MATEMATIKA 2

dr Bojana Borovićanin

2022.

3.5 KOMBINATORNA OPTIMIZACIJA

Kombinatorna optimizacija

Ekstremalni zadaci na konačnim (ili prebrojivim) skupovima proučavaju se pod imenom ***kombinatorna optimizacija***.

Takvi zadaci se obično mogu formulisati kao zadaci na tzv. ***težinskim grafovima***.

Definicija 1. ***Težinski graf*** je graf u kome je svakoj grani dodeljen neki broj. Drugim rečima, grafu $G = (V, E)$ pridruženo je preslikavanje $w: E \to R$ koje svakoj grani $e \in E$ dodeljuje broj $w(e)$ kao ***težinu***. Funkcija w zove se ***težinska funkcija*** grafa.

U raznim interpretacijama težina grane predstavlja njenu *dužinu*, *propusnu moć (ili kapacitet)*, *pouzdanost*, *cenu*, *prenos* itd.

Kombinatorna optimizacija

Neki ekstremalni zadaci kombinatorne optimizacije formulišu se na težinskim grafovima. Kod takvih grafova svakoj grani e između čvorova v_i i v_j pridružena je težina (dužina)

$$w(e) = d_{ij}.$$

Od veličina d_{ij} može se formirati kvadratna matrica

$$D = (d_{ij}).$$

Matrica D naziva se ***težinska matrica***.

Težinski graf G, tj. uređena trojka $G = (V, E, w)$, često se naziva i ***mreža***. U raznim interpretacijama njima odgovaraju konkretne fizičke mreže, kao što su telekomunikacione mreže, računarske mreže, saobraćajne mreže itd.

Kombinatorna optimizacija

Definicija 2. ***Minimalno razapinjuće stablo*** u grafu je stablo minimalne težine (dužine) u datom grafu koje sadrži sve čvorove grafa.

Razmatraćemo sledeće probleme:

- problem najkraće povezujuće mreže,
- problem ekstremalnih puteva u mreži,
- problem trgovačkog putnika.

Najkraća povezujuća mreža

Problem *Neka je dato n gradova koje treba povezati telefonskom mrežom najkraće dužine.*

Posmatrajmo graf tražene mreže, u kome su gradovi čvorovi, a grane telefonske linije. Dužina telefonske linije između dva grada pridružuje se odgovarajućoj grani grafa kao dužina (težina) grane.

Graf tražene telefonske mreže mora biti *povezan*, *ne sadrži konture* (jer bi se neke grane u konturama mogle udaljiti). Dakle, graf najkraće povezujuće mreže je **stablo**.

Najkraća povezujuća mreža

- Neka su gradovi numerisani sa $1, 2, \ldots, n$, a d_{ij} rastojanje između gradova i i j.
- Potrebno je formirati težinski graf G u kome je grani e između čvorova i i j pridružena težina (dužina) d_{ij} (grafu G pridružena je težinska matrica ili *matrica rastojanja* D).
- Najkraćoj povezujućoj mreži za datih n gradova odgovara u grafu G razapinjući podgraf oblika stabla. Problem se svodi na nalaženje razapinjućeg stabla minimalne dužine, tj. minimalnog razapinjućeg stabla.
- Biće opisana dva klasična algoritma za nalaženje minimalnog razapinjućeg stabla u proizvoljnom povezanom težinskom grafu G.

Kraskalov algoritam

1. Sortirati grane ulaznog grafa u neopadajući poredak (po težinama) i sve ih markirati kao nerazmatrane. Za izlazni graf uzeti graf bez grana (razapinjući podgraf ulaznog grafa).

2. Izabrati iz skupa nerazmatranih grana onu koja je najlakša, markirati je kao razmatranu i priključiti (bar privremeno) izlaznom grafu. Ako ta grana ne obrazuje u njemu konturu zadržati je, a ako obrazuje konturu odbaciti je.

3. Ako je broj grana rezultirajućeg grafa $n-1$ (polazni graf je povezan i ima n čorova) prekinuti rad. U suprotnom slučaju preći na korak 2.

Primov algoritam

1. Izabrati čvor v_0 grafa G i granu najmanje težine e_1 iz skupa grana koje sadrže čvor v_0, da bi se obrazovalo stablo T_1.

2. Ukoliko je dobijeno stablo T_k sa granama $e_1, e_2, \dots, e_k$ i ako postoji čvor koji nije u stablu T_k, potrebno je izabrati granu minimalne težine, koja je susedna grani stabla T_k i koja ima čvor koji ne pripada stablu T_k. Dodati ovu granu stablu T_k da bi se dobilo stablo T_{k+1}.

3. Ponavljati korak 2 dok svaki čvor grafa G ne bude u stablu.

Ekstremalni putevi u mreži

Mnogi problemi iz realnog života se mogu tretirati kao problemi vezani za ekstremalne puteve u težinskom grafu, tj. mreži. U tom kontekstu biće posmatrani sledeći optimizacioni zadaci:

1) Za dati par čvorova mreže, recimo s i t, naći dužinu najkraćeg puta i bar jedan najkraći put od čvora s do čvora t.

2) Za dati čvor mreže, recimo s, naći dužinu najkraćeg puta i bar jedan najkraći put od čvora s do bilo kog od preostalih čvorova mreže.

3) Za bilo koji par čvorova mreže naći dužinu najkraćeg puta i bar jedan najkraći put između njih.

Ekstremalni putevi u mreži

- U nastavku će biti izložen algoritam Dijkstra, ograničavajući se na rešavanje zadataka 1) i 2).
- Ovaj algoritam se zasniva na tehnici poznatoj kao *obeležavanje* (dodeljivanje oznaka).
- Na početku oznaka svakog je čvora je jednaka ∞ (osim početnog, za koji je jednaka 0).
- U toku rada algoritma, oznake odgovaraju dužinama trenutno nađenih puteva od početnog čvora do preostalih, dok na kraju postaju definitivne i odgovaraju stvarnim dužinama najkraćih puteva.

Ekstremalni putevi u mreži

Izloženi algoritam sadrži u sebi dve operacije: obradu grana i obradu čvorova.

- Pod obradom grane $e = xy$ podrazumeva se provera nejednakosti

$$d(y) > d(x) + w(e),$$

gde su $d(x)$ i $d(y)$ oznake čvorova x i y, redom, a $w(e)$ je težina grane e).

- Ako ova nejednakost važi tada se oznaka čvora y menja, tj. $d(y) = d(x) + w(e)$.

- Pod obradom čvora x podrazumeva se obrada svih grana $e = xy$ koje izlaze iz čvora x.

Algoritam Dijkstra

1. Svim čvorovima mreže dodeliti sledeće privremene oznake: čvoru s dodeliti oznaku $d(s) = 0$, a svim ostalim čvorovima, uključujući i čvor t, dodeliti oznake $d(x) = \infty$.

2. Ako nema čvorova sa privremenim oznakama preći na korak 3. U suprotnom, pronaći čvor x koji ima najmanju privremenu oznaku. Čvoru x dodeliti stalnu oznaku i obraditi ga, tj. za svaku granu $e = xy$ (čvor y ima privremenu oznaku), ako je $d(y) > d(x) + w(e)$ staviti $d(y) = d(x) + w(e)$. Zatim ponoviti korak 2.

3. Sada se prelazi na postupak izdvajanja najkraćeg puta. Potrebno je izdvojiti najkraći put između čvorova s i t.

Algoritam Dijkstra

Od svih čvorova susednih žvoru t odrediti čvor x_k za koji je

$$(1) \qquad d(t) - d(x_k) = w(x_k t).$$

Ponavljanjem opisanog postupka, tj. određivanjem čvora x_{k-1}, susednog čvoru x_k za koji je

$$(2) \qquad d(x_k) - d(x_{k-1}) = w(x_{k-1} x_k),$$

itd, na kraju se dolazi do čvora x_1, kome je susedan čvor s, i za koji je

$$d(x_1) - d(s) = w(s x_1).$$

Na ovaj način dobija se put koji ide preko čvorova $s, x_1, \dots, x_{k-1}, x_k, t$.

Algoritam Dijkstra

Sabiranjem jednakosti $(1), (2)$ itd. za sve čvorove na tom putu i imajući u vidu da je $d(s) = 0$, dobija se da je dužina ovog puta jednaka oznaci čvora t, tj.

$$d(t) = w(sx_1) + w(x_1x_2) + \cdots + w(x_{k-1}x_k) + w(x_kt).$$

- U ovoj verziji algoritma dobija se rešenje zadatka 2).
- Rešenje zadatka 1) se dobija ako se u koraku 2. prekine dalji rad kada čvor t dobije stalnu oznaku.

HVALA NA PAŽNJI!