Problem najmanjšega kroga

Jan Pristovnik in Jan Lampič

5.1.2018

1 Predstavitev problema

Problem najmanjšega kroga ali najmanjšega pokrivnega kroga je matematični probelm izračunavanja najmanjšega kroga, ki vsebuje vse določene množice točk v evklidski ravnini. Če želimo problem rešiti glede na dane točke $P = \{p_1, p_2, .., p_n\}$ v evklidski ravnini, moramo poiskati središče in polmer kroga, ki bo vseboval vse točke iz množice P. Ustrezni problem v n-dimenzionalnem prostoru, najmanjši problem z omejevalno sfero, je izračun najmanjše n-dimenzionalne sfere, ki vsebuje vse določene množice točk. Problem najmanjšega kroga je prvotno predlagal angleški matematik Joseph Sylvester leta 1857.
Problem najmanjšega kroga se pojavlja na različnih področjih uporabe, kot so:

- Primer težave z lokacijo objekta, v katerem je treba izbrati lokacijo novega objekta, ki bo zagotavljal storitve vsem v naprej izbranim strankam. Nov objekt mora biti postavljen tako, da bo čim bolj zmanjšal najdaljšo razdaljo, ki jo mora katera koli stranka prepotovati, da doseže novi objekt.
- Za reševanje problemov v okoljski znanosti (načrtovanje in optimizacija topil), prepoznavanje vzorcev (iskanje referenčnih točk), biologija (analiza beljakovin), politična znanost (analiza stranskih spektrov), strojništvo (optimizacija stresa) in računalniška grafika (sledenje žarkov, izločanje) itd.

Najenostavnejši algoritem bi bil, preveriti vse kroge definirane z dvema in tremi točkami, ter preveriti ali vsebuje vse ostale točke in nakoncu izbrati najmanjšega med ustreznimi. Če bi imeli prvotno n točk, bi bilo takih krogov $O(n^3)$. Za vsak krog bi potrebovali $O(n)$, da bi preverili ali vsebuje vse ostale točke, torej skupna časovna zahtevnost bi bila $O(n^4)$. Tak algoritem se je prvič pojavil že okoli leta 1869, od takrat se je zgodilo veliko izboljšav. Danes poznamo veliko praktično uporabnih algoritmov. Teoretično najbolj izpopolnjeni algoritmi imajo časovno zahtevnost $O(n)$.

2 Karakterizacija problema

Večina geometrijskih pristopov za problem išče točke, ki ležijo na meji najmanjšega kroga in temeljijo na naslednjih preprostih dejstvih:

- Najmanjši krog, ki pokriva vse izbrane točke je enoličen.
- Najmanjši krog, ki pokriva vse točke iz množice P lahko določimo z največ tremi točkami, ki ležijo na robu in so vsebovane v P. Če je krog določen samo z dvema točkama, mora razdalja med točkama predstavljati premer minimalnega kroga. Če ga sestavljajo tri točke, potem trikotnik sestavljen iz teh treh točk, ne vsebuje topih kotov.

Kot je pokazal Nimrod Megiddo, je problem iskanja najmanjšega kroga, ki pokriva vse izbrane točke, rešljiv v linearnem času. Enako velja tudi za iskanje najmanjše n-sfere, ki vsebuje vse množice točk v n-dimenzionalnem prostoru.

3 Algoritem Welzla

Za dano množico točk v ravnini P, z md(P) označimo najmanjši krog, ki vsebuje vse točke iz P. Dovolimo tudi, da je $P = \emptyset$, ko je md(P)=$\emptyset$ in $P = \{p\}$, ko je md(P)=p.
Pri konstrukciji algoritma nam bodo pomagala dejstva iz karakterizacije problema. tako vemo, da je md(P) določen z največ tremi točkami iz P, ki ležijo na rabou kroga md(P). To pomeni, da obstaja $S \subseteq P$ na robu md(P) tako, da je $|S| \leq 3$ in md(P)=md(S). Torej, če točka $p \notin S$ potem je md$(P - \{p\})$=md(P) oziroma, če je md$(P - \{p\}) \neq$ md(P) potem $p \in S$ in p leži na robu md(P).
Za množico n-tih točk P izračunamo md(P) postopoma. Začnemo s prazno množico, ki ji zaporedoma dodajamo točke ter ohranjamo najmanjši krog, ki vsebuje doslej obravnavane točke. Naj bo $P = \{p_1, p_2, ..., p_n\}$ in recimo, da smo že izračunali $D =$ md$(\{p_1, ..., p_i\})$ za nek $i, 1 \leq i < n$. Če je $p_{i+1} \in D$, potem je D tudi najmanjši krog, ki vesbuje prvih $i + 1$ točk in lahko nadaljujemo z naslednjo točko. V nasprotnem primeru ($p_{i+1} \notin D$), pa vemo, da mora p_{i+1} ležati na robu D'=md$(\{p_1, p_2, ..., p_{i+1}\})$. D' pa izračunamo s funkcijo `b_minidisk(`A, p`)`, ki izračuna najmanjši krog, ki vsebuje $A = \{p_1, ..., p_i\}$ in ima točko $p = p_{i+1}$ na robu. Ideja je ta, da problem postane bolj enostaven ko določimo točko p za robno. Predpostavimo, da `b_minidisk` že obstaja. Potem lahko zgornji algoritem formuliramo s sledečo rekurzivno zvezo.

Algorithm 1 `minidisk(`P`)`

Require: Končna množica točk v ravnini P
Ensure: md(P)
  if $P = \emptyset$ **then**
    $D := \emptyset$
  else
    Choose $p \in P$;
    $D :=$ `minidisk`$(P - \{p\})$;
    if $p \notin D$ **then**
      $D =$ `b_minidisk`$(P - \{p\}, p)$;
    end if
  end if
  return D;

Preden podamo opis funkcije `b_minidisk`, predpostavimo, da potrebuje $c|A|$ korakov za izračun najmanjšega kroga. Kakšna je potem časovna zahtevnost zgornjega algoritma? $p \in P$ izberemo naključno, vsako točko iz P z enako verjetnostjo $1/|P|$. Naj bo $t(n)$ pričakovano število korakov, ki jih potrebuje `minidisk(`P`)` za $|P| = n$. Potem za t velja

$$t(n) < 1 + t(n-1) + P(p \notin \mathrm{md}(P - \{p\}) \cdot c(n-1),$$

kjer "1"predstavlja konstanten čas za zahtevano delo, ostala izraza pa se nanašata na pričakovano delo, ki ga povzročata klica na funkciji `minidisk` in `b_minidisk`. Obstajajo največ tri točke p iz P tako, da md$(P)$$\neq$md$(P - \{p\})$; za vse ostale točke $p \in$ md$(P - \{p\})$ velja, da je $P(p \notin \mathrm{md}(P - \{p\}) \leq 3/n$ iz česar sledi, da je $t(n) \leq (1 + 3c)n$.
Algoritem za `b_minidisk(`A, p`)` je podoben kot `minidisk` podan zgoraj, vendar

sedaj potrebujemo še postopek za izračun najamnjšega kroga dane množice točk z dvema določenima točkama na robu in tako naprej. Preden opišemo postopke pa podamo še pojem in lemo.
Za končni množici točk P in R v ravnini, definiramo $\mathrm{b\_md}(P, R)$ kot najmanjši krog, ki vsebuje vse točke iz P in ima R na robu. Očitno velja $\mathrm{b\_md}(P, \emptyset)=\mathrm{md}(P)$ in $\mathrm{b\_md}(P, R)$ je lahko nedefiniran takoj, ko R ni prazna.

Lema 1 *Naj bosta P in R končni množici točk v ravnini, P neprazna in naj bo p točka iz P.*

(i) *Če obstaja krog, ki vsebuje P in ima R na robu, potem je* $\mathrm{b\_md}(P, R)$ *dobro definiran oz. enoličen.*

(ii) *Če $p \notin$* $\mathrm{b\_md}(P - \{p\}, R \cup \{p\})$*, potem p leži na robu* $\mathrm{b\_md}(P, R)$*, pod pogojem, da obstaja.*

(iii) *Če* $\mathrm{b\_md}(P, R)$ *obstaja, potem obstaja množica S največ* $\max\{0, 3 - |R|\}$ *točk iz P tako, da velja* $\mathrm{b\_md}(P, R) = \mathrm{b\_md}(S, R)$.

Upoštevati je treba, da v splošnem množica S ni enolična. Pomembna implikacija (iii)točke leme je, da obstaja največ $\max\{0, 3 - |R|\}$ točk iz P, ki niso vsebovane v $\mathrm{b\_md}(P - \{p\}, R)$.
Točka (ii) leme nakazuje kako izračunati $\mathrm{b\_md}(P, R)$. Če je $P = \emptyset$, je problem enostaven in izračunamo $\mathrm{b\_md}(\emptyset, R)$ direktno. V nasprotnem primeru pa naključno izberemo $p \in P$ in izračunamo $D = \mathrm{b\_md}(P - \{p\}, R)$. Če je $p \in D$, potem je $\mathrm{b\_md}(P, R)=D$; drugače pa je $\mathrm{b\_md}(P, R)=\mathrm{b\_md}(P - \{p\}, R \cup \{p\})$.

Algorithm 2 `b_minidisk`(P, R)

Require: Končni množici točk v ravnini P in R
Ensure: $\mathrm{b\_md}(P, R)$

```
if P = ∅ or |R| = 3 then
  D := b_md(∅, R)
else
  Choose random p ∈ P;
  D := b_minidisk(P − {p}, R);
  if p ∉ D then
    D := b_minidisk(P − {p}, R ∪ {p});
  end if
end if
return D;
```