Aukcijski algoritmi u mrežnoj optimizaciji

Stoić, Dora

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:465785

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Dora Stoić

AUKCIJSKI ALGORITMI U MREŽNOJ OPTIMIZACIJI

Diplomski rad

Voditelj rada:
dr. sc. Marko Vrdoljak, izv. prof.

Zagreb, 2018

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Ovaj rad istražuje klasu algoritama za rješavanje linearnog problema mrežnog toka i njegovih raznih posebnih slučajeva kao što su najkraći put, maksimalan tok, dodjeljivanje i transport. Te algoritme, nastale 1979. godine, zbog svoje konstrukcije bazirane na ideji aukcije, nazivamo aukcijskim algoritmima. Prototip metode, iz kojeg se mogu izvesti drugi algoritmi, je aukcijski algoritam za problem dodjeljivanja, a razlog tomu je što se problem toka s minimalnim troškovima lako može pretvoriti u problem dodjeljivanja. Ovo je intuitivna metoda koja djeluje poput prave aukcije gdje se osobe natječu za objekte podizanjem svojih cijena putem natječaja. Aukcijski algoritmi se miču od klasične ideje poboljšanja troškova te u svakoj iteraciji mogu pogoršati primarnu i dualnu funkciju, no na kraju ipak nalaze optimalnu vrijednost. Imaju izvrsnu računalnu složenost i njihova vremena izvođenja su slična, a često i superiornija vremenima izvođenja njima konkurentnih algoritama.

Na početku rada ćemo se fokusirati na osnovni aukcijski algoritam za problem dodjeljivanja, a zatim ćemo raspraviti o njegovom proširenju na druge probleme. U poglavljima 1 i 2 razvijamo aukcijski algoritam za simetrični problem dodjeljivanja. Konkretno, raspravljamo o većem broju pitanja koja su zajednička svim vrstama aukcijskih algoritama, uključujući ϵ-skaliranje i rješavanje nedopustivosti. Također raspravljamo o alternativnom obliku aukcijskog algoritma, nazvanog obrnuta aukcija; dok se kod regularne aukcije osobe natječu za objekte podizanjem njihovih cijena, kod obratnog algoritma se objekti natječu za osobe nudeći im popuste. U poglavlju 3 razvijamo aukcijski algoritam za problem najkraćeg puta, dok u poglavlju 4 raspravljamo o njegovom proširenju na transportni problem. U poglavlju 5 uvodimo generički algoritam za problem minimalnog toka te razvijamo metodu ϵ-relaksacije za isti problem kao specijalni slučaj generičkog algoritma.

Poglavlje 1

Dodjeljivanje naivnom aukcijom

Kod klasičnog problema dodjeljivanja imamo n osoba i n objekata koje trebamo spojiti na principu jedan-na-jedan. Neka je $a_{i,j}$ korist dodjeljivanja i–toj osobi j-tog objekta, a cilj nam je da dodjeljivanjem objekata osobama postignemo maksimalnu ukupnu korist. Skup objekata koji se mogu dodijeliti osobi i je neprazan i označavamo ga s $A(i)$. Skup svih mogućih parova (i, j) označavamo s $\mathcal{A}$,

$$\mathcal{A} = \{(i, j) | j \in A(i), i = 1, 2, ..., n\}.$$

Također, s $B(j)$ označavamo skup svih osoba koje se mogu dodijeliti objektu j. *Dodjeljivanjem* smatramo skup S (moguće prazan) parova (i, j) takvih da je $j \in A(i)$ za sve $(i, j) \in S$, za svaku osobu i postoji najviše jedan par $(i, j) \in S$ i za svaki objekt j postoji najviše jedan par $(i, j) \in S$. Obzirom na dodjeljivanje S, kažemo da je osoba i dodijeljena ako postoji par $(i, j) \in S$; inače kažemo da je i nedodijeljena. Istu terminologiju koristimo i za objekte. Dodjeljivanje je dopustivo ako sadrži n parova tako da su sve osobe i svi objekti dodijeljeni; u suprotnom ga zovemo nedopustivim dodjeljivanjem. Stoga ako postoji dopustivo dodjeljivanje tako nazovemo i problem, u suprotnom problem nazivamo nedopustivim. Dakle, tražimo dopustivo dodjeljivanje (skup parova $(1, j_1)$,...,(n, j_n)) iz $\mathcal{A}$, tako da su svi objekti $j_1, j_2, ..., j_n$ međusobno različiti), koje je optimalno u smislu da maksimizira ukupnu beneficiju $\sum_{i=1}^{n} a_{ij_i}$.

Opisani problem još nazivamo simetričnim problemom dodjeljivanja kako bi ga razlikovali od asimetričnog problema dodjeljivanja kod kojeg je broj osoba manji od broja objekata.

Problem dodjeljivanja je važan u mnogim praktičnim kontekstima, ali je također i od velike teorijske važnosti. Možemo ga formulirati kao zadaću linearnog programiranja. Jedna od najvažnijih klasa problema linearnog programiranja, minimizacija troškova mrežnog toka, može biti svedena na problem dodjeljivanja jednostavnom reformulacijom. Dakle, svaka metoda za rješavanje problema dodjeljivanja može se generalizirati za rješavanje

problema minimizacije troškova mrežnog toka.

Teorija dualnosti se bavi odnosom između originalnog problema mrežne optimizacije i problema optimizacije kojeg nazivamo dualni problem.
Kako bismo razvili intuitivno shvaćanje dualnosti, korisno je upoznati ekonomski ravnotežni problem koji je ekvivalentan problemu dodjeljivanja. Razmislite o mogućnosti sparivanja n objekata s n osobama kroz tržišni mehanizam, promatrajući svaku osobu kao ekonomskog agenta[1] koji djeluje u svom najboljem interesu. Pretpostavimo da objekt j ima cijenu p_j i da osoba koja prima objekt mora platiti cijenu p_j. Zatim, (mrežna) vrijednost objekta j za osobu i je a_{ij} - p_j, i logično, svaka osoba bi željela biti dodijeljena objektu j_i s maksimalnom vrijednosti, odnosno s

$$a_{ij_i} - p_{j_i} = \max_{j \in A(i)} \{a_{ij} - p_j\}. \tag{1}$$

Ekonomski sustav bi bio u ravnoteži kada niti jedna osoba ne bi imala potrebu tražiti neki drugi objekt.

Ravnotežna dodjeljivanja i cijene prirodno su od velikog interesa za ekonomiste, ali postoji i fundamentalna veza s problemom dodjeljivanja; ispada da ravnotežni problem dodjeljivanja nudi maksimalnu ukupnu korist (i time rješava problem dodjeljivanja), dok odgovarajući skup cijena rješava pripadni dualni problem.
Ovo je posljedica teorema dualnosti u linearnom programiranju. U terminologiji linearnog programiranja, odnos (1) poznat je kao *uvjet komplementarnosti* (eng. complementary slackness, kraće: CS).

Propozicija 1.0.1. *Ako dopustivo dodjeljivanje S i skup cijena $\bar{p}$ zadovoljavaju uvjet komplementarnosti (1) za sve osobe i, tada je dodjeljivanje optimalno i vektor cijena je optimalno rješenje dualnog problema, koji glasi: minimizirati po $p \in \mathbb{R}^n$ funkciju troška*

$$g(p) = \sum_{i=1}^{n} q_i(p) + \sum_{j=1}^{n} p_j,$$

pri čemu su funkcije q_i dane s

$$q_i(p) = \max_{j \in A(i)} \{a_{ij} - p_j\}, \quad i = 1, ..., n.$$

Nadalje, vrijednosti optimalnog dodjeljivanja i optimalnog troška su jednake.

[1] osoba, tvrtka ili organizacija koja utječe na ekonomiju proizvodnjom, kupnjom ili prodajom

Dokaz. Ukupna vrijednost svakog dopustivog dodjeljivanja $\{(i, k_i)|i = 1, ..n\}$ zadovoljava

$$\sum_{i=1}^{n} a_{ik_i} = \sum_{i=1}^{n}(a_{ik_i} - p_{k_i}) + \sum_{i=1}^{n} p_{k_i} \leq \sum_{i=1}^{n} \max_{j \in A(i)}\{a_{ij} - p_j\} + \sum_{j=1}^{n} p_j,$$

za svaki skup cijena $\{p_j|j = 1, ..., n\}$. S druge strane, dano dodjeljivanje i skup cijena označeni s $\{(i, j_i)|i = 1, ..n\}$ i $\{(\bar{p}_j|j = 1, ..n\}$ zadovoljavaju CS-uvjet. Stoga imamo

$$a_{ij_i} - \bar{p_{j_i}} = \max_{j \in A(i)}\{a_{ij} - \bar{p}_j\}, \quad i = 1, ..., n.$$

Sada, za dodjeljivanje S vrijedi

$$g(\bar{p}) = \sum_{i=1}^{n}(a_{i_{j_i}} - \bar{p}_{j_i}) + \sum_{j=1}^{n} \bar{p}_j = \sum_{i=1}^{n}(a_{i_{j_i}} - \bar{p}_{j_i}) + \sum_{i=1}^{n} \bar{p}_{j_i} = \sum_{i=1}^{n} a_{ij_i},$$

dok je za neko drugo dodjeljivanje K, $g(\bar{p}) \geq \sum_{i=1}^{n} a_{iki}$, pa je S optimalno dodjeljivanje. Također, za svaki p vrijedi $g(p) \geq \sum_{i=1}^{n} a_{ij_i} = g(\bar{p})$, pa je $\bar{p}$ točka minimuma funkcije g. Dakle, dodjeljivanje $\{(i, j_i)|i = 1, ..n\}$ je optimalno za primarni problem, a skup cijena $\{(\bar{p}_j|j = 1, ..n\}$ je optimalan za dualni problem. Nadalje, te dvije optimalne vrijednosti su jednake. □

1.1 Naivni aukcijski algoritam

Razmotrimo prirodan proces pronalaženja ravnotežnog dodjeljivanja i vektora cijena. Nazvati ćemo ovaj proces *naivni aukcijski algoritam*, jer ima ozbiljnu manu, što ćemo vidjeti uskoro. Ipak, taj će nedostatak motivirati izradu sofisticiranijeg i ispravnijeg algoritma. Naivni aukcijski algoritam se nastavlja u iteracijama i generira niz vektora cijena i dodjeljivanja. Na početku svake iteracije, CS-uvjet

$$a_{ij_i} - p_{j_i} = \max_{j \in A(i)}\{a_{ij} - p_j\}. \tag{2}$$

je zadovoljen za sve parove (i, j) iz dodjeljivanja. Ako su sve osobe dodijeljene, odnosno dodjeljivanje je dopustivo, algoritam završava. U suprotnom se odabire neprazan podskup I osoba i, koje nisu dodijeljene, te se na njemu radi daljnja analiza.

Tipična iteracija algoritma naivne aukcije

Neka je I neprazan podskup osoba i, koje nisu dodijeljene.

Faza licitiranja: Svaka osoba $i \in I$ traži objekt j_i koji nudi maksimalnu vrijednost, dan s

$$j_i \in \arg\max_{j \in A(i)} \{a_{ij} - p_j\} \tag{3}$$

i izračunava povećanje ponude

$$\gamma_i = v_i - w_i \tag{4}$$

gdje je v_i najbolja vrijednost objekta

$$v_i = a_{ij_i} - p_{j_i}, \tag{5}$$

a w_i najbolja vrijednost koju nude drugi objekti osim j_i

$$w_i = \max_{j \in A(i), j \neq j_i} \{a_{ij} - p_j\} \tag{6}$$

Ako je j_i jedini objekt u $A(i)$, tada za w_i uzimamo -∞, ili u računalne svrhe, broj puno manji od v_i.

Za svaki objekt j, neka $P(j)$ bude skup osoba od kojih je j primio ponudu u fazi licitacije.

Faza dodjele: Svaki objekt j koji je odabran kao najbolji objekt za neprazni podskup $P(j)$ osoba iz I, određuje najboljeg ponuditelja

$$i_j \in \arg\max_{i \in P(j)} \gamma_i, \tag{7}$$

povećava svoju cijenu p_j za $\max_{i \in P(j)} \gamma_i$, i dodjeljuje se najboljem ponuditelju i_j, a osoba koja je dodijeljena j na početku iteracije (ako postoji) postaje nedodijeljena. Algoritam se nastavlja s nizom iteracija sve dok svim osobama nije dodijeljen objekt. Uočimo da γ_i ne može biti negativna kako je $v_i \geq w_i$ (usporedite jednakosti (5) i (6)), tako da cijene objekata imaju tendenciju povećati se. Zapravo, kad je i jedini ponuditelj, γ_i je najveće povećanje ponude (eng. bidding increment) za koje je očuvan CS-uvjet obzirom na dodjeljivanje i objektu kojeg preferira.

Baš kao kod prave aukcije, povećanje i cijena će potaknuti konkurenciju čineći ponuditeljev željeni objekt manje atraktivnim za druge potencijalne ponuditelje.

Uočimo da postoji određena sloboda pri odabiru podskupa osoba I koje se natječu tijekom iteracije. Jedna mogućnost je da se I sastoji od jedne nedodijeljene osobe. Ova verzija, poznata kao Gauss-Seidelova verzija, zbog svoje sličnosti s Gauss-Seidelovim metodama za rješavanje sustava nelinearnih jednadžbi, obično radi najbolje u serijskom računalnom okruženju. Verzija kod koje se I sastoji od svih nedodijeljenih osoba najbolje funkcionira kod paralelnog računanja; poznata je kao Jacobijeva verzija zbog svoje sličnosti s Jacobijevom metodom za rješavanje sustava nelinearnih jednadžbi.

Poglavlje 2

Uvjet ϵ-komplentarnosti i aukcijski algoritam

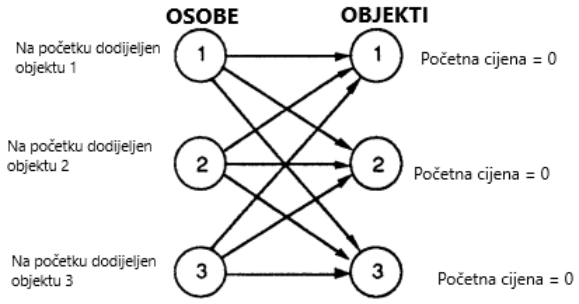
Nažalost, naivni aukcijski algoritam ne radi uvijek (iako je izvrstan inicijalizacijski postupak za druge metode koji se temelje na prilagodbi cijena, npr. primarna-dualna metoda ili relaksacija).

Poteškoća je što je povećanje ponude γ_i nula kada više od jednog objekta nudi maksimalnu vrijednost za ponuditelja i. Kao rezultat toga, može se stvoriti situacija u kojoj se više osoba natječe za manji broj jednako poželjni objekta bez podizanja cijena, stvarajući tako ciklus koji nikada ne završava; vidite sliku 1.

Da bismo prekinuli takve cikluse, uvodimo mehanizam perturbacije, motiviran stvarnim dražbama gdje se kod svake nove ponude za objekt mora povećati cijena objekta minimalnim pozitivnim prirastom, a ponuditelji ponekad moraju riskirati kako bi se domogli željenih objekta. Posebno, možemo fiksirati pozitivan skalar ϵ i reći da dodjeljivanje i vektor cijene p zadovoljavaju uvjet ϵ-komplementarnosti (ili kraće ϵ-CS) ako

$$a_{ij_i} - p_{ji} \geq \max_{j \in A(i)} \{a_{ij} - p_j\} - \epsilon, \tag{8}$$

za svaki par (i, j_i). Riječima, kako bi se zadovoljili ϵ-CS, sve dodijeljene osobe moraju biti dodijeljene objektima koji su za njih najbolji do na ϵ.



Početak iteracije #	Cijene objekata	Dodijeljeni parovi	Ponuditelj	Preferirani objekt	Povećanje ponude
1	0,0,0	(1,1), (2,2)	3	2	0
2	0,0,0	(1,1), (3,2)	2	2	0
3	0,0,0	(1,1), (2,2)	3	2	0

Slika 1: Ovdje je $a_{ij} = C > 0$ za sve (i, j) takve da je $i = 1, 2, 3$ i $j = 1, 2$, te je $a_{ij} = 0$ za sve $i = 1, 2, 3$ i $j = 3$. Prikazano je kako aukcijski algoritam možda nikada ne završi za problem s tri osobe i tri objekta. Dakle, objekti 1 i 2 pružaju korist $C > 0$ svim osobama, dok objekt 3 pruža korist 0 svim osobama. Osobe 2 i 3 podjednako žele objekte 1 i 2, stoga se stvara ciklus ukoliko one naizmjenično daju iste ponude (bez da ih mijenjaju odnosno. $\gamma_i = 0$).

2.1 Aukcijski algoritam

Sada preoblikujemo prethodni postupak dražbe kako bi povećanje ponude uvijek bilo barem jednako ϵ. Dobivena metoda, aukcijski algoritam, jednaka je naivnom aukcijskom algoritmu, osim što je povećanje ponude γ_i

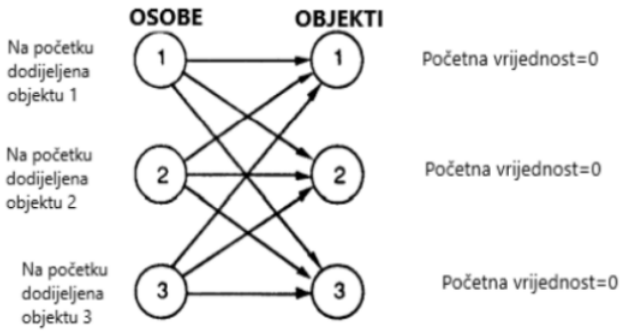
$$\gamma_i = v_i - w_i + \epsilon, \tag{9}$$

(za razliku od $\gamma_i = v_i - w_i$ kako imamo u jednakosti (4)). S ovim izborom, uvjet ϵ-CS je zadovoljen te je povećanje $\gamma_i = v_i - w_i + \epsilon$ najveće povećanje s ovim svojstvom. Algoritam bi funkcionirao i za manje povećanje γ_i dok god je $\gamma_i \geq \epsilon$, ali korištenje najvećeg mogućeg povećanja ubrzava algoritam. Ovo je slično stvarnim aukcijama koje imaju tendenciju brže završiti kada je nadmetanje agresivnije.

Može se pokazati da ovaj reformulirani aukcijski postupak završava u konačnom broju iteracija, pri čemu je dodjeljivanje nužno dopustivo i skup cijena zadovoljava ϵ-CS. Da bismo to vidjeli u slučaju potpunog gustog problema (kada $\mathcal{A}$ sadrži sve parove (i, j)), uočite da ako objekt prima ponudu u k iteracija, njegova cijena mora premašiti svoju početnu cijenu za najmanje $k\epsilon$. Dakle, za dovoljno velik k, objekt će postati dovoljno "skup" da ga se smatra inferiornim nekom objektu koji do sada nije dobio ponudu. Stoga, u ograničenom broju iteracija, neki objekti će primiti ponudu ili ponude dok neki drugi objekt možda još nije primio niti jednu ponudu. No nakon što svi objekti dobiju najmanje jednu ponudu, aukcija prestaje. Dakle, aukcijski algoritam mora završiti, a prethodni argument pokazuje, u slučaju da je početna cijena jednaka nuli, ukupan broj iteracija u kojima objekt prima ponude nije veći od $\dfrac{\max_{(i,j)} |a_{ij}|}{\epsilon}$.

Ako se u svakoj iteraciji dobije ponuda od samo jedne osobe, ukupan broj iteracija je ograničen s prethodnom procjenom broja iteracija pomnoženom s n, a budući da svaka ponuda zahtijeva $O(n)$ operacija, vrijeme izvođenja algoritma je $O(\dfrac{n^2 \max_{(i,j)} |a_{ij}|}{\epsilon})$.

Ovaj dokaz se može generalizirati i u slučaju rijetkog problema (onog gdje je skup osoba-objekt parova koji se mogu dodijeliti ograničen), sve dok je problem dopustiv; Slika 2 prikazuje kako aukcijski algoritam, temeljen na povećanju ponude $\gamma_i = v_i - w_i + \epsilon$ (jednakost (9)) nadilazi ciklički problem na Slici 1.



Na početku iteracije #	Cijene objekata	Dodijeljeni parovi	Ponuditelj	Preferirani objekt	Povećanje ponude
1	0,0,0	(1,1), (2,2)	3	2	ϵ
2	0,ϵ,0	(1,1), (3,2)	2	1	2ϵ
3	2ϵ,ϵ,0	(2,1), (3,2)	1	2	2ϵ
4	2ϵ,3ϵ,0	(1,2), (2,1)	3	1	2ϵ
5	4ϵ,3ϵ,0	(1,2), (3,1)	2	2	2ϵ
6	...	...	...	...	...

Slika 2: Ovdje je $a_{ij} = C > 0$ za sve (i, j) takve da je $i = 1, 2, 3$ i $j = 1, 2$, te je $a_{ij} = 0$ za sve $i = 1, 2, 3$ i $j = 3$. Prikazano je kako aukcijski algoritam nadilazi problem ciklusa sa Slike 1 tako da uvodi povećanje ponude za barem ϵ. Tablica pokazuje jedan mogući slijed ponuda i dodjeljivanja generiranih aukcijskim algoritmom koji započinje sa svim cijenama jednakim nula. U svim iteracijama, osim zadnje, osoba dodijeljena objektu 3 se natječe za objekt 1 ili 2 povećavajući svoju ponudu za ϵ u prvoj iteraciji, a za 2ϵ u svakoj sljedećoj iteraciji. U zadnjoj iteraciji, nakon što cijene od 1 i 2 narastu do ili iznad C, objekt 3 dobiva ponudu i aukcija završava.

Kada aukcijski algoritam završi, imamo dodjeljivanje koje zadovoljava ϵ-CS, ali je li to dodjeljivanje optimalno? U ovom slučaju ovaj odgovor strogo ovisi o veličini od ϵ. Na pravoj aukciji, razborit ponuditelj neće postaviti pretjerano visoku ponudu zbog straha da bi mogao osvojiti objekt po nepotrebno visokoj cijeni. U skladu s tim, možemo pokazati da ako je ϵ mali, krajnje dodjeljivanje će biti "gotovo optimalno". Posebno, sljedeća propozicija pokazuje da je ukupni trošak konačnog dodjeljivanja optimalan unutar $n\epsilon$. Ideja je da kada dopustivo dodjeljivanje i skup cijena zadovoljavaju ϵ-CS, oni također zadovoljavaju

i CS (stoga su i primarni i dualni problem optimalni) za malo izmijenjen problem gdje su svi troškovi a_{ij} isti kao i ranije, osim troškova n pridruženih parova, koje su izmijenjene u iznosu ne većem od ϵ.

Propozicija 2.1.1. *Za dopustivo dodjeljivanje S koje zadovoljava ϵ-CS zajedno s nekim vektorom cijena vrijedi da je razlika optimalne ukupne koristi od S i ukupne koristi najviše $n\epsilon$.*

Dokaz. Neka je A^* optimalna ukupna korist dodjeljivanja

$$A^* = \max_{\substack{k_i, i=1,\dots,n \\ k_i \neq k_m \ ako \ i \neq m}} \sum_{i=1}^{n} a_{ik_i}$$

i neka je optimalna dualna cijena

$$D^* = \min_{\substack{p_j \\ j=1,\dots,n}} \{\sum_{1}^{n} \max_{j \in A(i)} \{a_{ij} - p_j\} + \sum_{j=1}^{n} p_j\}.$$

Ako je $\{(i, j)|i = 1, ..., n\}$ dano dodjeljivanje koje zadovoljava uvjet ϵ-komplementarnosti zajedno s vektorom cijena $\bar{p}$, imamo

$$\max_{j \in A(i)} \{a_{ij} - \bar{p}_j\} - \epsilon \leq a_{ij_i} - \bar{p}_{j_i}.$$

Uvođenjem navedene relacije za sve i, vidimo da je

$$D^* \leq \sum_{i=1}^{n} (\max_{j \in A(i)} \{a_{ij} - \bar{p}_j\} + \bar{p}_{j_i}) \leq \sum_{i=1}^{n} a_{ij_i} + n\epsilon \leq A^* + n\epsilon$$

Kako su vrijednosti optimalnog dodjeljivanja i optimalnog troška dualnog problema jednake imamo $A^* = D^*$. Stoga, slijedi da je razlika ukupne koristi dodjeljivanja $\sum_{i=1}^{n}$ i vrijednosti A^* najviše $n\epsilon$ te isto vrijedi za razliku dualnog troška $\bar{p}$ i optimalnog dualnog troška.

□

Pretpostavimo sada da su svi a_{ij} cijeli brojevi, zbog jednostavnosti (ako su a_{ij} racionalni brojevi, mogu se transformirati do cijelih brojeva množenjem s najmanjim zajedničkim nazivnikom). Tada je i ukupna korist dodjeljivanje cijeli broj, pa ako je $n\epsilon < 1$, svako potpuno dodjeljivanje koje je unutar $n\epsilon$ optimalno mora biti optimalno. Slijedi, ako je $\epsilon < \frac{1}{n}$, a svi a_{ij} su cijeli brojevi, dodjeljivanje dobiveno nakon završetka aukcijskog algoritma je optimalno. Navedimo ovaj rezultat kao propoziciju.

Propozicija 2.1.2. *Promatramo dopustivo dodjeljivanje gdje su svi a_{ij} cijeli brojevi. Ako je*

$$\epsilon < \frac{1}{n}$$

aukcijski algoritam završava u konačnom broju iteracija s optimalnim dodjeljivanjem.

Dokaz se oslanja na sljedeće činjenice:

(a) Jednom kada se neki objekt dodijeli, ostaje dodijeljen tijekom ostatka algoritma. Nadalje, osim na kraju, uvijek će postojati barem jedan objekt koji nikad nije bio dodijeljen i ima cijenu jednaku početnoj cijeni. Razlog tomu je što faza ponude i dodjele mogu rezultirati preraspodjelom već dodijeljenog objekta nekoj drugoj osobi, ali ne mogu rezultirati da objekt postaje nedodijeljen.

(b) Svaki puta kada objekt dobije ponudu, njegova se cijena poveća za najmanje ϵ (Vidite jednakost (9)). Stoga, ako objekt prima ponuda beskonačni broj puta, cijena se povećava k ∞.

(c) Neka je $|A(i)|$ broj objekata u skupu $A(i)$, najbolja vrijednost objekta za ponuditelja i definirana je s

$$v_i = \max_{j \in A(i)} \{a_{ij} - p_j\}$$

i smanjuje se za najmanje ϵ. Razlog tomu je što ponuda osobe i smanjuje v_i za najmanje ϵ ili ga ostavlja nepromijenjenim ako postoji više objekata j za koje se postiže maksimum u prethodnoj jednakosti. Međutim, u drugom slučaju, cijena objekta j_i koji prima ponudu od osobe i će narasti za najmanje ϵ te objekt j_i neće primiti novu ponudu od osobe i dok se v_i ne smanji za najmanje ϵ. Zaključak je da ako osoba i daje ponudu beskonačno mnogo puta, v_i se mora smanjiti do $-\infty$.
Raspravimo sada o kontradikciji. Ako nije došlo do zaustavljanja algoritma, skup J^∞ objekata koji su primili beskonačan broj ponuda je neprazan. Skup I^∞ osoba koje su dale ponudu beskonačno mnogo puta je također neprazan. Kao što tvrdimo u (b), cijene objekata u J^∞ moraju težiti u ∞, dok prema (c), skalari $v_i = \max_{j \in A(i)} \{a_{ij} - p_j\}$ trebaju težiti u $-\infty$ za sve osobe $i \in I^\infty$. Dakle, $a_{ij} - p_j$ teži u $-\infty$ za sve $j \in A(i)$ što znači da

$$A(i) \subset J^\infty, \quad \forall i \in I^\infty.$$

Uvjet ϵ-komplementarnosti (8) navodi da je $a_{ij} - p_j \geq v_i - \epsilon$ za svaki dodijeljeni par (i, j), stoga nakon konačnog broja iteracija svaki objekt iz J^∞ može biti dodijeljen samo osobama iz I^∞. Budući da će nakon konačnog broja iteracija, na početku svake iteracije, barem jedna osoba iz I^∞ biti nedodijeljena, slijedi da je broj osoba u I^∞ strogo veći od broja objekata u J^∞. Ovo je kontradikcija s postojanjem dopustivog dodjeljivanja, jer se prema prethodnoj

relacije, osobe u I^∞ mogu dodijeliti samo objektima u J^∞. Stoga, algoritam mora završiti. Dopustivo dodjeljivanje dobiveno nakon prestanka zadovoljava ϵ-CS, pa prema Propoziciji 2.1.1, dodjeljivanje je do na $n\epsilon$ optimalno.

2.2 ϵ-skaliranje

Količina posla potrebnog za prekid aukcijskog algoritma može snažno ovisiti o vrijednosti ϵ i o maksimalnoj apsolutnoj koristi C danoj s

$$C = \max_{(i,j)\in\mathcal{A}} |a_{ij}| \tag{10}$$

Uglavnom, za mnoge vrste problema, broj iteracija do prestanka algoritma ima tendenciju biti proporcionalan s C/ϵ kao što smo ranije spomenuli za potpuno guste probleme. Za mali ϵ, metoda je osjetljiva na "ratove cijena", to jest, dugotrajne sekvence malih dizanja cijena od strane grupe osoba koje se natječu za manji broj otprilike jednako poželjnih objekata.

Također uočite da postoji ovisnost o početnim cijenama; ako su te cijene "blizu optimalnih", očekujemo da će broj iteracija za rješavanje problema biti relativno mali. Prethodna opažanja ukazuju na ideju ϵ-skaliranja koja se sastoji od primjene algoritma nekoliko puta, počevši s velikim ϵ i sukcesivno ga smanjujući do konačne vrijednosti koja je manja od neke kritične vrijednosti (na primjer, $\frac{1}{n}$, kada su beneficije a_{ij} cijeli brojevi). Tipični faktori smanjenja vrijednosti ϵ nakon svake faza skaliranja su reda od 4 do 10.

Svaka primjena algoritma pruža dobre početne cijene za sljedeću primjenu. Na temelju opsežnih eksperimentiranja koja su potvrdila učinkovitost ϵ-skaliranja kod mnogih vrsta problema dodjeljivanja, ono je uključeno i u originalnu ideju aukcijskog algoritma. Konkretno, ϵ-skaliranje je obično korisno za rijetke probleme. U određivanju je li ϵ-skaliranje potrebno je također važna i troškovna struktura problema.

Za cjelobrojne podatke, može se pokazati da je najduže moguće trajanje izvršavanja aukcijskog algoritma koristeći skaliranje i odgovarajuću strukturu podataka jednako $O(nA\log(nC))$. Na temelju eksperimenata, vrijeme trajanja algoritma za slučajno generirane probleme očito raste proporcionalno nečemu kao $A\log n$ ili $A\log n\log(nC)$.

2.3 Suočavanje s nedopustivosti

Budući da do završetka može doći samo ako se radi o dopustivom dodjeljivanju, kada je problem nedopustiv, aukcijski algoritam će nastaviti iteriranje (dok se korisnik pita je li problem nemoguć ili ga je teško riješiti). Dakle, za probleme u kojima nije a priori poznato je li dodjeljivanje dopustivo, aukcijski algoritam se mora nadopuniti s mehanizmom za otkrivanje nedopustivosti (eng.infeasibility). Postoji nekoliko takvih mehanizama.

Osnovni rezultat vezan za (simetrične i asimetrične) probleme dodjeljivanja je da s obzirom na nedopustivost dodjeljivanja S, postoje dvije međusobno isključive mogućnosti:

(a) S ima maksimalnu kardinalnost, odnosno ne postoji dodjeljivanje koje ima više dodijeljenih osoba nego S.

(b) Postoji neka nedodijeljena osoba i i neki nedodijeljeni objekt j te put proširenja (eng.augmenting path) obzirom na S koji počinje s i i završava s j odnosno

$$(i, j_1, i_1, ..., j_m, i_m, j)$$

tako da je $j_1 \in A(i)$,...,$j \in A(i_m)$ te je j_k dodijeljen i_k-u za $k = 1, ..., m$.

Ovaj rezultat se može dokazati na više načina. Na primjer, uvođenjem u graf problema dodjeljivanja novi čvor izvor s povezan sa svim čvorovima osoba i novi čvor ponor t povezan sa svim čvorovima objekata. Možemo vidjeti da je problem pronalaženja dodjeljivanja maksimalne kardinalnosti jednak problemu pronalaženje maksimalnog toka od s do t, a rezultat slijedi pomoću teorema dualnosti za problem maksimalnog toka te pripadne analize.

Posljedica posljednjeg rezultata je da za dani dopustivi problem dodjeljivanja i nedopustivo dodjeljivanje S, za svaku osobu i koja nije dodijeljena u S, postoji put proširenja obzirom na S s početkom u i. (Da bismo to dokazali, izmijenimo S tako ta dodijelimo sve osobe koje nisu dodijeljena u S, osim i, fiktivnom objektu, a zatim primijenimo prethodni rezultat.)

Jedan kriterij koji se može koristiti za otkrivanje nedopustivosti temelji se na maksimalnim vrijednostima

$$v_i = \max_{j \in A(i)} \{a_{ij} - p_j\}.$$

Ispada da će se tijekom algoritma aukcije sve ove vrijednosti ograničiti odozdo s unaprijed izračunatom granicom kada je problem dopustiv, no neke od tih vrijednosti će na kraju biti smanjene i ispod ove granice ako je problem nedopustiv. Konkretno, pretpostavimo da se primjenjuje aukcijski algoritam s početnim cijenama objekta $\{p_j^0\}$. Tada se može pokazati da za bilo koji i, ako je osoba i nedodijeljena obzirom na trenutno dodjeljivanje S te postoji put proširenja s obzirom na S koji počinje u i, imamo

$$v_i \geq -(2n-1)C - (n-1)\epsilon - \max_j \{p_j^0\}, \tag{11}$$

gdje je $C = \max_{(i,j) \in \mathcal{A}} |a_{ij}|$. Dokaz se dobiva dodavanjem ϵ-CS uvjeta uz put proširenja. Ako je problem dopustiv, kao što je ranije spomenuto, postoji put proširenja koji počinje s nedodijeljenom osobom svaki puta, tako da će se donja granica (11) od v_i očuvati za sve nedodijeljene osobe i kroz čitav algoritam aukcije. S druge strane, ako je problem nedopustiv, neke će osobe i dati ponude beskonačno puta, a odgovarajuće vrijednosti v_i će se smanjivati prema -∞. Tako da možemo primijeniti aukcijski algoritam i pratiti vrijednosti v_i kako se smanjuju. Ako neki v_i dođe ispod svoje donje granice, znamo da je problem nedopustiv.

Nažalost, moguće je da će za neke v_i biti potreban veliki broj iteracija kako bi došli do svoje donje granice. Alternativna metoda za otkrivanje nedopustivosti bi bila pretvaranje problema u dopustiv problem dodavanjem skupa umjetnih (artificijelnih) parova $\bar{\mathcal{A}}$ izvornom skupu $\mathcal{A}$. Beneficije tih parova bi trebale biti vrlo male, tako da nitko od njih ne sudjeluje u optimalnom dodjeljivanju, osim ako je problem nedopustiv. Konkretno, može se pokazati da ako je izvorni problem dopustiv, niti jedan par $(i, j) \in \bar{\mathcal{A}}$ neće sudjelovati u optimalnom dodjeljivanju, pod uvjetom da

$$a_{ij} < -(2n-1)C, \qquad \forall (i,j) \in \bar{\mathcal{A}}, \tag{12}$$

gdje je $C = \max_{(i,j)\in\mathcal{A}} |a_{ij}|$. Kako bi ovo dokazali pomoću kontradikcije, pretpostavimo da dodavanjem u skup $\mathcal{A}$ skup umjetnih parova $\bar{\mathcal{A}}$ stvaramo optimalno dodjeljivanje S^* koje sadrži neprazan podskup $\bar{S}$ umjetnih parova. Zatim, za svako dodjeljivanje S koje se sastoji isključivo od parova iz originalnog skupa $\mathcal{A}$ moramo imati

$$\sum_{(i,j)\in\bar{S}} a_{ij} + \sum_{(i,j)\in S^*-\bar{S}} a_{ij} \geq \sum_{(i,j)\in S} a_{ij},$$

iz čega imamo

$$\sum_{(i,j)\in\bar{S}} a_{ij} \geq \sum_{(i,j)\in S} a_{ij} - \sum_{(i,j)\in S^*-\bar{S}} a_{ij} \geq -(2n-1)C.$$

Što je kontradikcija s (12). Uočimo da ako je $a_{ij} \geq 0$ za svaki $(i, j) \in \mathcal{A}$, prethodni argument se može izmijeniti tako da pokazuje kako je, za sve umjetne parove (i, j), dovoljno imati $a_{ij} < -(n-1)C$.

S druge strane, dodavanje umjetnih parova s beneficijama $-(2n-1)C$ kao u formuli (12) proširuje raspon troškova problema faktorom $(2n-1)$. U kontekstu ϵ-skaliranja to zahtijeva mnogo veću početnu vrijednost od ϵ i odgovarajući veliki broj faza ϵ-skaliranja. Ako je problem dopustiv te se dodatne faze skaliranja gube. Dakle, kod problema za koje očekujemo da će biti dopustivi je možda bolje umjetnim parovima dodijeliti beneficije koje su reda $-C$, a zatim te beneficije postupno spuštati prema pragu $-(2n-1)C$ ukoliko umjetni parovi ustraju u dodjeljivanjima dobivenim aukcijskim algoritmom. Ovaj postupak silaznog skaliranja beneficija umjetnih parova se može na više načina uklopiti u postupak ϵ-skaliranja.

Još jedna metoda za otkrivanje nedopustivosti se temelji na sljedećem svojstvu: čak i kada je problem nedopustiv, aukcijski algoritam će naći dodjeljivanje maksimalne kardinalnosti u konačnom broju iteracija ukoliko su nedodijeljene osobe raspoređene za davanje ponude u cikličkom redoslijedu ili osigurava da svaka osoba dobije priliku dati ponudu

barem jednom u nekom fiksnom broju iteracija. Dokaz ovoga se temelji na donjoj granici (11) i svojstvu dodjeljivanja maksimalne kardinalnosti navedenom ranije. Konkretno, da trenutno dodjeljivanje nikada ne postiže maksimalnu kardinalnost bi postojala nedodijeljena osoba i put koji počinje u i i povećava se s obzirom na trenutno dodjeljivanje. Dodavanjem ϵ-CS uvjeta tom putu vidimo da će v_i uvijek zadovoljavati donju granicu (11) što je kontradikcija, jer će v_i težiti u $-\infty$ za sve i koji beskonačno mnogo puta daju ponudu.

Pretpostavimo sada da povremeno prekidamo aukcijski algoritam i provjerimo postoji li put proširenja od neke nedodijeljene osobe do nekog nedodijeljenog objekta. Tako će, jednom kada kardinalnost trenutnog dodjeljivanja postane maksimalna, ali je manja od n, ova provjera utvrditi da je problem nedopustiv. Tako promijenjeni aukcijski algoritam zajamčeno pronalazi dopustiva dodjeljivanja i skup cijena zadovoljava ϵ-CS ili utvrđuje da je problem nedopustiv i dobiva dodjeljivanje maksimalne kardinalnosti. U posljednjem slučaju, može se pokazati da će izvorni (simetrični) problem biti odvojeni u dvije komponente koje odgovaraju (asimetričnim) problemima dodjeljivanja. Tada je moguće koristite aukcijske algoritme za asimetrične probleme kako bismo optimizirali dodjeljivanje unutar svake komponente i dobili optimalano dodjeljivanje unutar klase svih dodjeljivanja maksimalne kardinalnosti.

2.4 Dobit i obrnuta aukcija

Budući da je problem dodjeljivanja simetričan, moguće je zamijeniti uloge osoba i objekata. To dovodi do *obrnute aukcije* gdje se objekti natječu za osobe tako da nude popuste (odnosno smanjujući svoje cijene). Ugrubo, obzirom na vektor cijena p, možemo promatrati mrežnu vrijednost najboljih objekata za osobu i

$$\max_{j \in A(i)} \{a_{ij} - p_j\} \tag{13}$$

kao *dobit* osobe i. Kada objekti smanjuju svoje cijene, oni nastoje povećati dobit osoba. Dakle, za osobe dobit ima analognu ulogu kao cijena kod objekata. Stoga, obrnutu aukciju možemo opisati na dva različita načina; prvi gdje nedodijeljeni objekti smanjuju svoje cijene što je više moguće kako bi privukli osobu bez narušavanja ϵ-CS uvjeta, te drugi kod kojeg nedodijeljeni objekti odabiru najbolju osobu i podižu njezinu dobit što je više moguće bez narušavanja ϵ-CS uvjeta. Drugi opis je analitički više prikladan, budući da će s ovim opisom aukcije unaprijed i obrnute aukcije biti matematički ekvivalentne.

Uvedimo sada profitnu varijablu π_i za svaku osobu i pri čemu uzmimo u obzir sljedeći ϵ-CS uvjet za dodjeljivanje S i vektor dobiti π:

$$a_{ij} - \pi_i \geq \max_{k \in B(j)} \{a_{kj} - \pi_k\} - \epsilon, \qquad \forall (i,j) \in S \tag{14}$$

gdje je $B(j) = \{i|(i, j) \in \mathcal{A}\}$ skup (pretpostavimo neprazan) osoba koje mogu biti dodijeljene objektu j. Odnos između profitne varijable π_i i izraza $\max_{j\in A(i)}\{a_{ij} - p_j\}$ će postati vidljiv kasnije kada ćemo promatrati ϵ-CS uvjet u nešto drugačijem stanju koje uključuje i cijene i dobit. Uočite simetriju između ϵ-CS uvjeta (14) s odgovarajućim uvjetom za cijene (8).

Obrnuti aukcijski algoritam održava da na početku svake iteracije dodjeljivanje S i vektor dobiti π zadovoljavaju ϵ-CS uvjet (14), a završava kada je dodjeljivanje dopustivo.

Tipična iteracija obrnute aukcije

Neka je J neprazan podskup objekata j koji su nedodijeljeni.

Faza ponude: Za svaki objekt $j \in J$ pronađi "najbolju" osobu i_j tako da je

$$i_j = \arg\max_{i\in B(j)}\{a_{ij} - \pi_i\},$$

i odgovarajuća vrijednost

$$\beta_j = \max_{i\in B(j)}\{a_{ij} - \pi_i\},$$

te nađi

$$\omega_j = \max_{i\in B(j), i\neq i_j}\{a_{ij} - \pi_i\} \tag{15}$$

(Ako je i_j jedina osoba u $B(j)$ tada definiramo ω_j kao -∞ ili , u računalne svrhe, kao broj mnogo manji od β_j.)

Faza dodjele:Svaka osoba i koja je odabrana kao najbolja osoba za neprazni podskup $P(i)$ objekata iz J, određuje najboljeg ponuditelja

$$j_i = \arg\max_{j\in P(i)}\{\beta_j - \omega_j + \epsilon\}, \tag{16}$$

povećava π_i prema najvećem mogućem povećanju ponude $\max_{j\in P(i)}\{\beta_j - \omega_j + \epsilon\}$ i dodjeljuje se najboljem ponuditelju j_i, a objekt koji je bio dodijeljen i na početku iteracije (ako postoji) postaje nedodijeljen.

Uočite da je obrnuta aukcija identična aukciji unaprijed samo sa zamijenjenim ulogama osoba i objekata, te zamijenjenim ulogama dobiti i cijena. Dakle, korištenjem odgovarajućih rezultata za aukcije (unaprijed) (Propozicije 2.1.1 i 2.1.2), imamo:

Propozicija 2.4.1. *Ako postoji najmanje jedno dopustivo dodjeljivanje, algoritam obrnute aukcije završava u konačnom broju iteracija. Dopustivo dodjeljivanje dobiveno nakon završetka je optimalno do na $n\epsilon$ (i optimalno je ako su podaci unutar problema cijeli brojevi te $\epsilon < 1/n$).*

2.5 Kombinacija aukcije unaprijed i obrnute aukcije

Jedan od razloga zašto nas zanima obrnuta aukcija je zbog izgradnje algoritama koji se prebacuju s aukcije unaprijed na obrnutu aukciju, i natrag. Takvi algoritmi moraju istodobno održavati da vektor cijene p zadovoljava ϵ-CS uvjet (8) i da vektor dobiti π zadovoljava ϵ-CS uvjet (14). U tu svrhu uvodimo ϵ-CS uvjet za par (π, p), koji će, kao što ćemo vidjeti, podrazumijevati i navedena dva. Održavanje ovog stanja je nužno za precizno prebacivanje između aukcije unaprijed i obrnute aukcije.

Kažemo da dodjeljivanje S i par (π, p) zadovoljavaju ϵ-CS ako

$$\pi_i + p_j \geq a_{ij} - \epsilon, \qquad \forall (i,j) \in \mathcal{A}, \tag{17a}$$

$$\pi_i + p_j = a_{ij}, \qquad \forall (i,j) \in S. \tag{17b}$$

Propozicija 2.5.1. *Pretpostavimo da dodjeljivanje S zajedno s parom dobit-cijena (π, p) zadovoljava ϵ-CS. Tada*

(a) S i π zadovoljavaju ϵ-CS uvjet

$$a_{ij} - \pi_i \geq \max_{k \in B(j)} \{a_{kj} - \pi_k\} - \epsilon, \qquad \forall (i,j) \in S. \tag{18}$$

(b) S i p zadovoljavaju ϵ-CS uvjet

$$a_{ij} - p_j \geq \max_{k \in A(i)} \{a_{ik} - p_k\} - \epsilon, \qquad \forall (i,j) \in S. \tag{19}$$

(c) Ako je S dopustivo, tada je S optimalno dodjeljivanje do na $n\epsilon$.

Dokaz. (a) S obzirom na jednakost (17b), za sve $(i,j) \in S$ imamo $p_j = a_{ij} - \pi_i$, stoga jednakost (17a) povlači $a_{ij} - \pi_i \geq a_{kj} - \pi_k - \epsilon$ za sve $k \in B(j)$. To pokazuje jednakost (18).

(b) Dokaz je isti kao i u (a) dijelu samo π i p zamijene uloge.

(c) Obzirom na dio (b), ϵ-CS uvjet (19) je zadovoljen, pa po Propoziciji 1.0.1 imamo da je S optimalan unutar $n\epsilon$.

□

Sada uvodimo kombinirani algoritam koji održava da dodjeljivanje S i par dobit-cijena (π, p) zadovoljavaju ϵ-CS uvjete (17). Algoritam završava kada je dodjeljivanje dopustivo. Uobičajeni način za inicijalizaciju algoritma tako da su ϵ-CS uvjeti zadovoljeni je

da uzmemo S prazan i p proizvoljan te odaberemo π kao funkciju od p preko relacije $\pi_i = \max_{k \in A(i)} \{a_{ik} - p_k\}$ za svaku osobu i.

Kombinirani unaprijed/obrnuti aukcijski algoritam

Korak 1: (Pokretanje aukcije unaprijed) Izvršite nekoliko iteracija algoritma aukcije unaprijed i na kraju svake iteracije (nakon povećanja cijena objekata koji su primili ponudu), postavite

$$\pi_i = a_{ij_i} - p_{j_i}, \tag{20}$$

za svaki par (i, j_i) koji se pridružio dodjeljivanju za vrijeme iteracije. Idite na korak 2.

Korak 2: (Pokretanje obrnute aukcije) Izvršite nekoliko iteracija algoritma obrnute aukcije i na kraju svake iteracije (nakon povećanja dobiti osoba koje su primile ponudu), postavite

$$p_j = a_{i_j j} - \pi_{i_j}, \tag{21}$$

za svaki par (i_j, j) koji se pridružio dodjeljivanju za vrijeme iteracije. Idite na korak 1.

Uočite da je dodatni račun kombiniranog algoritma naspram algoritma unaprijed ili obrnutog algoritma minimalan: potrebno je samo jedno ažuriranje dano s (20) ili (21) po iteraciji za svaki objekt ili osobu koji su primili ponudu tijekom iteracije. Važno je svojstvo da ažuriranje (20) i (21) održava ϵ-CS uvjete (17) za par (π, p), i stoga, prema Propoziciji 2.5.1, održava i potrebne ϵ-CS uvjete (18) i (19) za π i p. Ovo je navedeno u sljedećoj propoziciji.

Propozicija 2.5.2. *Ako dodjeljivanje i par dobit-cijena dostupni na početku iteracije, bilo aukcijskog algoritma unaprijed ili obrnutog, zadovoljavaju ϵ-CS uvjete (17), isto vrijedi i za dodjeljivanje i par dobit-cijena koji su dobiveni na kraju iteracije, pod uvjetom da se jednakost (20) koristi za ažuriranje π (u slučaju aukcije unaprijed), a jednakost (21) za ažuriranje p (u slučaju obrnute aukcije).*

Uočite da se tijekom aukcije unaprijed cijene objekata p_j povećavaju, dok se dobiti π_j smanjuju; kod obrnute aukcije se događa točno suprotno. Iz tog razloga, dokaz o prestanku algoritma korišten za aukciju unaprijed ne primjenjuje se na kombiniranu metodu. Moguće je konstruirati primjere dopustivih problema kod kojih kombinirana metoda nikada ne staje ako je prebacivanje između aukcije unaprijed i obrnute aukcije proizvoljno. Međutim, lako je osigurati da kombinirani algoritam ipak završava za dopustiv problem; dovoljno je osigurati da se prije prebacivanja obavlja neki "nepovratni napredak" između aukcije unaprijed i obrnute aukcije. Jedna mogućnost lake implementacije je da ne vršimo

prebacivanja dok barem još jedan par nije dodan dodjeljivanju. Na taj način može biti najviše $(n - 1)$ prebacivanja između Koraka 1 i Koraka 2 algoritma. Za dopustiv problem aukcija unaprijed i obrnuta aukcija, same po sebi, garantiraju završetak u konačnom broju koraka, stoga će posljednji korak završiti s dopustivim dodjeljivanjem koje zadovoljava ϵ-CS uvjet.

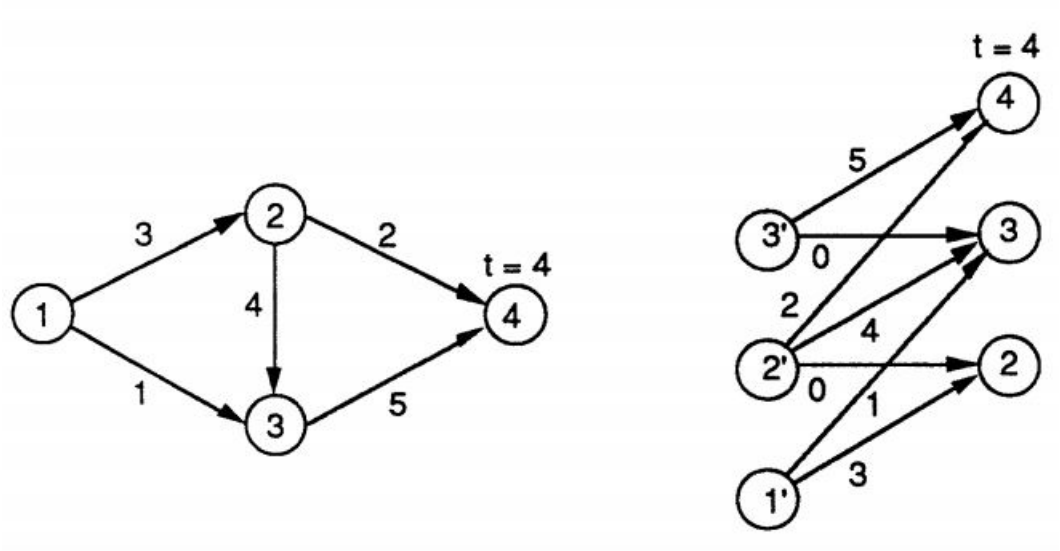
Kombinirani aukcijski algoritam obično radi značajno (i često dramatično) brže od algoritma unaprijed, kao što je eksperimentalno prikazano u izvornom članku [1], te opsežnije u D. Castafion [2]. Čini se da na njega manje utječe fenomen "rat cijena" koji je motivacija za ϵ-skaliranje. Ratovi cijena se ipak mogu dogoditi i u kombiniranom algoritmu, ali nastaju kod složenijih struktura problema nego kod algoritma unaprijed. Zato, dobra izvedba kombiniranog aukcijskog algoritma manje ovisi o ϵ-skaliranju nego njemu odgovarajući algoritam unaprijed. Jedna od posljedica toga jest da je polaženje od $\epsilon = 1/(n + 1)$ i zaobilaženje ϵ-skaliranja često najbolja opcija. Druga posljedica je da veći faktor ϵ-redukcije može biti korišten bez da ratovi cijena imaju utjecaj u ϵ-skaliranju kombinirane aukcije nego kod ϵ-skaliranja aukcije unaprijed. Kao rezultat toga, obično je potrebno manje faza ϵ-skaliranja kod kombinirane aukcije za učinkovito rješavanje problem ratova cijena.

Poglavlje 3

Aukcijski algoritam za probleme najkraćeg puta

Sada ćemo obratiti pozornost na druge vrste problema mrežnog toka. Naša ideja za izgradnju aukcijskih algoritama za takve probleme je pretvoriti ih u probleme dodjeljivanja, a zatim prikladno primijeniti aukcijski algoritam i pojednostaviti računanja. Započinjemo s klasičnim problemom najkraćeg puta. Pretpostavimo da nam je dan usmjereni graf sa skupom čvorova $\mathcal{N}$, skupom lukova $\mathcal{A}$ i duljina a_{ij} za svaki luk (i, j). U ovom odjeljku, pod *usmjerena šetnja* mislimo na niz čvorova $(i_1, i_2, ..., i_k)$ takvih da je (i_m, i_{m+1}) luk za sve $m = 1, 2, ..., k-1$. Dodatno, ako su čvorovi $i_1, i_2, ..., i_k$ različiti niz $(i_1, i_2, ..., i_k)$ nazivamo *usmjereni put*. Duljina puta definirana je kao zbroj njegovih duljina lukova. Uz pretpostavku da svi usmjereni ciklusi imaju pozitivne duljine, želimo pronaći usmjereni put minimalne duljine među svim usmjerenim putovima iz zadanog polazišta (čvor 1) i u zadano odredište (čvor t).

Uočimo da postoji transformacija koja pretvara ovaj problem u određenu vrstu problema dodjeljivanja kao što je prikazano na Slici 3. Možemo primijeniti aukcijski algoritam za rješavanje ekvivalentnog problema, ali ispada da je struktura tog problema takva da i algoritam naivne aukcije ($\epsilon = 0$) funkcionira. Pod pretpostavkom da su sve duljine lukova nenegativne, možemo pokrenuti naivni algoritam aukcije s nul-vektorom cijene i dodjeljivanjem koje dodjeljuje, za $i \neq 1$ i $i \neq t$, svaku osobu i' objektu i; ovaj par dodjeljivanje-cijena zadovoljava CS (budući da su sve duljine lukova nenegativne i dodijeljeni lukovi (i', i) imaju trošak nula), ali nije dopustivo jer ostavlja osobu $1'$ i objekt t nedodijeljenima.



Slika 3: Na slici se nalaze problem najkraćeg puta (s polazištem 1 i odredištem $t = 4$) i pripadajući problem dodjeljivanja.

To se može pokazati pomoću indukcije ili praćenjem koraka naivnog aukcijskog algoritma tako da svako generirano dodjeljivanje sadrži (moguće prazan) niz oblika

$$(1', i_1), (i_1', i_2), ..., (i_{k-1}', i_k),$$

zajedno s lukovima

$$(i', i) \quad za \quad i \neq i_1, ..., i_k, t;$$

ovaj niz odgovara putu $P = (1, i_1, ..., i_k)$. Dok god je $i_k \neq t$, (jedinstvena) nedodijeljena osoba u naivnom aukcijskom algoritmu je osoba i_k', koja odgovara terminalnom čvoru puta. Ako je $i_k = t$, naivni aukcijski algoritam prestaje. U suprotnom, nedodijeljena osoba i_k' daje ponudu i imamo dvije mogućnosti:

(a) Najbolji objekt je i_k; u tom slučaju i_{k-1} postaje nedodijeljeni, a put $(1, i_1, ..., i_{k-1})$ koji odgovara novom dodjeljivanju se dobiva iz prethodnog puta $(i, i_1, i_2, ..., i_k)$ pomoću "kontrakcije" jednim čvorom.

(b) Najbolji objekt je $i_{k+1} \neq i_k$; u tom slučaju put $(1, i_1, ..., i_k, i_{k+1})$ koji odgovara novom dodjeljivanju se dobiva iz prethodnog puta $(1, i_1, i_2, ..., i_k)$ pomoću "proširenja" jednim čvorom.

Sad ćemo opisati naivni aukcijsku algoritam u terminima problema najkraćeg puta.

3.1 Aukcijski algoritam za traženje najkraćeg puta

Aukcijski algoritam za najkraći put održava u svakom trenutku usmjereni put $(1, i_1, i_2, ..., i_k)$. Ako je i_{k+1} čvor koji ne pripada putu $P = (1, i_1, i_2, ..., i_k)$ i (i_k, i_{k+1}) je luk, tada produžiti P za i_{k+1} zapravo znači zamijeniti put P s putom $(1, i_1, i_2, ..., i_k, i_{k+1})$, i to nazivamo *proširenje* od P za i_{k+1}. Ako se P ne sastoji samo od izvornog čvora 1, *kontrakcija* puta P znači zamijeniti P s putom $(1, i_1, i_2, ..., i_{k-1})$.

Algoritam održava i vektor cijene p tako da zajedno s P zadovoljava sljedeće svojstvo

$$p_i \leq a_{ij} + p_j, \qquad \forall (i, j) \in \mathcal{A}, \tag{22a}$$

$$p_i = a_{ij} + p_j, \quad \text{za sve parove uzastopnih čvorova i i j od P,} \tag{22b}$$

kojeg zovemo uvjetom komplementarnosti (CS). Ovaj uvjet može biti povezan s CS-uvjetom za ekvivalentni problem dodjeljivanja kao i s CS-uvjetom za pridruženi problem toka minimalnih troškova.

Može se pokazati da ako par (P, p) zadovoljava CS-uvjet tada je dio od P između čvora 1 i bilo kojeg čvora $i \in P$ najkraći put od 1 do i, dok je $p_1 - p_i$ odgovarajuća najkraća udaljenost. Kako bi to vidjeli, uočimo da je prema jednakosti (22b), $p_i - p_k$ udaljenost dijela P između i i k, te svaki put između i i k mora imati duljinu najmanje jednaku $p_i - p_k$. Pretpostavimo da početni par (P, p) zadovoljava CS-uvjet. Ovo nije restriktivna pretpostavka kada su sve duljine lukova nenegativne, jer se tada može koristiti zadani par

$$P = (1), \qquad p_i = 0, \quad \forall i. \tag{23}$$

Algoritam nastavlja s iteracijama, pretvarajući par (P, p) koji zadovoljava CS-uvjet u drugi par koji također zadovoljava CS-uvjet. Pri svakoj iteraciji, put P je produžen novim čvorom ili je skraćen brisanjem njegovog terminalnog čvora. U drugom slučaju, cijena terminalnog čvora se strogo povećava.

Do degenerativnog slučaja dolazi kada se put sastoji samo od izvornog čvora 1; u tom slučaju put je ili produžen ili ostaje nepromijenjen s cijenom p koja se strogo povećava. Iteracija je sljedeća.

Tipična iteracija aukcijskog algoritma /Algoritam za traženje najkraćeg puta

Neka je i terminalni čvor od P. Ako vrijedi

$$p_i < \min_{(i,j)\in\mathcal{A}} \{a_{ij} + p_j\},$$

idite na Korak 1; inače idite na Korak 2.

Korak 1 (Skraćenje puta): Postaviti

$$p_i := \min_{(i,j)\in\mathcal{A}} \{a_{ij} + p_j\},$$

i ako je $i \neq 1$, skratite P. Idite na sljedeću iteraciju.

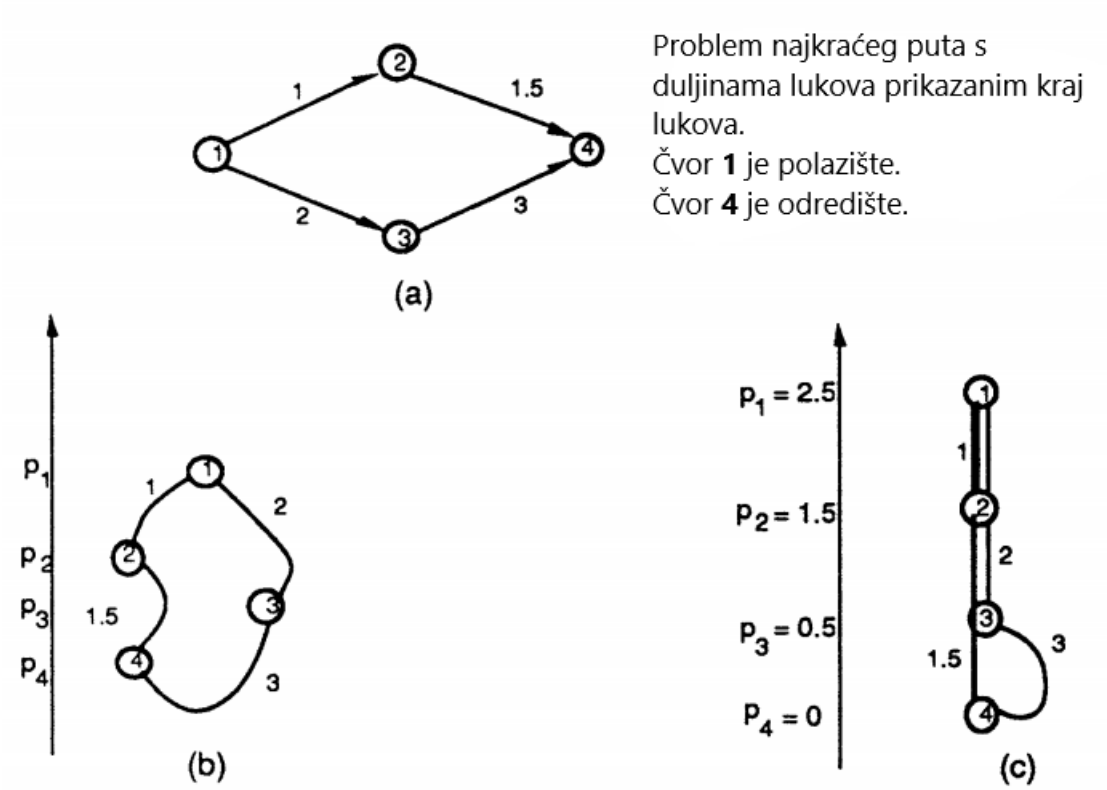
Korak 2 (Produženje puta): Produžite P s čvorom j_i gdje je

$$j_i = \arg\min_{(i,j)\in\mathcal{A}} \{a_{ij} + p_j\},$$

Ako je j_i odredište t, stanite; P je željeni najkraći put. U suprotnom idite na sljedeću iteraciju.

Uočite da je nakon produženja (Korak 2) P najkraći usmjereni put od 1 do j_i; da to nije tako, onda bi dodavanje j_i u P stvorilo ciklus, a za svaki luk (i, j) ovog ciklusa imali bi $p_i = a_{ij} + p_j$. Dodavanjem ovog uvjeta na ciklus vidimo da ciklus mora imati duljinu nula, što nije moguće s našim pretpostavkama.

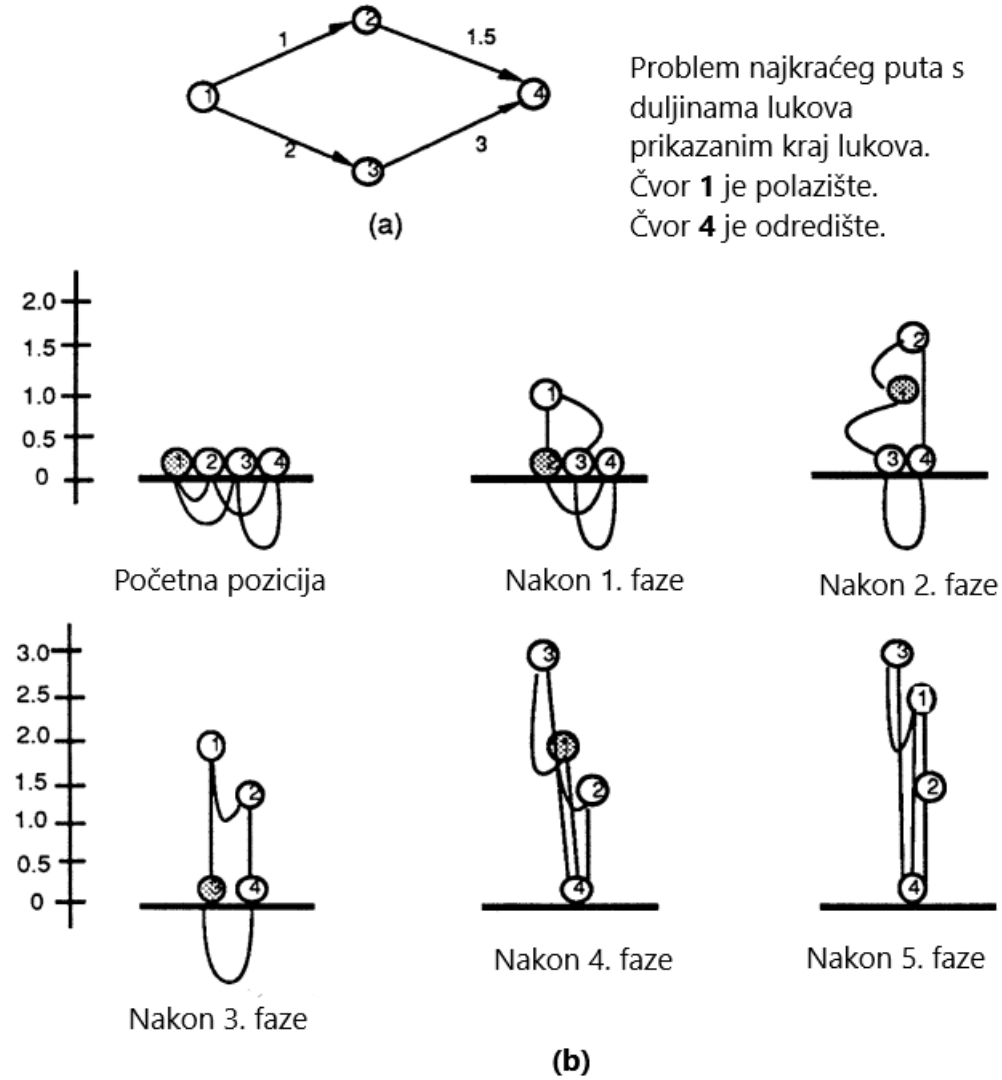
Postoji zanimljiva interpretacija CS-uvjeta u smislu mehaničkog modela. Zamislimo da je svaki čvor lopta, te za svaki luk $(i, j) \in \mathcal{A}$ povežemo i i j žicom duljine a_{ij}. (To zahtijeva da $a_{ij} = a_{ji} > 0$, što pretpostavljamo radi interpretacije.) Neka su dobivene lopte i žice u proizvoljnom položaju u trodimenzionalnom prostoru, te neka je p_i vertikalna koordinata čvora i. Tada CS-uvjet $p_i - p_j \leq a_{ij}$ očito vrijedi za sve lukove (i, j), kao što je prikazano na Slici(4b). Ako objesimo model za izvorni čvor kao na Slici(4c), tada za sve napete žice (i, j) imamo $p_i - p_j = a_{ij}$, stoga napeti lanac tih žica odgovara najkraćem putu između krajnjih čvorova. Konkretno, duljina napetog lanca koji povezuje izvorni čvor 1 s bilo kojim drugim čvorom i je $p_1 - p_i$ i jednaka je najkraćoj udaljenost od 1 do i.



Slika 4: Ilustracija CS-uvjeta za problem najkraćeg puta.

Algoritam aukcije za traženje najkraćeg puta također može biti interpretiran kao model lopti povezanih žicama; može se promatrati kao proces kod kojeg se čvorovi podižu u fazama, kao što je prikazano na Slici 5. U početku svi čvorovi leže na ravnoj površini. U svakoj fazi podižemo posljednji čvor u napetom lancu, koji počinje s izvornim čvorom, do razine na kojoj bar još jedna žica postaje napeta.

Kada postoji jedan izvorni čvor i više odredišta, algoritam se može primijeniti gotovo bez izmjena. Jednostavno zaustavljamo algoritam kada su sva odredišta postala terminalni čvor puta P barem jednom. Također uočimo da algoritam možemo primijeniti na problem kod kojeg imamo više izvornih čvorova i jedno odredište tako da prvo zamijenimo uloge izvora i odredišta te smjerove svih lukova.



Slika 5: Ilustracija aukcije/problema najkraćeg puta u terminima modela lopti povezanih žicama za problem prikazan pod (a).

3.2 Obrnuti algoritam

Postoji mnogo načina za ubrzanje osnovnog algoritma. Najznačajniji od njih se odnosi na dvostrane verzije algoritma koji održava, uz put P, drugi put R koji završava na odredištu. Da bismo razumjeli ovu verziju, prvo uočimo da se kod problema najkraćeg puta može izmijeniti uloga izvora i odredišta tako da se okrene orijentacije svih lukova. Stoga je moguće koristiti obrnutu verziju algoritma koji održava put R (koji završava na

odredištu) i mijenja se pri svakoj iteraciji pomoću kontrakcije ili produžetka. Ovaj algoritam, zvan obrnuti algoritam, matematički je ekvivalentan ranijem algoritmu unaprijed i paralelan obrnutom aukcijskom algoritmu za problem dodjeljivanja iz prethodnog odjeljka. U početku, u obrnutom algoritmu, R je bilo koji put koji završava na odredištu, i p je bilo koji vektor cijene koji zadovoljava CS-uvjete (22) zajedno s R; na primjer,

$$R = (t), \qquad p_i = 0, \quad \forall i,$$

ako su duljine lukova nenegativne.

Tipična iteracija obrnutog algoritma

Neka je j početni čvor od R. Ako

$$p_j > \max_{(i,j)\in\mathcal{A}} \{p_i - a_{ij}\},$$

idite na Korak 1, inače idite na Korak 2.

Korak 1:(Skraćenje puta): Postaviti

$$p_j := \max_{(i,j)\in\mathcal{A}} \{p_i - a_{ij}\},$$

i ako je $i \neq t$, skratite R (odnosno obrišite početni čvor). Idite na iduću iteraciju.

Korak 2(Produženje puta): Produžite R za čvor j_x (odnosno postavite j_x za početni čvor od R, dakle prije j), gdje je

$$j_x = \arg\min_{(i,j)\in\mathcal{A}} \{p_i - a_{ij}\},$$

Ako je j_x izvorni čvor 1, stanite; R je željeni najkraći put. U suprotnom idite na sljedeću iteraciju.

Obrnuti algoritam je najkorisniji kada je u kombinaciji s algoritmom unaprijed. U kombiniranom algoritmu počinjemo s vektorom cijene p i dva puta P i R koji zajedno s p zadovoljavaju CS-uvjet, pri čemu P započinje u izvoru dok R započinje u odredištu. Putovi P i R se proširuju ili skraćuju prema pravilima algoritma unaprijed i obrnutog algoritma, a kombinirani algoritam prestaje kada P i R imaju zajednički čvor. Kako i P i R zajedno s p zadovoljavaju CS-uvjet, kada se P i R sretnu, na primjer u čvoru i, kompozicija putova koja se sastoji od dijela puta P (od 1 do i) i dijela puta R (od i do t) će biti najkraći put.

Kombinirani Unaprijed/Obrnuti aukcijski/Najkraći put algoritam

Korak 1: (Pokrenite algoritam unaprijed) Izvršite nekoliko iteracija algoritma

unaprijed (ovisno o uvjetu prekida), od kojih barem jedna dovodi do povećanja izvorne cijene p_i. Idite na Korak 2.

Korak 2: (Pokrenite obrnuti algoritam) Izvršite nekoliko iteracija obrnutog algoritma (ovisno o uvjetu prekida), od kojih barem jedna dovodi do spuštanja odredišne cijene p_t. Idite na Korak 1.

Kombinirani algoritam se također može interpretirati kao model lopti povezanih žicama sa Slike 4. Ponovno, na početku svi čvorovi leže na ravnoj površini. Tada koristeći dio algoritma s algoritmom unaprijed, podižemo čvorove, u fazama, kao na Slici 5. Kada koristimo dio algoritma s obrnutim algoritmom, spuštamo čvorove, u fazama; u svakoj fazi, spuštamo najviši čvor u napetom lancu koji završava na odredištu na razinu na kojoj bar još jedna žica postaje napeta. Uočimo da se slučaj višestrukih odredišta može riješiti pomoću zasebnog obrnutog puta za svako od tih odredišta. Tada alterniramo između koraka unaprijed i obrnutog koraka kao u prethodnom algoritmu, uzimajući ciklički različita odredišta u različitim koracima obrnutog algoritma.

Poglavlje 4

Proširenje na transportni problem

Razmotrimo sada proširenje aukcijskog algoritma na nekapacitirani transportni problem. Ovdje imamo bipartitni graf s m izvora i n ponora. Problem je veoma sličan problemu dodjeljivanja i ima sljedeći oblik

$$\sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij} \to \max$$

$$\sum_{j\in\mathcal{A}(i)} x_{ij} = \alpha_i, \quad \forall i = 1, ..., m, \tag{23}$$

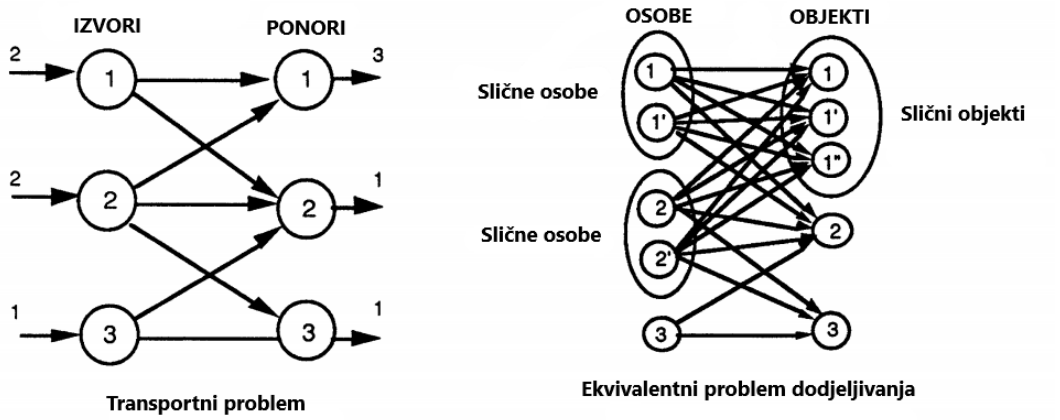
$$\sum_{i\in\mathcal{B}(j)} x_{ij} = \beta_i, \quad \forall j = 1, ..., n,$$

$$0 \leq x_{ij}, \quad \forall (i,j) \in \mathcal{A}.$$

Ovdje su α_i i β_i pozitivni cijeli brojevi koji zbog dopustivosti moraju zadovoljavati

$$\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j.$$

Ovaj problem jednostavno možemo pretvoriti u problem dodjeljivanja tako da svaki čvor izvor (ili ponor) zamijenimo kolekcijom više istih čvorova osoba (ili objekata). Konkretno, čvor izvor i s opskrbom α_i zamjenjujemo s α_i osoba, a čvor ponor j s potražnjom β_j zamjenjujemo s β_j objekata. Nadalje, za svaki luk (i, j) moramo stvoriti luk koristi a_{ij} koji povezuje svaku osobu koja odgovara i sa svakim objektom koji odgovara j. Primjer je dan Slikom 6.

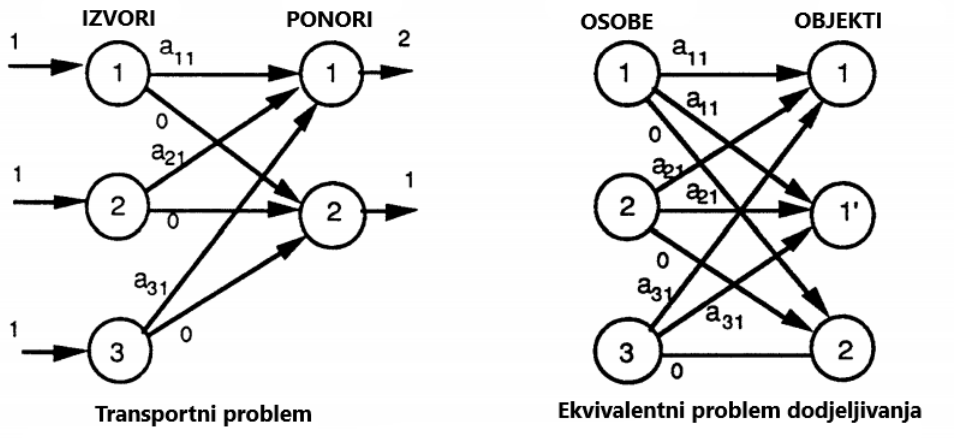


Slika 6

Ekvivalentni problem dodjeljivanja je moguće riješiti jednostavnom primjenom aukcijskog algoritma, no postoje dvije poteškoće:

(a) Dimenzija problema je znatno povećana (broj osoba i broj objekata postaju $\sum_{i=1}^{m} \alpha_i$).

(b) Struktura ekvivalentnog problema dodjeljivanja je takva da su dugotrajni ratovi cijena neizbježni zbog duplikata objekata i osoba stvorenih transformacijom. Primjer je dan Slikom 7.



Slika 7

Navedene poteškoće je moguće riješiti prilagodbom aukcijskog algoritma. Posebno, obzirom na problem dodjeljivanja, kažemo da su dva objekta j i j' *slična*, i pišemo $j \sim j'$, ako mogu biti povezani s istim osobama s jednakim vrijednostima, odnosno

$$B(j) = B(j') \quad i \quad a_{ij} = a_{ij'} \quad \forall i \in B(j)$$

Slično, kažemo da su dvije osobe i i i' *slične*, i pišemo $i \sim i'$, ako

$$A(i) = A(i') \quad i \quad a_{ij} = a_{i'j} \quad \forall j \in A(i)$$

Za svaki objekt j skup svih njemu sličnih objekata se naziva *klasa sličnosti* od j i označava se s $M(j)$. Za svaku osobu i skup svih njoj sličnih osoba se naziva *klasa sličnosti* od i i označava se s $M(i)$. Da bismo shvatili kako možemo baratati s klasama sličnosti, razmislimo o aukcijskom algoritmu iz Poglavlja 2 primijenjenom na problem dodjeljivanja sa Slike 7. Ovdje će osoba i ($i = 1, 2, 3$) nastaviti licitirati za slične objekte 1 i $1'$ dok obje cijene p_1 i p'_1 ne budu barem jednake a_{i1} i objekt 2 ne postane manje atraktivan osobi i nego objekti 1 i $1'$. Stoga se a_{i1} može promatrati kao "prag nadmetanja" jer ga sve cijene sličnih objekata moraju dosegnuti prije nego što osoba postane zainteresirana za neki objekt izvan njihove klase sličnosti. Naravno, "pragovi nadmetanja" objekata u nekoj klasi sličnosti mogu biti znatno veći od cijene objekta (koja je, prema pravilima aukcijskog algoritma, ograničena s ϵ obzirom na ostale cijene objekata iste klase odnosno $|p_j - p_k| \leq \epsilon$ za sve $j, k \in M(j)$). Pretpostavimo sada da kada osoba licitira za neki objekt iz klase sličnosti ona osim što ažurira cijenu, ažurira i odgovarajući "prag nadmetanja". Nadalje, pretpostavimo da se "prag nadmetanja" svih objekata iz klase sličnosti diže iznad (uglavnom zajedničke) cijene tih objekata. Tada se može, bez narušavanja ϵ-CS uvjeta, istodobno podići cijene svih objekata klase na minimalni "prag nadmetanja", i time se rješava odgovarajući rat cijena.

4.1 Aukcijski algoritam sa sličnim objektima

Prethodna ideja se može provesti vrlo jednostavno algoritmom koji nazivamo *aukcijskim algoritmom sa sličnim objektima*. Za svaki objekt j održavamo "prag nadmetanja" $\hat{p}_j$. Ako je j nedodijeljen, "prag nadmetanja" $\hat{p}_j$ je jednak inicijalnoj cijeni p_j; svaki put kada je j dodijeljen novoj osobi i, "prag nadmetanja" $\hat{p}_j$ je postavljen na zbroj ϵ i minimalne vrijednosti koju p_j mora postići prije nego i pronađe objekt iz neke druge klase sličnosti koji je jednako atraktivan kao j. Cijene su određene "pragom nadmetanja". Konkretno, cijena objekta j je minimalni "prag nadmetanja" među objektima iste klase sličnosti od j, $M(j)$:

$$p_j = \min_{k \in M(j)} \hat{p}_k \tag{24}$$

Dakle, svi objekti unutar klase sličnosti imaju istu cijenu, ali mogu imati različite "pragove nadmetanja"

Formalnije, aukcijski algoritam sa sličnim objektima isti je kao aukcijski algoritam Poglavlja 3 osim jedne razlike: u fazi ponude, nedodijeljena osoba i pronalazi njoj najbolji objekt j_i i odgovarajuću najbolju vrijednost na temelju "praga nadmetanja",

$$j_i = \arg\max_{j \in A(i)} \{a_{ij} - \hat{p}_j\}, \quad v_i = a_{ij_i} - \hat{p}_{j_i}. \tag{25}$$

Tada, i postavlja "prag nadmetanja" $\hat{p}_{j_i}$ na zbroj p_{j_i} i povećanja ponude $v_i - \omega_i + \epsilon$, što se temelji na drugom najboljem objektu iz različite klase sličnosti odnosno

$$\omega_i = \max_{j \in A(i), j \neq M(j_i)} \{a_{ij} - \hat{p}_j\} \tag{26}$$

(umjesto $\omega_i = \max_{j \in A(i), j \neq j_i} \{a_{ij} - p_j\}$). Kada "prag nadmetanja" svih objekata iste klase sličnosti poraste tada i cijena klase također raste do minimalnog "praga nadmetanja" prema formuli (24). Uočimo da zbog relacije $p_j = \min_{k \in M(j)} \hat{p}_k$, $\hat{p}_j$ u jednakostima (25) i (26) može biti zamijenjen s p_j. Stoga, nedodijeljena osoba licitira za "najbolji" objekt iz "najbolje" klase sličnosti, ali povećanje ponude se sada temelji na "pragu nadmetanja" te može biti znatno veće nego povećanje ponude koje se temeljilo na cijenama.

Što se tiče valjanosti algoritma, može se pokazati da cijene objekata i dalje zadovoljavaju ϵ-CS i, u suštini, ponavljanjem dokaza Propozicije 2.1.2, slijedi da će se algoritam zaustaviti u konačno mnogo koraka. Međutim, za cijeli broj a_{ij} se može pokazati da će konačno dodjeljivanje biti optimalno ako je $\epsilon < \frac{1}{m}$, gdje je m broj objekata iz klase sličnosti, prije nego za $\epsilon < \frac{1}{n}$, što je slučaj kod aukcijskog algoritma Poglavlja 2. Prema tome povećana dimenzionalnost zbog dupliciranih objekata stvorenih transformacijom transportnog problema (23) u problem dodjeljivanja se ne odražava na odgovarajuće smanjenje vrijednosti ϵ koja vodi do optimalnog rješenja.

4.2 Aukcijski algoritam sa sličnim osobama

Može se dodatno poboljšati izvedba aukcijskog algoritma uzimajući u obzir prisutnost sličnih osoba. Ideja je podnijeti *zajedničku ponudu* za sve osobe unutar klase sličnosti ako barem jedna osoba u klasi nije dodijeljena. Kao rezultat toga, osobe u istoj klasi se ne "natječu" jedne protiv drugih za iste objekte, a podnesene ponude su veće nego što bi inače bile. Zapravo, osobe iste klase sličnosti surađuju kako bi "komprimirali" rat cijena i riješili ga unutar jedne iteracije.

Ideja zajedničke ponude za klasu sličnosti osoba se može kombinirati s ranije spomenutom idejom o "pragu nadmetanja" te s odgovarajućim aukcijskim algoritmom za slične

objekte. Konkretno, kooperativna ponuda klase sličnosti osoba temelji se na "pragu nadmetanja" objekata. S pravilnim racionaliziranjem, dobiva se algoritam za transportni problem (23), koji je u suštini aukcijski algoritam za odgovarajući ekvivalentni problem dodjeljivanja (vidite Sliku 6), ali sa sličnim osobama i objektima koji su ispravno uzeti u obzir. U ovom aukcijskom/ transportnom algoritmu, svaki ponor j ima cijenu p_j, a svaki luk (i, j) koji nosi pozitivan tok x_{ij} ima "prag nadmetanja" y_{ij}. Cijena p_j je jednaka izvornoj cijeni ponora j ako potražnja za ponorom još nije iscrpljena ($\sum_{i\in B(j)} x_{ij} < \beta_j$), a u suprotnom je jednaka minimalnom "pragu nadmetanja" ulaznih lukova koji imaju pozitivni tok ($\min_{i\in B(j), x_{ij}>0} y_{ij}$). Izvori licitiraju za ponore povećavanjem "pragova nadmetanja" odgovarajućih lukova, a cijena ponora se povećava jednom nakon što se iscrpi sva potražnja i poveća se pripadni "prag nadmetanja". Kako se cijene ponora povećavaju, odgovarajuća dobit izvora definirana s

$$\pi_i = \max_{j\in A(i)}\{a_{ij} - p_j\} \tag{27a}$$

se smanjuje, dok se ϵ-CS uvjet

$$\pi_i + p_j \le a_{ij} + \epsilon, \qquad \forall (i, j) \in \mathcal{A}, \quad x_{ij} > 0 \tag{27b}$$

održava. ϵ-CS uvjeti (27a) i (27b) proširuju odgovarajući uvjet (8) za problem dodjeljivanja. Za cjelobrojni problem, vektor dopustivog toka je optimalan ako postoji vektor cijena p i odgovarajući vektor profita π koji zadovoljava ϵ-CS za $\epsilon < \frac{1}{\min\{m,n\}}$.

Poglavlje 5

Generički aukcijski algoritam za probleme toka minimalnog ukupnog troška

5.1 Svođenje problema minimalnog toka na problem dodjeljivanja

Sada promatramo općeniti algoritam aukcijskog tipa za probleme minimizacije troškova toka. On uključuje specijalne slučajeve aukcijskog algoritma za probleme dodjeljivanja i transportne probleme kao i za ϵ-relaksacijsku metodu o kojoj ćemo kasnije govoriti. Usavršavanjem ovog općenitog algoritma za probleme mrežnog toka sa specijalnom strukturom, može se dobiti učinkovite metode; na primjer, općeniti algoritam je osnova za algoritam najkraćih k putova bez zajedničkih vrhova(osim prvog i zadnjeg).

Imamo usmjereni graf sa skupom čvorova $\mathcal{N}$ i skupom lukova $\mathcal{A}$. Broj čvorova je označen s N, a broj lukova s A. Za svaki luk (i, j) postoji optimizacijska varijabla x_{ij} koju nazivamo *tok na luku (i,j)*. S x označavamo vektor toka $\{x_{ij}|(i, j) \in \mathcal{A}\}$. Naš problem minimalnog troška toka tada glasi

$$\sum_{(i,j)\in \to \min \mathcal{A}} a_{ij}x_{ij} \qquad (MCF)$$

pod uvjetom da

$$\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji} = s_i, \quad \forall i \in \mathcal{N}, \tag{28}$$

$$b_{ij} \leq x_{ij} \leq c_{ij}, \quad \forall (i, j) \in \mathcal{A}, \tag{29}$$

pri čemu su a_{ij}, b_{ij}, c_{ij} i s_i dani cijeli brojevi. Problem je dopustiv ako postoji vektor toka x koji zadovoljava uvjete (28) i (29); inače je nedopustiv.

Problem se može pretvoriti u ekvivalentan transportni problem što se može vidjeti na Slici 9. Tu ekvivalenciju je moguće iskoristiti za preobrazbu aukcijskih/transportnih ideja prethodnog poglavlja za kontekst problema minimalnog toka. Izvedimo najprije prigodnu formu ϵ-CS uvjeta.

Obzirom na ekvivalentni transportni problem na Slici 9, označimo s λ_{ij} cijenu ponora koji odgovara luku (i, j), te označimo s p_i dobit izvora koji odgovara čvoru i. Iz jednakosti (27a) (uz zamjenu a_{ij} s -a_{ij} kako bismo uzeli u obzir promjenu iz maksimizacije u minimizaciju) imamo

$$p_i + \lambda_{ij} \geq 0, \quad p_j + \lambda_{ij} \geq -a_{ij}, \quad \forall (i, j) \in \mathcal{A}. \tag{30}$$

Pretpostavimo da je tijekom aukcije ukupni ulazni tok u ponor koji odgovara luku (i, j) uzduž dvaju transportnih lukova $(j, (i, j))$ i $(i, (i, j))$ jednak $c_{ij} - b_{ij}$; za bilo koji početni skup p_i moguće je nametnuti ovaj uvjet uz očuvanje ϵ-CS uvjeta odabirom početnog λ_{ij} kao $\lambda_{ij} = \min\{-p_i, -p_j - a_{ij}\}$. Zatim, iz jednakosti (27b) također imamo

$$p_i + \lambda_{ij} \leq \epsilon \quad \forall (i, j) \in \mathcal{A}, \quad x_{ij} < c_{ij}, \tag{31a}$$

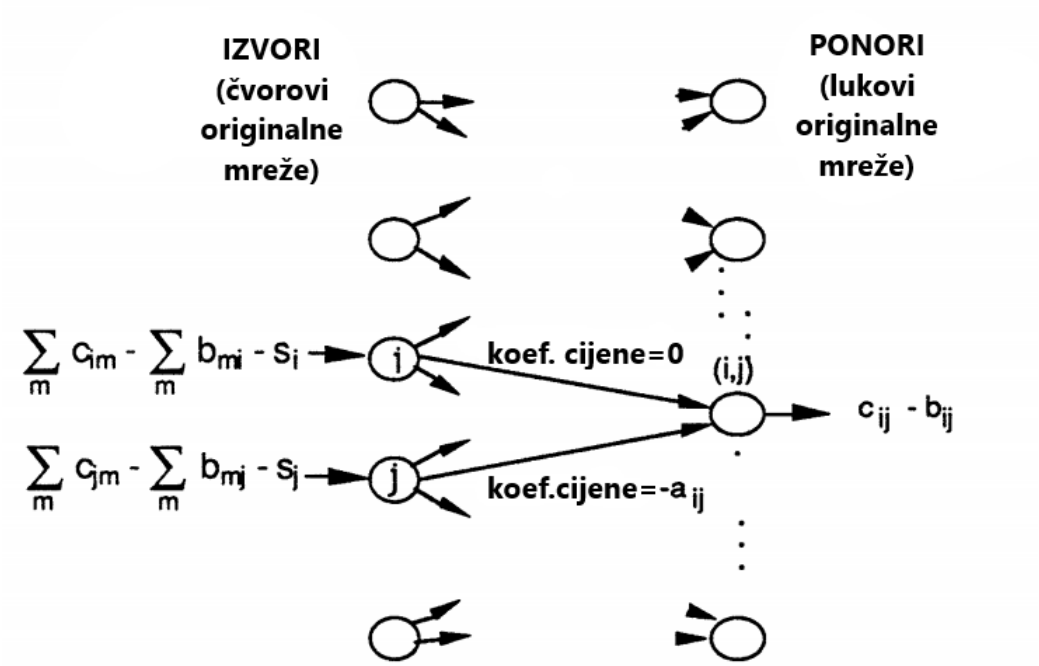
$$p_j + \lambda_{ij} \leq -a_{ij} + \epsilon \quad \forall (i, j) \in \mathcal{A}, \quad b_{ij} < x_{ij}. \tag{31b}$$

Iz formula (30) i (31a) dobivamo $p_j + a_{ij} \geq -\lambda_{ij} \geq p_i - \epsilon$ ako je $x_{ij} < c_{ij}$, stoga

$$p_i - p_j \leq a_{ij} + \epsilon \quad \forall (i, j) \in \mathcal{A}, \quad x_{ij} < c_{ij}. \tag{32a}$$

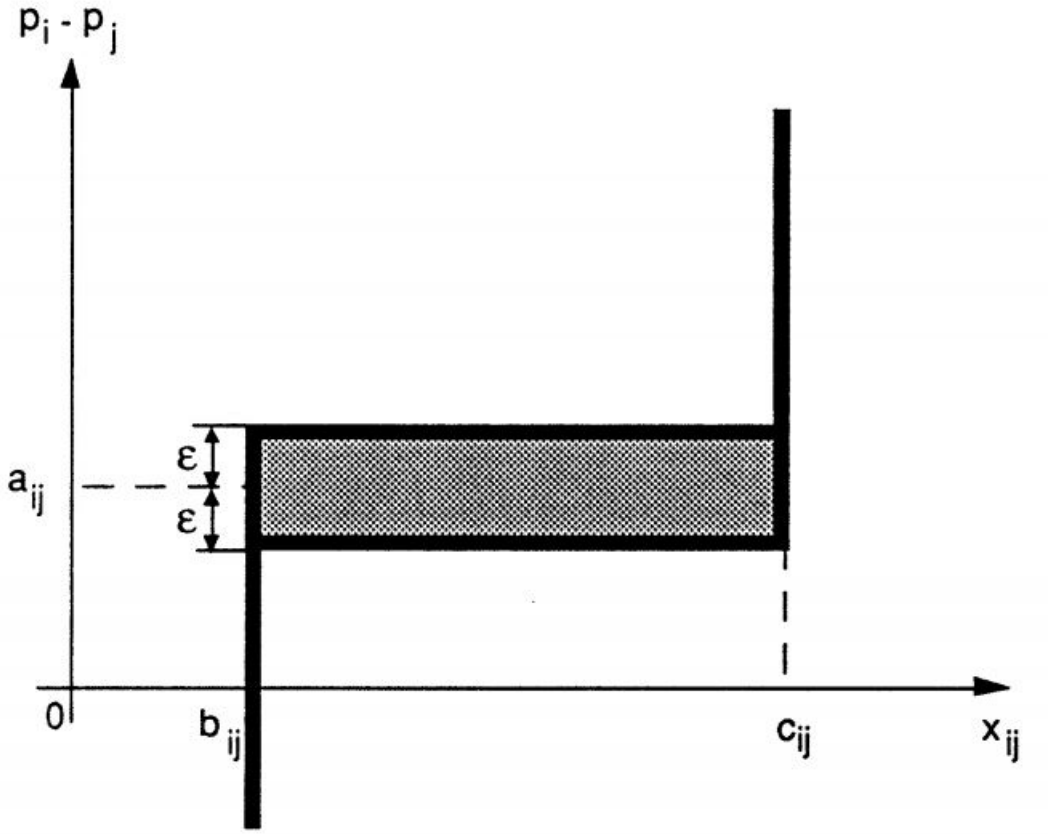
Također iz formule (30) dobivamo $p_i \geq -\lambda_{ij} \geq a_{ij} + p_j - \epsilon$ ako je $b_{ij} < x_{ij}$, stoga

$$p_i - p_j \leq a_{ij} - \epsilon \quad \forall (i, j) \in \mathcal{A}, \quad b_{ij} < x_{ij}. \tag{32b}$$



Slika 8: Transformacija problema minimalnog toka u transportni problem (23).

Kažemo da par tok-cijena (x, p) zadovoljava uvjet ϵ-komplementarnosti (kraće ϵ-CS) ako x zadovoljava (29) te vrijede (32a) i (32b); vidite Sliku 10. Sljedeća propozicija se podudara s Propozicijom 2.1.1 za problem dodjeljivanja.



Slika 9: Ilustracija ϵ-CS. Svi parovi, tokovi lukova x_{ij} i razlika cijena $p_i - p_j$, bi trebali ležati na podebljanoj liniji ili na osjenčanoj površini između podebljanih linija.

Propozicija 5.1.1. *Ako je x dopustiv i za neki vektor cijene p par* (x, p) *zadovoljava* ϵ*-CS za* $\epsilon < \frac{1}{N}$ *(N je broj čvorova), tada je x optimalan za problem toka minimalnog troška (MCF).*

Pri razvoju aukcijskih algoritama za problem minimalnog toka motiviraju nas aukcijski algoritmi za ekvivalentne probleme - transportni problem i problem dodjeljivanja sa sličnim objektima i osobama. Pojam ponude jednog izvora i pripadajući rast toka uz najbolji izlazni luk se može protumačiti, u kontekstu problema minimalnog toka, kao povećanje toka jednog čvora uz "najbolje" izlazne lukove ili smanjenje toka uz "najbolje" ulazne lukove. Zbog tih izmjena toka također bi moglo uslijediti i povećanje cijene čvora, uz održavanje ϵ-CS uvjeta. Budući da smo u kontekstu algoritma za transport koristili zajedničku ponudu od strane nekoliko čvorova, ovdje ima smisla uzeti u obzir istodobno povećanje cijena za nekoliko čvorova.

Sada uvodimo neke računalne operacije koje će predstavljati temelje generičkog aukcijskog algoritma za problem toka minimalnog troška. Svaka od sljedećih definicija pretpostavlja da je par (x, p) tok-cijena vektor koji zadovoljava ϵ-CS uvjet.

Definicija 5.1.2. *Kažemo da je luk (i, j) ϵ^+-neblokiran ako*

$$p_i = p_j + a_{ij} + \epsilon, \quad x_{ij} < c_{ij}.$$

Luk (j, i) je ϵ^--neblokiran ako

$$p_i = p_j - a_{ji} + \epsilon, \quad b_{ji} < x_{ji}.$$

Označimo s W_i skup lukova (i, j) koji su ϵ^+-neblokirani i lukova (j, i) koji su ϵ^--neblokirani.

U sljedećim algoritmima dopušteno je povećanje toka samo uz ϵ^+-neblokirane lukove i dopušteno je smanjenje samo uz ϵ^--neblokirane lukove.

Definicija 5.1.3. *Za luk (i, j) (ili luk (j, i)) iz skupa W_i neka je δ skalar takav da je $0 < \delta \leq c_{ij} - x_{ij}$(odnosno $0 < \delta \leq x_{ji} - b_{ji}$). Za taj δ uvodimo novi tok povećanjem postojećeg tok za δ na lukovima $(i, j) \in W_i$ i smanjenjem za δ na lukovima $(j, i) \in W_i$, ostavljajući sve ostale tokove, kao i vektor cijene, nepromijenjenima.*

Iz definicije je očito da novi tok čuva ϵ-CS. Sljedeća operacija se sastoji od podizanja cijene podskupa čvorova za maksimalni zajednički prirast γ za koji se neće prekršiti ϵ-CS.

Definicija 5.1.4. *Porast cijena nepraznog, pravog podskupa čvorova od I (odnosno $I \neq \emptyset, I \neq \mathcal{N}$) se sastoji od ostavljanja vektora toka x i cijena čvorova koji ne pripadaju I nepromijenjenima, te povećavanjem cijena čvorova koji pripadaju I za iznos γ koji je dan s*

$$\gamma = \begin{cases} \min\{S^+ \cup S^-\} & \text{ako je } S^+ \cup S^- \neq \emptyset \\ 0, & \text{ako je } S^+ \cup S^- = \emptyset \end{cases} \tag{33}$$

gdje su S^+ i S^- skupovi skalara dani s

$$S^+ = \{p_j + a_{ij} + \epsilon - p_i | (i, j) \in \mathcal{A} \text{ t.d. } i \in I, j \notin I, x_{ij} < c_{ij}\}, \tag{34}$$

$$S^- = \{p_j - a_{ji} + \epsilon - p_i | (j, i) \in \mathcal{A} \text{ t.d. } i \in I, j \notin I, x_{ji} > b_{ji}\} \tag{35}$$

Koristeći definicije se može potvrditi da porast cijena održava ϵ-CS. Uočite da je svaki skalar u skupovima S^+ i S^- nenegativan prema ϵ-CS uvjetima (32a) i (32b) stoga je i skalar γ nenegativan. Ako je $\gamma > 0$ kažemo da je porast cijene *značajan*, a ako je $\gamma = 0$ kažemo da je porast cijene *trivijalan*. (Trivijalan porast cijena ne mijenja niti vektor toka niti vektor cijene, uvodi se zbog lakšeg shvaćanja.)

U slučaju kada se podskup I sastoji od samo jednog čvora i, porast cijene jednočlanog skupa i se također odnosi na povećanje cijene čvora i. Rast cijena jednog čvora i je značajan ako i samo ako skup $S^+ \cup S^-$ nije prazan, ali je pripadni skup W_i vrha i prazan. Za dopustiv problem se može pokazati da ako je pripadni skup W_i čvora i prazan, tada skup $S^+ \cup S^-$ mora biti neprazan.

5.2 Generički algoritam

Ukratko opisano, generički algoritam se sastoji od slijeda povećanja toka za δ i operacija podizanja cijene. Redoslijed izvršavanja ovih operacija podložan je veoma blagim restrikcijma, čime se omogućava fleksibilnost kako bismo mogli što bolje iskoristiti samu strukturu problema.

Pretpostavimo da je problem toka minimalnog troška dopustiv, te da par (x, p) zadovoljava ϵ-CS. Za dani vektor toka x definiramo *višak* g_i čvora i kao razliku ukupnog toka koji ulazi u i i ukupnog toka koji izlazi iz i, odnosno

$$g_i = \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji} - \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij} + s_i.$$

Pretpostavimo da čvor i ima $g_i > 0$. Tada imamo dvije mogućnosti:

(a) Pripadni skup W_i čvora i je neprazan, i u tom slučaju postoji δ iz Definicije 5.1.3.

(b) Pripadni skup W_i čvora i je prazan, tada će, kao što smo ranije spomenuli, porast cijene čvora i biti značajan.

Dakle, ako je $g_i > 0$ za neki i i problem je dopustiv slijedi da ili postoji δ iz Definicije 5.1.3 ili je moguć značajan porast cijene na čvoru i. Nadalje, budući da će nakon porasta cijena na čvoru i pripadni skup W_i čvora i biti neprazna (vidite formule (33)-(35)), za dopustiv problem spomenuti δ uvijek postoji za čvor i s $g_i > 0$.

Prethodna opažanja čine osnovu metode nazvane generički algoritam. Generički algoritam koristi fiksnu pozitivnu vrijednost ϵ i počinje s parom (x, p) koji zadovoljava ϵ-CS, a prekida se kada je $g_i \leq 0$ za sve čvorove i; u suprotnom nastavlja s iteracijama. Svaka se iteracija sastoji od niza povećanja toka za δ i podizanja cijene, uključujući barem jedno povećanje toka za δ kao što je opisano u nastavku.

Tipična iteracija generičkog algoritma

Izvršite konačan niz, u bilo kojem poretku, povećanja toka za δ i porasta cijene; mora postojati barem jedno povećanje toka za δ, ali to nije nužno i za porast cijene. Nadalje:

(1) Svako povećanje toka za δ treba izvesti na nekom čvoru i s $g_i > 0$, a prirast toka δ mora zadovoljiti $\delta < g_i$.

(2) Svaki rast cijena trebao bi se izvršiti na skupu I gdje je $g_i > 0$ za svaki $i \in I$.

Sljedeća propozicija utvrđuje valjanost generičkog algoritma.

Propozicija 5.2.1. *Pretpostavimo da je problem minimalnig toka dopustiv. Ako je povećanje δ uvijek cijeli broj, tada generički algoritam završava s parom (x, p) koji zadovoljava ϵ-CS. Vektor toka x je dopustiv i optimalan ako je $\epsilon < \frac{1}{N}$.*

Ideja dokaza je da se tok ne može beskonačno povećavati za δ bez nekog srednje značajanog porasta cijene. Dakle, ako se algoritam ne prekine, cijene nekih čvorova s pozitivnim viškom se moraju povećati do beskonačnosti, dok cijene nekih drugih čvorova s negativnim viškom ostaju na početnoj razini. No, lako se vidi, da za dopustiv problem to podrazumijeva narušavanje ϵ-CS uvjeta, što dovodi do kotradikcije.

Ako je problem nedopustiv, moguće je da neće doći do završetka algoritma. Jedan od načina rješavanja nedopustivosti je pretvoriti problem u problem koji će sigurno biti dopustiv tako da uvedemo artificijelne lukove s visokom cijenom. Odgovarajuća razina troškova može se odrediti slično kao u slučaju problema dodjeljivanja (vidite (11) i (12)).

5.3 ϵ-relaksacijska metoda

Operacije povećanja cijene generičkog algoritma mogu uključivati više čvorova. Ako zahtijevamo da je samo jedan čvor uključen u svaki porast cijene, dobivamo metodu ϵ-relaksacije.

Koristimo fiksnu pozitivnu vrijednost ϵ, te počinjemo s parom (x, p) koji zadovoljava ϵ-CS. Nadalje, početni tok je cjelobrojan; što će algoritam i očuvati. Na početku tipične iteracije imamo vektor (x, p) koji zadovoljava ϵ-CS, a odabiremo čvor i s $g_i > 0$; ako ne možemo pronaći takav čvor, algoritam prestaje. Tijekom iteracije izvodimo nekoliko povećanja toka i podizanja cijene koji uključuju čvor i.

Tipična iteracija ϵ-relaksacijske metode

Odaberite čvor i s $g_i > 0$. Ako takvog čvora nema, prekinite algoritam, inače idite na Korak 1.

Korak 1: Ako je W_i prazan skup idite na Korak 3; inače odaberite luk $a \in W_i$ i idite na Korak 2.

Korak 2: Neka je j drugi čvor luka a. Neka je

$$\delta = \begin{cases} \min\{g_i, c_{ij} - x_{ij}\} & \text{ako je } a = (i, j), \\ \min\{g_i, x_{ji} - b_{ji}\} & \text{ako je } a = (j, i). \end{cases} \tag{36}$$

Izvršite povećanje za δ na luku a. Ako ste kao rezultat ove operacije dobili $g_i = 0$, idite na Korak 3; inače idite na Korak 1.

Korak 3: Izvršite porast cijene čvora i. Ako je $g_i = 0$ stanite; inače idite na Korak 1.

Lako se vidi da je metoda ϵ-relaksacije poseban slučaj generičkog algoritma na koji se odnosi Propozicija 5.2.1. Za dopustiv problem, algoritam završava s parom (x, p) koji zadovoljavaju ϵ-CS; tok x je optimalan ako je $\epsilon < 1/N$. Slično kao kod problema dodjeljivanja, moguće je koristiti ϵ-skaliranje zajedno s opisanom metodom, a to je često ključno za dobru izvedbu.

Bibliografija

[1] Bertsekas, D. P., Castaiion, D. A., i Tsaknakis, H., Reverse Auction and the Solution of Inequality Constrained Assignment Problems", SIAM Journal on optimisation 3 (1993) 268-297

[2] Castafion, D. A., Reverse Auction Algorithms for Assignment Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (1992)

[3] Bertsekas, D. P., Auction Algorithms for Network Flow Problems: A Tutorial Introduction, Computational optimization and applications 1 (1992) 7-66

[4] Bertsekas, D. P., Network Optimization: Continuous and Discrete Models, Athena Scientific, (1998) 251-337

Sažetak

Aukcijski algoritmi, temeljeni na principu klasične aukcije, djeluju suprotno klasičnim algoritmima mrežne optimizacije jer u svakoj iteraciji ne poboljšavaju nužno ni primarnu ni dualnu funkciju cilja, no na kraju nalaze optimalnu vrijednost. Zasnovani su na metodama nediferencijabilne optimizacije. U početku opisujemo aukcijski algoritam za klasični problem dodjeljivanja, te se bavimo raznim pitanjima kao što su ϵ-skaliranje, problem nedopustivosti, obrnuti algoritam i slično. Razlog tomu je što se problem toka minimalnog troška lako može pretvoriti u problem dodjeljivanja, te su ta pitanja zajednička svim vrstama promatranih problema. Stoga, kasnije u radu, promatramo adaptaciju algoritma za problem maksimalnog toka, transportni problem, problem najkraćih putova i problem toka minimalnog troška. Za problem toka minimalnog troška uvodimo generički algoritam, te kao njegov specijalni slučaj, razvijamo metodu ϵ-relaksacije.

Summary

Auction algorithms, based on real auction, work counter to classical network optimization algorithms because in each iteration they don't necessarily improve the primal or dual cost, although in the end they find an optimal primal solution. They are based on methods of nondifferentiable optimization. Initially, we describe the auction algorithm for classical assignment problem and we deal with various issues such as ϵ-scaling, dealing with infeasibility, reverse auction etc.. The reason is that the minimum cost flow problem can be easily transformed to an assignment problem, so these issues are common to all types of observed problems. Therefore, later in the paper, we observe the algorithm adaptation for the max-flow problem, transportation problem, shortest path problem and minimum cost flow problem. For minimum cost flow problem we introduce generic algorithm and develop ϵ-relaxation method as a special case of the generic algorithm.

Životopis

Rođena sam 19. lipnja 1991. godine u Zagrebu. Osnovno obrazovanje sam stekla u Područnoj školi Vladimir Nazor u Remetama te u Osnovnoj školi Vladimir Nazor na Jordanovcu. Po završetku osnovnoškolskog obrazovanja upisala sam XV. gimnaziju u Zagrebu. Maturirala sam 2010. godine te iste godine upisala preddiplomski sveučilišni studij Matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu, nakon čega upisujem diplomski studij Primijenjene matematike na istom fakultetu.