Matematički fakultet

Projekat iz predmeta Konstrukcija i analiza algoritama II

Školska 2018/2019

Određivanje preseka dva konveksna mnogougla u linearnom vremenu

Student:
Stefan Marić 1118/2018

Mentori:
dr Vesna Marinković
Mirko Spasić

March 3, 2019

Sadržaj

1 Problem

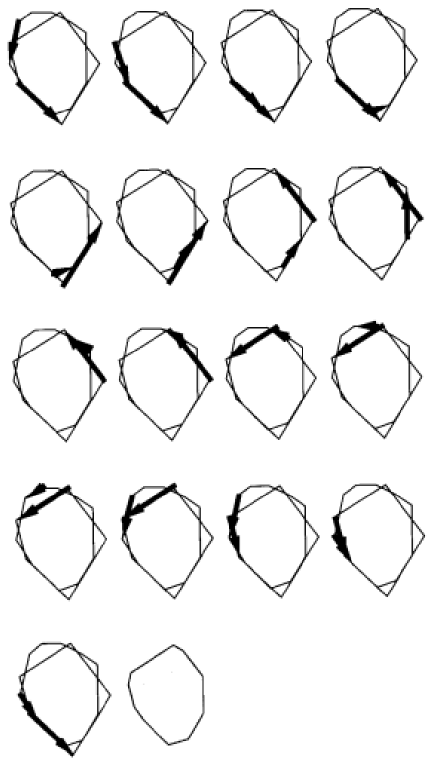
Problem koji nam se postavlja spada u domen geometrije, a skup algoritama koji ga rešavaju potpadaju u takozvane geometrijske algoritme. Pre izlaganja samog problema definisaćemo i razjasniti osnovne pojmove, a sve zarad boljeg upoznavanja čitaoca sa problemom (a samim tim i sa rešenjem).
Tačka p u ravni predstavlja se uređenim parom svojih koordinata (x, y). **Duž** se zadaje parom tačaka p i q koje predstavljaju krajeve te duži i označavaćemo je sa $p - q$. **Put** P je niz tačaka $p_1, p_2, ..., p_k$ i duži $p_1 - p_2, p_2 - p_3, ..., p_{k-1} - p_k$ koje ih povezuju. **Zatvoreni put** je put čija se poslednja tačka poklapa sa prvom. Zatvoreni put se naziva i **mnogougao**. Tačke koje definišu mnogougao su njegova **temena**.Mnogougao se predstavlja nizom, a ne skupom, tačaka jer je bitan redosled kojim se tačke zadaju. **Konveksni mnogougao** je mnogougao čija unutrašnjost, sa svake dve tačke koje sadrži, sadrži i sve tačke te duži.
Sada smo spremni da definišemo problem.

Data su dva konveskna mnogougla cikličkim rasporedom svojih temena. Konstruisati algoritam linearne vremenske složenosti za određivanje preseka ovih mnogouglova. Izlaz (takođe konveskan mnogougao) treba da bude predstavljen ciklički uređenom listom temena.

2 Rešenje

Pretpostavimo da su nam granice mnogougla orijentisane u pozitivnom matematickom smeru (dakle u smeru suprotnom od kretanja kazaljke na časovniku) i neka su **A** i **B** *usmerene* ivice na mnogouglovima **P**, odnosno **Q**. Osnovna ideja algoritma leži u tome da **A** i **B** "jure" jedan drugoga, prilagođavajući svoju brzinu, tako da se "sastaju" na svim presecima ova dva mnogougla. Osnovna struktura algoritma prikazana je u pseudokodu ispod.

Algoritam: *Presek dva konveksna mnogougla*
Izaberi A i B proizvoljno.
repeat:
    if A seče B **then**:
        Proveri da li je kraj.
        Ažuriraj unutrašnji fleg.
    Napreduj sa A ili sa B u zavisnosti od uslova
until A i B ne obiđu svoje mnogogulove u potpunosti
Obradi $P \cap Q = \emptyset$ i $P \subset Q$ i $Q \subset P$ slučajeve



Slika 1: Ilustracija kretanja vektora po ivicama mnogouglova

Neka je a indeks temena određen vrhom (glavom) od A i neka je b indeks temena određen vrhom (glavom) od B. Ako B "cilja" na pravu koja sadrži A, ali je ne seče, onda želimo da napredujemo po B da bismo se "približili" mogućem preseku sa A.

Neka je $H(A)$ otvorena poluravan levo od vektora A. Kako mnogouglove obilazimo u pozitivnom matematičkom smeru, to znači da je unutrašnjost mnogougla upravo levo od vektora.
Ključnu ulogu u razradi algoritma imaće vektorski proizvod i njegova primena na određivanje najkraćeg zaokreta od A ka B.
Pošto svi vektori pripadaju jendoj ravni, možemo uvesti vektorski proizvod vektora A i B, u oznaci $A \times B$ kao:

$$A \times B = (A_x \cdot B_y)\text{-}(B_x \cdot A_y)$$

Ovo je ništa drugo do znak rezultujućeg vektora u standardnom vektorskom proizvodu.

Primetimo sada, ukoliko je $A \times B > 0$ jasno sledi da je najkraći zaokret od A ka B u pozitivnom matematičkom smeru.

Iz gore-navedenih redova da se zaključiti sledeće:

$$\text{Ako je} A \times B > 0 \text{ i} b \notin H(A)\text{, ili}$$

$$\text{Ako je} A \times B < 0 \text{ i} b \in H(A)\text{, ili}$$

tada napreduj po B.
Ukoliko se i prisetimo da $B \times A =$ -$A \times B$, slično možemo izvesti i za A:

$$\text{Ako je} A \times B < 0 \text{ i} a \notin H(B)\text{, ili}$$

$$\text{Ako je} A \times B > 0 \text{ i} a \in H(B)\text{, ili}$$

tada napreduj po A.
Ako su oba vektora usmerena jedan ka drugome, može se napredovati po bilo kom. Ukoliko nijedan nije usmeren ka onom drugom, napredujemo po onome koji je izvan leve poluravni ili po bilo kome, ukoliko su oba izvan. Uz nešto razmišljanja, može se rezonovati da ukoliko $a \in H(B)$ i$b \in H(A)$, vektori su usmereni jedan ka drugome. Imajući sve ovo u vidu, slučajevi se mogu organizovati sledećom tabelom:

$A \times B$	$a \in H(B)$	$b \in H(A)$	Napreduj
> 0	T	T	A
> 0	T	F	A ili B
> 0	F	T	A
> 0	F	F	B
< 0	T	T	B
< 0	T	F	B
< 0	F	T	A ili B
< 0	F	F	A

Tabela se može uopštiti, tako da poprimi sledeći izgled:

$A \times B$	Uslov poluravni	Napreduj
> 0	$b \in H(A)$	A
> 0	$b \notin H(A)$	B
< 0	$a \in H(B)$	B
< 0	$a \notin H(B)$	A

2.1 Zaustavljanje:

Program bi se mogao završavati pri ponovnom nailasku na prvu tačku preseka, u slučajevima da takva tačka postoji, no to ovde nije slučaj. Kada oba vektora obiđu sve ivice mnogouglova, posao je završen i program treba prekinuti. Ipak, u nekim degenerativnim slučajevima, vektor ivice jednog poligona neće napredovati, te je empirijski utvrđeno da je dovoljan uslov zaustavljanja da bilo koji vektor obiđe poligon dva puta.

3 Rezultati

Kao što je već rečeno, rezultat rada programa je lista temena preseka. Pored toga, ilustrovaćemo složenost $O(n + m)$ gde je n broj temena poligona P, a m broj temena poligona Q.

n	m	$n + m$	broj temena preseka	vreme u sekundama
3	3	6	6	0.000668
4	4	8	0	0.000315
4	4	8	4	0.000771
10	5	15	12	0.000831
9	9	18	0	0.000313
9	9	18	7	0.002057

3.1 Zaključak

Na ovom mestu završavamo priču o preseku konveksnih mnogouglova uz par propratnih opaski o konkretnom algoritmu. Naime, sam po sebi, algoritam je daleko od savršenog. Verovatno najveći nedostatak je što postoji tirada specijalnih slučajeva za koje će algoritam dati pogrešno rešenje, i koje bi shodno tome trebalo posebno obraditi. No, kako je poenta ovog rada prikazati linearnu složenost po broju temena, taj zadatak se prepušta čitaocu. Još neki od nedostataka su što ovako implementiran algoritam neće dobro raditi ukoliko su temena mnogouglova, iako ciklički, data u negativnoj matematičkoj orijentaciji, kao i to što uslov sigurnog izlaska iz petlje dvostrukim prolaskom po granicama mnogouglova nije najsrećnije rešenje.

4 Reference

- *dr Vesna Marinković* - Konstrukcija i analiza algoritama 2
- *Nikola Ajzenhamer* - Konstrukcija i analiza algoritama 2
- *Joseph O'Rourke* - Computational geometry in C
- Gnuplot
- Vektorski proizvod