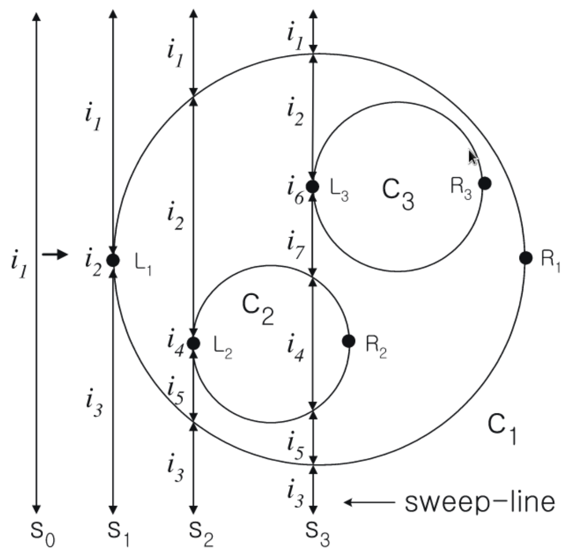
2.2.2 Generisanje intervala

Brušuća prava je u svakom trenutku podeljena na *intervale*. Granice intervala su presečne tačke obima kružnica i brišuće prave. Čuvaju se u crveno crnim stablima kako bi obezbedili efikasnu pretragu, umetanje i brisanje u $O(\log n)$ vremenu.

Svaki interval i sadrži dve vrednosti za donju i gornju granicu. Ako se granice intervale poklope takav interval nazivamo *nula interval*. Slika 3 predstavlja 4 scenarija brišuće prave i intervale u njima.

Inicijalno u S_0 imamo samo jedan interval i_1. U trenutku L_1, interval i_1 se deli tri intervala $[i_i, i_2, i_3]$. U trenutku L_2, interval i_2 se deli na $[i_2, i_4, i_5]$, te lista intervala postaje $[i_1, i_2, i_4, i_5, i_3]$. Pored toga, u trenutku L_2, kako je interval i_2 unutar kružnice C_1 možemo zaključiti da je kružnica C_2 sadržana u kružnici C_1.

Opštije, ako se tačka L_j, nalazi unutar interval i_k, onda interval i_k delima na tri intervala $[i_k, i_{m+1}, i_{m+2}]$, gde je m trenutan ukupan broj intrvala. Pri tome, ako je kružnica



Slika 3: Nastajanje novih itervala na događajima rađanja.

C_k unutar intervala i_k, a C_j kružnica kojoj odgovara najlevija ekstremna tačka L_j, onda C_k sadrži C_j.

2.2.3 Brisanje intervala

Kada naiđemo na najdešnju ekstremu tačku, tj. događaj umiranja, treba ažurirati intervale. Na slici 4 je prikazana detekcija događaja umiranja, zajedno sa brisanjem dva intervala koja su nastala pri rađanju odgovarajuće kružnice.

2.2.4 Algoritam

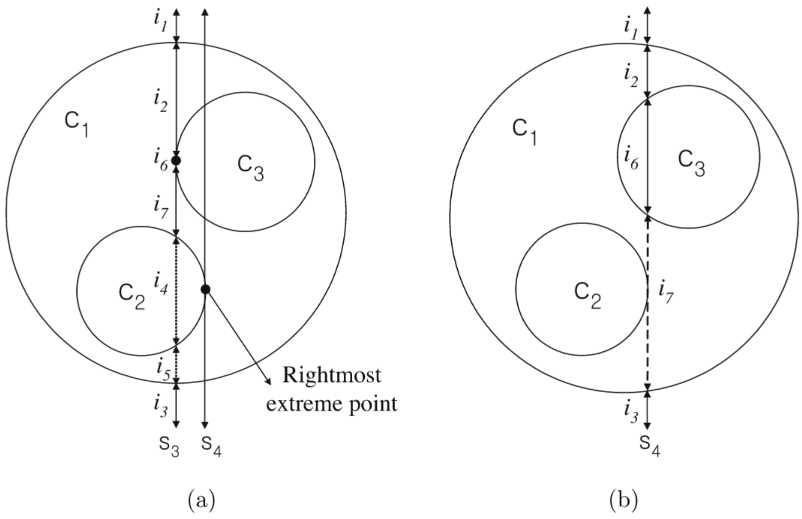
Algoritma brišuće prave za detekciju hijerarhije između kružnica je prikazan algoritmom 2.

3 Strukture podataka

Kružnica Svaka kužnica ima pridruženi centar (par realnih brojeva) i poluprečnik (neki realan broj). Pored toga svaka od njih ima definisanu funkciju za pronalaženje gornje i donje tačke sa obima za zadatu x koordinatu. Ova funkcija je od značaja u pronalaženju odgovarajućeg intervala u crveno crnom stablu intervala.

Događaj Događaj je predstavljen odgovarajućom x koordinatom, indikatorom da li je događaj rađenje/umiranje kuržnice, i pokazivačom na odgovarajuću kružnicu.

Interval Interval je ključna struktura koja treba da definiše između ostalog i poredak u crveno crnom stablu. Od veliko značaja je ažurirati ovu strukturu korektno tokom izvođenja algoritma kako bi se očuvao poredak. Struktura čuva pokazivač na kružnicu koju sadrži, pored toga, čuva i gornju i donju granicu preko pokazivača na gornju i donju



Slika 4: Brisanje dva intervala na događaju umiranja: (a) i_4 i i_5 su odgovarajući intervali za brisanje; (b) modifikovani intervali brišuće prave nako brisanja dva intervala.

Algorithm 2: Brušuća prava

Data: Skup kružnica $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$

Inicijalizuje praznu listu događaja E;

for $C_i \in \mathcal{C}$ **do**

  $E \leftarrow E + \{Event(C_i, open)\}$;

  $E \leftarrow E + \{Event(C_i, close)\}$;

end

Sortiraj ekstremne tačke E u neopadajućem poretku po njihovim x koordinatama;

Inicijalizuje crveno crno stablo intervala $I \leftarrow \{i_0\}$, gde je $i_0 = (-\infty, +\infty)$;

for $e_j \in E$ **do**

  if $e_j.open$ **then**

    $i = (y, y') \leftarrow I.find(e_j.y)$;

    if i_2 *sadrži* C_k **then**

      $C_k.sadrzi \leftarrow e_j.circle$;

    end

    $i' \leftarrow (y, e_j.y)$;

    $i'' \leftarrow (e_j.y, y')$;

    $I \leftarrow I + \{i', i''\}$;

  else

    $i' \leftarrow I.find(e_j)$;

    $i \leftarrow i'.prev$;

    $i'' \leftarrow i'.next$;

    $I \leftarrow I - \{i', i''\}$;

  end

end

---

kružnicu. Kako brišuća prava preseca obim kružnice u potencijalno dve tačke treba voditi računa o tome da li uzimamo gornju ili donju tačku za svaku od graničnih kružnica.

Na osnovu toga možemo lako izračunati gornju i donju granicu intervala. Kako intervala nemaju presek lako je definisati poredak između njih:

$$i_1 < i_2 \iff i_1.LB + \epsilon < i_2.LB \lor i_1.UB + \epsilon < i_2.UB.$$

Jednakost između dva intervala se sada lako definiše kao

$$i_1 = i_2 \iff \neg(i_1 < i_2) \land \neg(i_2 < i_1).$$

3.1 Crveno crna stabla

Za čuvanje intervala koristimo crveno crno stablo.

Pretraga intervala Svaki put kada bršuća prava naiđe na novu kružnicu generiše se najlevlja ekstremna tačka. Treba odrediti u kom se intervalu ona nalazi. Kako se intervali čuvaju u crveno crnom stablu odgovarajući interval se može pronaći u $O(\log n)$ vremnu. Tokom pretrage treba korisiti nove granice intervala koje računamo za trenutnu x vrednost brišuće prave.

Primetimo da ovo neće promeniti poredak intervala.

Insertovanje intervala Kada naiđemo na najlevlju ekstremu taču kružnice dva nova intervala se kreiraju. Njihovo dodavanje u crveno crno stablo zahteva $O(\log n)$ vremena.

Brisanje intervala Kada nastane događaj umiranja na najdešnjoj ekstemnoj tački kuržnice treba obrisati odgovarajuće intervale iz crveno crnog stabla. Dva intervala brišemo, dok treći ažuriramo tako da njegovo sledeće računanje gornje i donje granice ostane konzistentno. Ove operacije se mogu izvršiti u $(\log n)$ vremenu.

4 Složenost algoritma

Generisanje liste događaja i njeno sortiranje zahteva $O(n \log n)$ koraka, kako se za svaku kružnicu generišu dva događaja, događaj rađanja i događaj umiranja, za levu i desnu ekstremnu tačku kružnice, respektivno. Inicijalizacija crveno crnog stabla interval ima konstantnu složenost. Za svaku događaj u najgorem slučaju vršimo $O(\log n)$ operacija, te obrada svih događaja ima složenost $O(n \log n)$.

Ukupa složenost algoritma je $O(n \log n)$.

Literatura

[1] Deok-Soo Kim, Byunghoon Lee i Kokichi Sugihara. „A Sweep-Line Algorithm for the Inclusion Hierarchy among Circles". U: *Japan Journal of Industrial and Applied Mathematics* 23.1 (2006.), str. 127–138.