PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652
Letnik **23** (1995/1996)
Številka 5
Strani 294–297

Stane Indihar:

NAJMANJŠI KROG

Ključne besede: matematika, ravninska geometrija, C-P algoritem, trikotniki, krogi.

Elektronska verzija: http://www.presek.si/23/1268-Indihar.pdf

NAJMANJŠI KROG

1. Uvod

V lanski 4. številki Preseka je na strani 213 Martin Juvan zastavil naslednjo nalogo:

Opiši algoritem, ki za dano končno množico točk v ravnini poišče krog z najmanjšim polmerom, ki vsebuje vse točke iz množice. Algoritem naj bo čim učinkovitejši (in tudi dovolj preprost, da ga je mogoče brez prevelikih težav sprogramirati).

V naslednji številki Preseka pa je bil na straneh 296, 297 objavljen Juvanov algoritem, kako takšen krog poiskati.

Ker ima omenjena naloga kar precejšnjo zgodovino in tudi posplošitve, spregovorimo o njej nekoliko obširneje.

V literaturi najdemo podatek, da je nalogo v letu 1857 zastavil J. J. Sylvester. Tri leta zatem je konstrukcijsko rešitev podal B. Peirce, v letu 1885 pa je do enakega postopka kot Peirce prišel tudi G. Chrystal. Njun pristop k reševanju naloge je sedaj znan pod imenom Chrystal – Peirceov algoritem (C-P algoritem). Ogledali si ga bomo pozneje.

Zanimanje za nalogo se je spet pojavilo po drugi svetovni vojni z razvojem operacijskega raziskovanja in matematičnega programiranja. V prvo področje sodi npr. naslednja praktična naloga:

Kje naj postavimo urgentno helikoptersko reševalno postajo, da bo razdalja do najbolj oddaljenega kraja čim manjša?

Pri tej nalogi imajo vsi kraji enako "težo". Če pa upoštevamo, da so kraji lahko različno "obteženi", dobimo splošnejšo nalogo. Kot kriterij za različne obtežitve je npr. število prebivalcev.

Posplošitev Sylvestrove naloge dobimo tudi, če namesto množice točk v ravnini vzamemo končno množico točk v 3-razsežnem ali pa kar n-razsežnem prostoru ($n \geq 3$). V tem primeru iščemo n-razsežno kroglo z najmanjšim polmerom, ki vsebuje vse točke dane množice. To nalogo sta J. Elzinga in D. W. Hearn prevedla v posebni problem nelinearnega programiranja, ki je rešljiv v končno mnogo korakih.

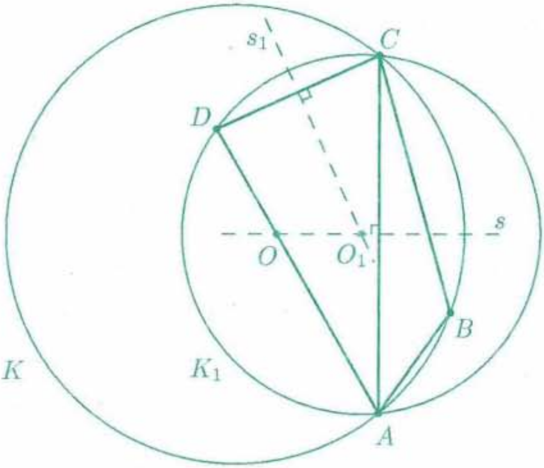
2. C-P algoritem

Osnovan je na naslednji očitni lastnosti trikotnika:

- *za topokotni (ali pravokotni) trikotnik je najdaljša stranica hkrati premer najmanjšega kroga, ki trikotnik vsebuje;*
- *za ostrokotni (ali pravokotni) trikotnik je najmanjši krog, ki vsebuje ta trikotnik, kar njemu očrtani krog;*

in na izreku:

Naj bo ABC topokotni trikotnik s topim kotom pri oglišču B in točka D znotraj trikotniku ABC očrtanega kroga K. Če je kot $\measuredangle ADC <$ $< \measuredangle ABC$, potem ima trikotniku ADC očrtani krog K_1 manjši polmer, kot je polmer kroga K, in v K_1 se nahaja tudi točka B.



Dokaz izreka. Na sliki je točka O središče kroga K, točka O_1 pa središče kroga K_1. Središči obeh krogov sta na simetrali s daljice AC. Ker je točka D znotraj kroga K, seka simetrala s_1 daljice CD (na tej simetrali je središče kroga K_1) simetralo s v točki O_1, ki se nahaja med točko O in razpoloviščem daljice AC. Krog K_1 ima zato manjši polmer, kot je polmer kroga K, in celotni lok ABC brez krajnih točk je znotraj kroga K_1, torej tudi točka B.

Preidimo sedaj na prikaz C-P algoritma.

Naj bo S končna množica točk v ravnini. Iščemo najmanjši krog, ki to množico vsebuje.

Korak 0 (začetni korak). Narišemo krog, ki poteka skozi dve točki A, B množice S in vsebuje vse točke te množice.

Korak 1. Med točkami množice $S - \{A, B\}$ poiščemo točko, pri kateri je kot $\measuredangle ADB$ najmanjši. Najdeno točko (ki morda ni enolično določena) označimo s C.

a) Če je $\measuredangle ACB \geq \frac{\pi}{2}$ je naloga rešena. Daljica AB je premer najmanjšega kroga, ki vsebuje množico S.

b) Če je $\measuredangle ACB < \frac{\pi}{2}$ preidemo na naslednji korak.

Korak 2. Trikotniku ABC očrtamo krog.

a) Če trikotnik ABC ni topokoten, je naloga rešena. Njegov očrtani krog je najmanjši krog, ki vsebuje množico S.

b) Če je ABC topokotni trikotnik, odstranimo izmed točk A, B tisto, pri kateri je topi kot – naj bo to npr. točka B – in preidemo na korak 1, pri katerem točko B zamenja točka C.

Skladno z izrekom bo vsak naslednji krog, ki ga konstruiramo, manjši od predhodnega in bo vseboval vse točke množice S. Pri iskanju najmanjšega kota v prvem koraku pa ni treba več upoštevati točk, ki jih odstranimo v drugem koraku. Ker je točk množice S končno mnogo in se pri vsaki ponovitvi iskanja manjšega kroga, ki vsebuje množico S, polmer zmanjša, je algoritem končen.

V zvezi z začetnim korakom, ki je nekako nedorečen, sta R. K. Chakraborty in P. K. Chaudhuri (1981) predlagala poseben postopek, po katerem najdemo začetni par točk A, B. Prav tako sta opazila, da v drugem koraku lahko poleg točke B odstranimo iz nadaljnjih računov oziroma konstrukcij tudi vse tiste točke $D \in S - \{A, C\}$, pri katerih je $\sphericalangle ADC$ topi kot.

Računske izkušnje, do katerih sta pri preskušanju algoritma prišla avtorja članka, ki je naveden v uvodu, so zelo ugodne. Število točk v množici S je bilo med 100 in 1100. Število operacij je bilo sorazmerno s številom točk v množici S.

3. Zaključek

Da bi bil prikaz reševanja Sylvestrove naloge popolnejši, navedimo še naslednje.

V delu *B. J. Oommen, An Efficient Geometric Solution to the Minimum Spanning Circle Problem, Operations Research 35(1987)1, 80-86*, je prikazan algoritem, ki izboljša tudi korak 2b v P-C algoritmu. Pri vseh računskih poskusih, razen v zelo redkih izjemah, je bil s tem algoritmom najmanjši krog najden v natanko dveh iteracijah. Oommen navaja, poleg drugih dosežkov, tudi metodo, ki jo je razvil N. Megiddo (1982). Problem najmanjšega kroga Megiddo prevede v dvorazsežni problem linearnega programiranja in ga reši v času, ki je linearno odvisen od števila točk.

Zanimivo je omeniti, da je Oommen svoj algoritem preskušal tudi pri iskanju najmanjših krogov, ki objemajo Velika jezera Severne Amerike.