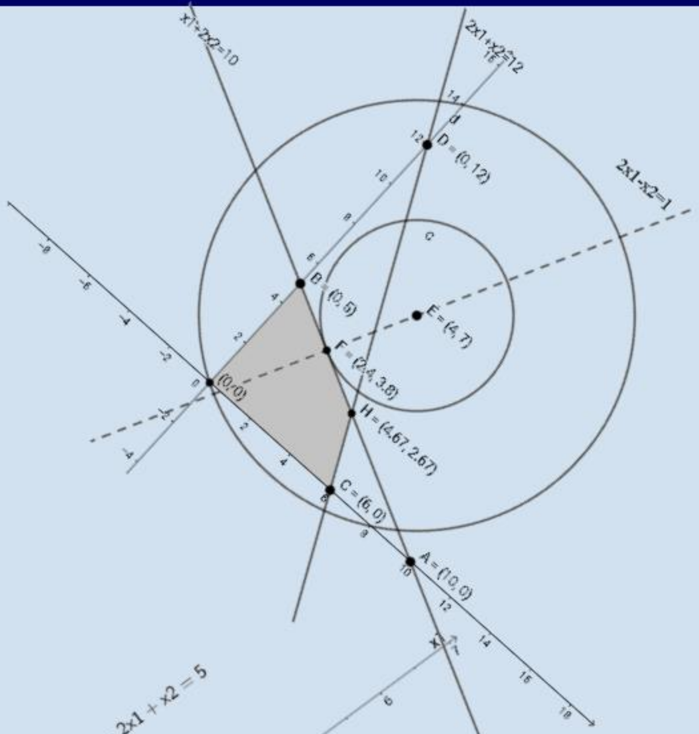
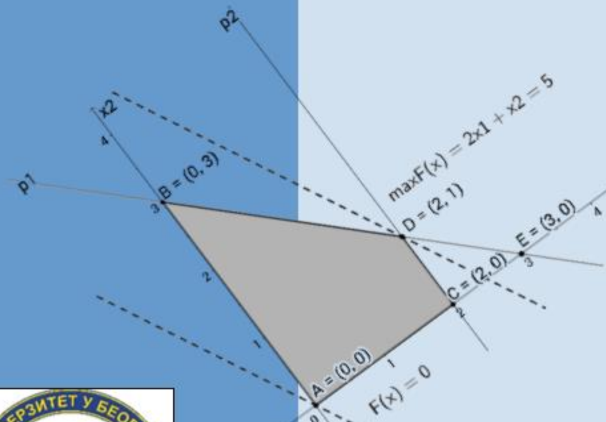
OPERACIONA ISTRAŽIVANJA 1

Dejan Bogdanović
Ivan Jovanović





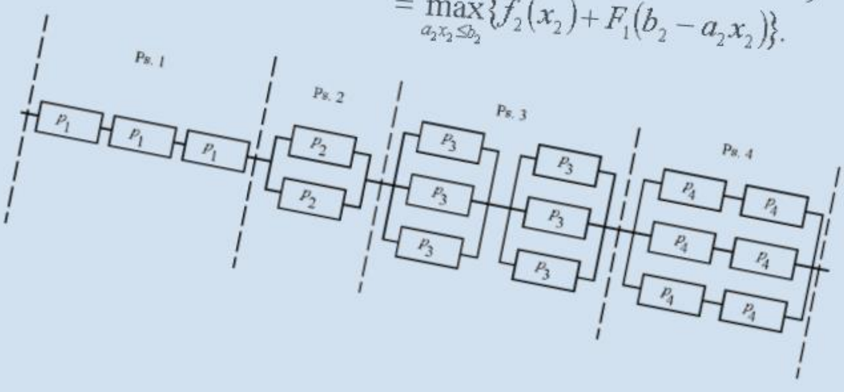
$$F_1(b_1) = \max_{a_1 x_1 \le b_1} \{f_1(x_1)\} = \max_{x_1 \le b_1/a_1} \{f_1(x_1)\},$$

$$F_2(b_2) = \max_{a_1 x_1 + a_2 x_2 \le b_2} \{f_1(x_1) + f_2(x_2)\}$$

$$= \max_{a_2 x_2 \le b_2} \left\{ \max_{a_1 x_1 \le b_2 - a_2 x_2} \left[f_1(x_1) + f_2(x_2) \right] \right\}$$

$$= \max_{a_2 x_2 \le b_2} \left\{ f_2(x_2) + \max_{a_1 x_1 \le b_2 - a_2 x_2} \left[f_1(x_1) \right] \right\}$$

$$= \max_{a_2 x_2 \le b_2} \{ f_2(x_2) + F_1(b_2 - a_2 x_2) \}.$$



UNIVERZITET U BEOGRADU

Tehnički fakultet u Boru

2019. god.

Univerzitet u Beogradu
Tehnički Fakultet u Boru

Dejan Bogdanović
Ivan Jovanović

OPERACIONA ISTRAŽIVANJA 1

Bor, 2019.

Autori: **Dr Dejan Bogdanović,** redovni profesor
Dr Ivan Jovanović, vanredni profesor
Univerzitet u Beogradu, Tehnički fakultet u Boru

OPERACIONA ISTRAŽIVANJA 1

Osnovni udžbenik

Recenzenti:
Prof. dr Miloš Tanasijević,
Univerzitet u Beogradu, Rudarsko – geološki fakultet Beograd
Prof. dr Đorđe Nikolić,
Univerzitet u Beogradu, Tehnički fakultet u Boru

Izdavač: Tehnički fakultet u Boru, Univerzitet u Beogradu

Za izdavača:
Prof. dr Nada Štrbac, dekan Tehničkog fakulteta u Boru

Urednik:
Prof. dr Milan Trumić

Odobren odlukom dekana broj I/6-314/2 od 21.02.2019. god.

Tiraž: 100 primeraka

Štampa: Satcip d.o.o., Kruševac

ISBN 978-86-6305-090-7

CIP - Katalogizacija u publikaciji – Narodna biblioteka Srbije, Beograd

519.8(075.8)

БОГДАНОВИЋ, Дејан, 1965-
Operaciona istraživanja 1 : [osnovni udžbenik] / Dejan Bogdanović, Ivan Jovanović. - Bor : Tehnički fakultet, Univerzitet u Beogradu, 2019 (Kruševac : Satcip). - 268 str. : ilustr. ; 26 cm

Tiraž 100. - Bibliografija: str. 263-268.

ISBN 978-86-6305-090-7

1. Јовановић, Иван, 1965- [аутор]

а) Операциона истраживања
COBISS.SR-ID 274231820

PREDGOVOR

Osnovna karakteristika savremenog, postindustrijskog društva je postojanje velikog broja složenih sistema, kao što su tehnološko – proizvodni sistemi, energetski sistemi, komunikacioni sistemi, transportni sistemi, oružani sistemi, razni sistemi u poljoprivredi, industriji, u oblasti usluga, obrazovanja, i dr.

Postojeći sistemi se karakterišu velikom složenošću koja proizilazi iz njihove strukture, resursa i procesa koji se u njima odvijaju. Menadžeri koji upravljaju njima se suočavaju sa stalnim pritiskom vezanim za donošenje najboljih odluka.

U ovakvim uslovima, upravljanje se mora vršiti metodično, sa programom koji podrazumeva planiranje odluka, odnosno izrada plana i upravljanje realizacijom plana.

Operaciona istraživanja su nastala upravo na tim temeljima i zahtevima sa ciljem da primenom postojećih i razvojem novih naučnih metoda na kvantitativnim osnovama daju odgovore po pitanju najboljeg funkcionisanja složenih sistema u postojećim uslovima. Danas postoji čitav niz raznih naučnih metoda i tehnika koje su razvijene i koje imaju primenu kod rešavanja raznih problema upravljanja sistemima.

Metode i tehnike operacionih istraživanja su prvenstveno namenjene donosiocima odluka (menadžeri), ali i ekspertima i specijalistima koji dobro poznaju prirodu problema koji se rešava i koji, zajedno sa menadžerima, trebaju biti uključeni u ceo proces donošenja optimalnih odluka.

U ovoj knjizi su obrađene određene metode i tehnike operacionih istraživanja koje se danas koristi, kako u svetu, tako i kod nas. Udžbenik *Operaciona istraživanja 1* je svojom namenom i sadržajem prilagođen nastavnom programu istoimenog predmeta na Tehničkom Fakultetu u Boru, Univerziteta u Beogradu. Knjiga razmatra sledeće metode i tehnike: linearno programiranje, nelinearno programiranje, dinamičko programiranje i optimalno rezerviranje sa ciljem da pruži studentima, stručnjacima i menadžerima potrebna znanja iz ove oblasti kako bi što uspešnije upravljali raznim procesima, sistemima i organizacijama u praksi.

Koristimo priliku da se zahvalimo recenzentima na korisnim sugestijama tokom pripreme rukopisa kao i ostalim učesnicima koji su doprineli kvalitetnom oblikovanju teksta. Autori će biti posebno zahvalni svima onima koji svojim sugestijama i predlozima doprinose poboljšanju ponuđenog teksta.

Autori

SADRŽAJ

UVOD

Predmet izučavanja operacionih istraživanja su organizacioni sistemi, koji su definisani svojom strukturom, resursima i procesima koji se u njima odvijaju. Bitna odrednica je namera da se pronađu najbolje odluke u upravljanju operacijama koje se preduzimaju radi ostvarenja utvrđenih ciljeva sistema. Obzirom da se bave istraživanjem operacija u organizacionim sistemima, otuda i potiče ime *operaciona istraživanja*. Opštost operacionih istraživanja se ogleda u tome da se primenjuju na sve tipove organizacionih sistema: poslovne, industrijske, poljoprivredne, vojne, zdravstvene, obrazovne, vladine i slično. Korisnici operacionih istraživanja su donosioci odluka (menadžeri), čiji je zadatak da efikasno i efektivno upravljaju organizacionim sistemima. Operaciona istraživanja su veoma bliska, i može se reći sinonimna, pojmu nauka o menadžmentu, tako da se u anglosaksonskoj literaturi često sreće akronim *OR*/*MS* (*Operations Research* / *Management Science*), a Američko udruženje za operaciona istraživanja se naziva Institut za operaciona istraživanja i nauku o menadžmentu (INFORMS–*Institute for Operations Research and Management Science*).

Operaciona istraživanja obuhvataju veći broj kvantitativnih metoda i tehnika za rešavanje konkretnih problema, koje se još uvek razvijaju. I prema korišćenim metodama i tehnikama, i prema sadržaju problema koje rešavaju, operaciona istraživanja obuhvataju tako široke oblasti da je vrlo teško dati njihovu celovitu i preciznu definiciju. Zbog toga svi pokušaji definisanja operacionih istraživanja polaze od isticanja njihovog predmeta istraživanja, cilja, metoda i tehnika koje koriste, da bi se, na osnovu određenih karakteristika, dala njihova opisna definicija. Odgovor na pitanje šta su operaciona istraživanja mogu dati sledeće konstatacije:

- Operaciona istraživanja su skup modela, kvantitativnih metoda i algoritama, pomoću kojih se određuje najpovoljnije rešenje složenih problema iz svih oblasti delovanja ljudi.
- Operaciona istraživanja predstavljaju naučni pristup donošenja odluka, kojim se istražuje kako na najbolji način dizajnirati i funkcionisati sistem, obično pod uslovima postojanja potrebe za alokacijom oskudnih resursa.
- Predmet kojim se bave operaciona istraživanja vezan je za upravljanje organizacionim, tehničkim i drugim sistemima s ciljem istraživanja optimalnih rezultata za pripremanje i donošenje upravljačkih odluka i praćenje njihovog sprovođenja u odnosu na planirane ili očekivane efekte.
- Operaciona istraživanja se bave modeliranjem determinističkih i stohastičkih sistema iz najrazličitijih svera ljudskog života i donošenjem odluka o njima i u njima.

- Pošto stručnjak u oblasti operacionih istraživanja sistematski traži i utvrđuje optimalnu od raspoloživih mogućnosti, može da se nada da će time pružiti bolji putokaz nego zastareli uobičajeni iskustveni metod.

"Primena naučnih metoda u kompleksnim problemima vezanim za upravljanje velikim sistemima ljudi, mašina, materijala i novčanih sredstava u industriji, poslovanju, javnoj upravi i vojnoj industriji" je definicija koja se nalazi u zaglavlju svakog broja jednog od najuglednijih naučnih časopisa za operaciona istraživanja: *Journals of Operation Research*. U pitanju je takav naučni pristup koji u sebi uključuje merenje određenih faktora, sa kojima se predviđaju i porede ishodi alternativnih odluka, strategija i upravljačkih akcija.

Glavni cilj je pomoći donosiocu odluka da naučno odabere svoj vid politike i način upravljanja. Da bi se odredila optimalna politika ili odlučivanje, primenjuju se razne naučne metode i tehnike, i to:

Matematičko programiranje se koristi za nalaženje minimalne ili maksimalne vrednosti funkcija cilja više promenjivih pri zadatim ograničenjima. Tu spadaju:

- linearno programiranje,
- celobrojno programiranje,
- kvadratno programiranje,
- nelinearno programiranje,
- geometrijsko programiranje,
- dinamičko programiranje,
- stohastičko programiranje,
- mrežno planiranje,
- teorija igara.

Stohastički procesi se koriste u opisivanju skupa slučajnih promenjivih kojima je poznata distribucija verovatnoće. U ovu grupu spadaju:

- teorija odlučivanja,
- Markovljevi procesi,
- redovi čekanja,
- upravljanje zalihama,
- teorija simulacije,
- teorija pouzdanosti.

Statističke metode omogućavaju analiziranje eksperimentalnih podataka i, uz izgradnju empirijskih modela, dobijanje najtačnije predstavljanje fizikalnosti problema, kako bi se dobila aproksimacija stvame situacije, kao što je:

- regresiona analiza,
- klaster analiza,
- planiranje eksperimenata,
- faktorska analiza.

U ovoj knjizi su obrađene sledeće oblasti i teme:

1. Linearno programiranje
2. Nelinearno programiranje
3. Dinamičko programiranje
4. Optimalno rezerviranje

Linearno programiranje obuhvata grafičku metodu, simpleks metodu, dualni problem i postoptimalnu analizu. Takođe, ovu oblast čine i transportni problem (opšti model transportnog problema, metode za pronalaženje početnog rešenja, metode za pronalaženje optimalnog rešenja, degeneracija u transportnom problemu, otvoreni i zatvoreni model transportnog modela i dr.), zatim metode raspoređivanja i celobrojno linearno programiranje.

Nelinearno programiranje obuhvata klasifikaciju rešivih zadataka (nelinaearno programiranje sa linearnim skupom ograničenja, nelinaearno programiranje sa separabilnom funkcijom cilja, kvadratno programiranje, celobrojno programiranje,...), kao i metode rešavanja zadataka nelinaearnog programiranja (Kun-Takerova metoda, gradijentna metoda, kvadratno programiranje, separabilno programiranje, celobrojno programiranje,...)

Dinamičko programiranje obuhvata funkcije i vrste procesa dinamičkog programiranja, opšte karakteristike i primena dinamičkog programiranja (prosta raspodela jednorodnog resursa, raspodela poslova na mašine, optimalna politika zamene opreme,...)

Optimalno rezerviranje obuhvata pojmovi i oznake optimalnog rezerviranja, postavku zadataka optimalnog rezerviranja i metode rešavanja zadataka optimalnog rezerviranja.

1. RAZVOJ, KARAKTERISTIKE I OBLAST PRIMENE OPERACIONIH ISTRAŽIVANJA

1.1. ISTORIJSKI RAZVOJ OPERACIONIH ISTRAŽIVANJA

Operaciona istraživanja su nastala početkom četrdesetih godina u Velikoj Britaniji, formiranjem grupe istraživača, čiji je osnovni zadatak bio da primenom postojećih i razvojem novih naučnih metoda i tehnika na kvantitativnim osnovama daju odgovore po pitanju najboljeg ili "dovoljno dobrog" funkcionisanja: tehničkih, organizacionih, ekonomskih i drugih sistema u postojećim uslovima. Potrebno je napomenuti da je to period početka II svetskog rata kada se Velika Britanija našla pred problemom odbrane od napada sa mora i iz vazduha. Obim proizvodnje i iskazane potrebe su bile u toj meri neusklađeni da je bilo nužno na takav način rasporediti oskudne resurse za pojedine ratne operacije, da ukupni efekti njihove upotrebe budu najpovoljniji. Pošto se problem protiv vazdušne odbrane nije mogao rešiti na zadovoljavajući način, britanski vojni organi su se obratili za pomoć naučnicima i stručnjacima, da analiziraju i daju predloge za ogranizovanje u vezi sa nekoliko vojnih problema, kao što su na primer: izbor mesta radarskim stanicama, upravljanje kretanjem vojnim konvojima, organizovanje operacija bombardovanja, protiv podmorničke zaštite i miniranja.

U toku rata, pri komandama armija Engleske i njihovih saveznika, SAD, Kanade i Francuske, formirana su odeljenja sastavljena od naučnika čiji je zadatak bio da rešavaju probleme vojne prirode. Ovi timovi naučnika bili su pridodati izvršnim vojnim organima zaduženim za izvođenje operacija. Njihov rad je postao poznat u Velikoj Britaniji kao „operaciono istraživanje“, a u SAD pod nizom različitih naziva kao: operaciona istraživanja (*operations research*), analiza operacija (*operations analysis*), procena operacije (*operations evaluation*), sistemska analiza (*systems analysis*), procena sistema (*systems evalution*), i nauka o upravljanju (*menagement science*). Naziv operaciona istraživanja je bio i ostao najšire u upotrebi u SAD. Tako nastaju matematičke metode koje su danas poznate pod imenom *Operaciona istraživanja*.

Nakon rata usledio je period ubrzanog industrijskog razvoja velikog broja zemalja. Resursi proizvodnje su bili ograničeni, a istovremeno pojedini delovi velikih industrijskih giganata su postavljali najčešće međusobno konfliktne ciljeve. Logično je da su, u takvim uslovima privređivanja, timovi operacionih istraživača dobijali sve značajniju ulogu u rešavanju nastalih problema, usklađivanju ciljeva, raspoređivanju oskudnih resursa u svim oblastima privrede i nadgradnje: industriji, poljoprivredi, trgovini, ostalim pratećim delatnostima, državnoj upravi, itd. U privredno razvijenim zemljama veoma brzo su stvoreni uslovi, a zatim i interesovanje, za razvoj metoda operacionih istraživanja i njihovu primenu.

Džordž Dancig (*George Dantzig*) je još 1951. god. osmislio i predstavio naučnoj javnosti *simpleks algoritam*, a u narednom periodu, tokom pedesetih godina, od strane istraživača razvijen je i predložen veliki broj novih metoda za rešavanje formulisanih problema. Do kraja pedesetih, određene tehnike linearnog programiranja, dinamičkog programiranja, metoda optimizacije zaliha i teorije redova čekanja postale su standardne u primeni. Mnogi stručnjaci smatraju da je linearno programiranje jedan od najznačajnijih alata koji su doprineli privrednom i društvenom razvoju XX veka.

Tih godina pojavljuje se i elektronski računar koji je suočio industrijski menadžment sa mogućnošću automatizacije. Čoveka sve više zamenjuju mašine, jer su sposobnije da brzo i tačno obrađuju veliku količinu podataka. Računar je obezbedio varnicu koja je započela ono što se ponekad naziva drugom industrijskom revolucijom. U cilju eksploatacije nove tehnologije kontrole, industrijski menadžment kompanija je počeo da se obraća naučnicima za pomoć, onako kao što su to činili vojni rukovodioci pre njih. Oni su apsorbovali naučnike i istraživače iz oblasti operacionih istraživanja koji su jedan za drugim prelazili iz vojske i podsticali akademske institucije da obrazuju još kadrova za rad u ovoj oblasti. U jednom periodu bilo je bar isto toliko stručnjaka iz ove oblasti u akademskim, državnim i industrijskim organizacijama, koliko ih je bilo u vojnim. Udruženje operacionih istraživača Amerike osnovano je 1953. god., a ovaj primer slede i drugi, tako da je 1973. god. osnovana Međunarodna federacija operacionih istraživača. Knjige i časopisi iz ove olasti počeli su se pojavljivati na velikom broju različitih jezika. Univerzitetski kursevi i nastavni programi za operaciono istraživanje počeli su se množiti u SAD i u drugim zemljama.

Brz razvoj računarske tehnike i softvera doprineo je unapređenju razvoja kvantitativnih metoda, uticajući istovremeno i na njihovu što širu primenu. Praktična primena metoda operacionih istraživanja skoro uvek zahteva obavljanje velikog broja obimnih računskih operacija. Računarska tehnika pripada našoj sadašnjosti, a njihovi kapaciteti, zajedno sa razvijenim računarskim programima, pružaju mogućnost za obavljanje različitih aspekata podrške odlučivanju.

1.2. KARAKTERISTIKE OPERACIONIH ISTRAŽIVANJA

Uopšteno se može reći da operaciona istraživanja obuhvataju skup kvantitativnih naučnih metoda, čiji je zadatak stvaranje realno kvantitativne podloge, koja će omogućiti prihvatanje i donošenje optimalnih privrednih odluka. Iz ove definicije proizilaze i dve osnovne karakteristike metoda operacionih iatraživanja, a to su:

– upotreba *matematičkih* i *statističkih* modela i metoda za rešavanje problema, i

– pomoću metoda i tehnika operacionih istraživanja pronalaze se *optimalna* rešenja posmatranih problema.

Iz osnovnih karakteristika operacionih istraživnja proizilaze i druge karakteristike, koje su takve prirode da deluju i kao potrebni uslovi, kao što su:

– problem koji se rešava mora se predstaviti pogodnim matematičkim modelom,
– pri rešavanju problema potreban je timski rad stručnjaka iz raznih oblasti,
– kvantitativno istraživanje problema zahteva brojne i pouzdane polazne podatke,
– razvoj i uspešna primena metoda operacionih istraživanja podrazumeva korišćenje elektronske računske tehnike.

1.3. OBLAST PRIMENE OPERACIONIH ISTRAŽIVANJA

Područje primene metoda operacionih istraživanja veoma je široko. Njima se mogu rešavati najrazličitiji makroekonomski i mikroekonomski problemi. To su, pre svega, problemi upravljanja ekonomskim sistemima vezani za racionalno korišćenje proizvodnih resursa. Do sada su najveću pažnju istraživača privlačili, i najčešće su rešavani, sledeći problemi:

– proizvodni problemi, među kojima se izdvajaju:
    ✓ određivanje optimalnog programa proizvodnje,
    ✓ problemi poslovne saradnje i integracije,
    ✓ problemi proširenja kapaciteta,
    ✓ problemi upravljanja zalihama,
    ✓ problemi održavanja i servisiranja opreme,
    ✓ problemi zamene sredstava i opreme,
    ✓ određivanje redosleda operacija i poslova,
    ✓ kalendarsko planiranje poslova i drugi,
– izrada programa investicionih ulaganja,
– transportni problemi,
– lokacioni problemi,
– problemi raspoređivanja,
– utvrđivanje plana spoljnotrgovinske razmene,
– rešavanje različitih konfliktnih situacija i drugi.

Optimizacija u širem smislu se može primeniti kod rešavanja bilo kog problema. Nekoliko primera primene u različitim industrijskim disciplinama:

- Konstruisanje aviona najmanje težine.
- Iznalaženje optimalne putanje vazdušne letilice.
- Projektovanje transportnih sistema.
- Projektovanje sistema rešetkastih konstrukcija, mostova, tornjeva itd.
- Proizvodnja sistema navodnjavanja sa maksimalnim stepenom iskorišćenja.
- Optimalno projektovanje reduktora.

- Izbor režima rezanja pri obradi metala sa minimalnim proizvodnim troškovima.
- Najkraći put trgovca pri putovanju u posetu različitim gradovima.
- Optimalno planiranje proizvodnje i kontrole.
- Optimalno planiranje realizacije projekta.
- Optimalno održavanje i zamena opreme u cilju smanjenja troškova.
- Optimalni raspored operacija i korigovanje rasporeda mašina.
- Planiranje najbolje strategije ostvarenja najvećeg profita.
- Optimalno planiranje zaliha.

1.4. ETAPNI PRISTUP REŠAVANJA PROBLEMA

Konkretno rešavanje određenog problema, korišćenjem metoda operacionih istraživanja, odvija se u nekoliko etapa:

- Početna etapa je izbor i definisanje problema. Na osnovu kvalitativne analize problema specificiraju se uslovi koji određuju mogućnost i kvalitet rešenja problema kao i cilj rešavanja problema.
- Izbor metode za rešavanje problema. U zavisnosti od vrste i prirode problema biramo odgovarajuće metode kojima ćemo problem rešiti.
- Prikupljanje i obrada potrebnih polaznih podataka o problemu. Posebna pažnja mora se posvetiti pitanju pouzdanosti pripremljenih podataka.
- Formiranje matematičkog modela. Na osnovu prikupljenih podataka i kvalitativne analize formira se pogodan matematički model koji služi za rešavanje problema. Na osnovu cilja problema rešavanja formira se funkcija cilja (funkcija kriterijuma) a ostali uslovi se kvantifikuju i čine sistem ograničenja.
- Rešavanje problema pomoću modela i izabrane metode. U ovoj etapi se formirani matematički model rešava korišćenjem odgovarajuće metode i algoritma uz korišćenje elektronskog računara.
- Analiza dobijenog optimalnog rešenje. Analiziraju se ekonomski efekti koje će obezbediti dobijeno rešenje, analizira se stabilnost tog rešenja u odnosu na moguće promene polaznih uslova, kao i uslovi i potrebne pripreme za realizaciju rešenja. Na osnovu tih analiza pripremiće se potrebne odluke koje će obezbediti primenu optimalnog rešenja.

2. LINEARNO PROGRAMIRANJE

2.1. UVOD

Linearno programiranje (LP) je najpoznatija i najviše primenjivana metoda operacionih istraživanja, koja omogućava rešavanje velikog broja problema. Najčešće se izučava u oblasti planiranja ekonomskog razvoja, gde se rešavaju praktični zadaci kako na nivou preduzeća tako i na širem regionalnom ili najširem društvenom planu.

Prvi rad iz oblasti linearnog programiranja objavio je sovjetski naučnik L.V. Kantorovič, pod nazivom “Matematičke metode u organizaciji i planiranju proizvodnje” 1939. godine, gde se po prvi put definiše transportni zadatak. Međutim, u prve radove iz ove oblasti spada i rad pukovnika Vlastimira Ivanovića iz 1940. god., pod nazivom “Pravila za proračun potrebnog broja transportnih sredstava”, gde je pokazano kako se izračunava minimalan broj vozila za prevoz date količine materijala korišćenjem ”Principa najveće ekonomije”. Po nekim autorima se nastanak linearnog programiranja vezuje za članak koji je napisao J. Egervary 1931. godine pod nazivom “Matrične kombinatorne osobine”. Prvi algoritam za rešavanje transportnog problem razradio je Frenk L. Hitchcock 1941. godine u radu “Distribucija jednog proizvoda iz više izvora do većeg broja lokaliteta”. Na formulisanju transportnog problema i njegovom rešavanju radio je i T.C. Koopmans, koji je rezultate svojih istraživanja objavio 1947. godine.

Najveći doprinos razvoju linearnog programiranja dao je George B. Dantzig, koji je 1947. godine formulisao opšti problem linearnog programiranja i postavio simpleks metod. Radovi John von Neuman-a iz tog perioda su omogućili teorijsko formulisanje dualnog problema kao i pronalaženje veze između linearnog programiranja i teorije igara. Metod za otklanjanje degeneracije predložen je u radu “Optimalnost i degeneracija kod linearnog programiranja”. Značajni su i radovi S. Gass-a i T. Saaty-a iz 1955. god. o parametarskom programiranju kao i radovi E. Beale-a i R. Gomory-a iz 1958. god. o celobrojnom programiranju.

Problemi matematičkog programiranja javljaju se u različitim disciplinama. Na primer, menadžer na berzi mora da odabere ulaganja koja će generisati najveći mogući profit a da pri tome rizik od velikih gubitaka bude na unapred zadatom nivou; menadžer proizvodnje organizuje proizvodnju u fabrici tako da količina proizvoda i kvalitet budu maksimalni a utrošak materijala, vremena i škart minimalni, pri čemu ima na raspolaganju ograničene resurse (broj radnika, kapacitet mašina, radno vreme); naučnik pravi matematički model fizičkog procesa koji najbolje opisuje određenu fizičku pojavu, uz uslov da model ne sme biti suviše komplikovan, a na raspolaganju ima konačni broj mernih rezultata.

U svim ovim situacijama možemo da identifikujemo tri zajednička pojma:

- Postoji globalna veličina (cilj) koja se želi optimizirati (profit, razlika između predviđanja modela i eksperimentalnih podataka).
- Uz globalni cilj obično su prisutni dodatni zahtevi ili ograničenja, koja moraju biti zadovoljena (ograničen rizik, resursi, kompleksnost modela).
- Postoje određene veličine tako da ako se njihove vrednosti izaberu ”dobro”, zadovoljeni su i cilj i ograničenja. Te veličine se nazivaju optimizacione promenljive ili parametri.

Znači, da bi se zadao problem matematičkog programiranja potrebno je:

- odabrati jednu ili više optimizacionih promenljivih,
- odabrati funkciju cilja i
- formirati skup ograničenja.

Nakon toga, identifikuje se klasa problema kojoj dobijeni matematički model pripada i bira se metod za njegovo rešavanje.

Postoji više metoda za rešavanje problema linearnog programiranja. Geometrijski metod je primenljiv na probleme kod kojih je broj promenljivih n=2 ili kada je $n-m$=2, gde je m broj ograničenja. U principu, zadatak linearnog programiranja se može rešiti geometrijski i u slučaju kada je n=3. Nedostatak geometrijskog metoda je u tome što ne rešava opšti zadatak linearnog programiranja već samo neke specijalne slučajeve. Do prvog opšteg metoda za rešavanje problema linearnog programiranja došao je američki matematičar Dancig (*Dantzig*) 1947. godine. U svom radu je formulisao opšti oblik problema linearnog programiranja i dao algoritam za njegovo rešavanje, poznat kao ***simpleks metod***. Dantzig-ov rad je više godina kružio među stručnjacima i poslužio kao osnova svim narednim razmatranjima problema linearnog programiranja. Iako su u međuvremenu pronađeni i drugi metodi, ovaj metod se i danas koristi kroz brojne modifikacije. I geometrijski i simpleks metod traže ekstremne vrednosti (maksimum, minimum) funkcije cilja na rubovima oblasti ograničenja.

U kasnijem periodu se pojavljuju i alternativni pristupi rešavanju problema linearnog programiranja, kao što su radovi u kojima se koriste generilisani inverzi, radovi Conn-a i Dax-a koji ne koriste simpleks metod. Iako je praksa pokazala da je simpleks metod veoma efikasan, 1972. godine su V. Klee i G.L. Minty dokazali da on nije polinomijalan. Oni su konstruisali jednostavan primer zadatka linearnog programiranja kod koga je dopustivi skup deformisana n-dimenzionalna kocka sa 2^n temena, za čije je rešavanje simpleks metodu sa standardnim izborom vodećeg elementa potrebno 2^n-1 iterativnih koraka. Prvi polinomijalni algoritam za rešavanje problema linearnog programiranja je dao Hačijan (*Khachian*) 1979. godine. Time je napravljen veliki zaokret u razvoju linearnog programiranja. Hačijan je pokazao da njegov metod elipsoida rešava problem linearnog programiranja za $O(n^4L)$ elementarnih aritmetičkih operacija, gde je L broj bitova potrebnih za zapis svih parametara problema (matrice ograničenja, funkcije cilja i desne strane ograničenja). Hačijan je rešio vrlo važno teorijsko pitanje. Međutim, ispostavilo se da metod elipsoida nije primenljiv u praksi. Testiranja su pokazala da je simpleks metod daleko efikasniji, jer on “retko” dostiže gornju granicu složenosti, za razliku od metoda elipsoida, koji to često čini. Ovakav zaključak je inicirao da se pronađu novi metodi za rešavanje problema linearnog programiranja koji će osim polinomijalne složenosti imati i praktičnu primenljivost. Prvi takav metod predložio je 1984. godine N. Karmarkar. Karmarkar-ov algoritam je neke test primere rešavao i do 50 puta brže od simpleks metoda. Karmarkarov rezultat je izazvao pravu revoluciju u razvoju ove oblasti.

Posle Karmarkarovog metoda pojavila se čitava familija metoda koji su poznati pod nazivom unutrašnji metodi. Danas su ***primar-dual*** metodi unutrašnje tačke dominantni za rešavanje problema linearnog programiranja. Iako su otkriveni polinomijalni metodi, simpleks metod se i danas koristi i kroz brojne modifikacije još uvek živi. Simpleks metod je mnogo bolji od metoda unutrašnje tačke na tzv. loše uslovljenim problemima, zbog svoje spore, ali pouzdane konvergencije. Isto tako, u praksi se često javlja potreba za rešavanjem klase srodnih problema gde se optimalno rešenje jednog od njih može efikasno iskoristiti kao početna tačka za rešavanje ostalih problema. I u ovom slučaju je simpleks metod bolji izbor od metoda unutrašnje tačke.

Unutrašnji metodi dele se na *primarne, dualne* i *primarno-dualne.*

Primarni metodi rešavaju problem linearnog programiranja u standardnom obliku:

$$\begin{aligned} &\min \quad C^T \cdot x, \\ &\text{p.o.} \quad A \cdot x = b, \qquad x \geq 0, \end{aligned} \tag{2.1.1.}$$

Dualni metodi rešavaju dualni problem oblika:

$$\begin{aligned} &\max \quad b^T y, \\ &\text{p.o.} \quad A^T y + s = c, \qquad s \geq 0. \end{aligned} \tag{2.1.2.}$$

gde je A matrica tipa $m \times n$, dok su a, b i c vektori odgovarajućih dimenzija.

Primarno-dualni metodi rade simultano (istovremeno) i sa primarnim i sa dualnim problemom (primar-dual metodi). Danas su primar-dual metodi dominantni. Izučavanje metoda unutrašnje tačke za rešavanje problema linearnog programiranja pripada savremenom i modernom trendu u svetu što potvrđuje veći broj radova. Poslednjih desetak godina se objavljuje veliki broj radova iz ove oblasti tako da bibliografija broji više hiljada naslova.

Postoje tri alata u rešavanju problema linearnog programiranja:

- *Modeli*: formulacija problema u detaljnim matematičkim terminima.
- *Algoritmi*: tehnike za rešavanje modela.
- *Kompjuteri i softveri*: mašine za izvršavanje algoritamskih koraka.

Postoji veći broj programskih paketa, koji su našli široku primenu u praksi, kao:

HOPDM je napisan u programskom jeziku FORTRAN.

LIPSOL je napisan u programskim jezicima MATLAB i FORTRAN.

LOQO je napisan u programskom jeziku C.

PCx je napisan u programskim jezicima C i FORTRAN..

MOSEK je napisan u programskom jeziku C++ i predstavlja jedan od najjačih solvera ne samo za probleme linearnog programiranja već uopšte i za probleme kvadratnog i nelinearnog programiranja. Za razliku od predhodno pomenutih programa, ovaj program je komercijalan.

LINDO (Linear Interactive and Discrete Optimizer) je interaktivni softverski paket koji se može koristiti za rešavanje problema linearnog programiranja. Razvijen je 1980. godine i od tada je prilagođen Windows okruženju i grafički orijentisanim programima. Softverski paket LINDO se koristi za rešavanje problema zadatih direktno sa tastature.

MarPlex je domaći (srpski) program. Napisan je u programskom jeziku Visual Basic 6.0 i koristi modifikacije simpleks metoda, Izdvaja se od ostalih programa zbog svog jednostavnog interface-a kao i ne toliko obimnog i čitljivog koda. Pošto je implementiran u Visual Basic-u, i koristi simpleks metod, zaostaje po pitanju brzine.

RevMarPlex je takođe domaći program i predstavlja revidiranu verziju programa MarPlex napisanu u programskom jeziku MATHEMATICA.

2.2. OPŠTI ZADATAK LINEARNOG PROGRAMIRANJA

Linearno programiranje (LP) je matematička disciplina koja izučava metode za iznalaženje ekstremne (maksimalne i minimalne) vrednosti linearnih funkcija cilja (kriterijuma) sa dve ili više promenljivih, pod uslovom da one zadovoljavaju određeni broj ograničenja postavljenih u odgovarajuće linearne odnose (jednačine ili nejednačine). Ovi metodi koriste se za rešavanje velikog broja industrijskih, ekonomskih, vojnih i drugih zadataka, pri čemu se traži njihovo najbolje (optimalno) rešenje sa nekog odgovarajućeg stanovišta.

Da bi se određeni ekonomski zadaci i zadaci drugih oblasti rešavali metodama linearnog programiranja, neophodno je da se odgovarajuće pojave predstave u matematičkoj formi. Proces predstavljanja industrijskih i ekonomskih pojava u matematičkoj formi, poznat je pod imenom: *ekonomsko-matematički modeli* industrijskih i ekonomskih procesa.

Matematičko formulisanje zadataka, rešavanih metodom linearnog programiranja, je vrlo važan pristup, jer se, bez pravilno postavljenog matematičkog modela nekog procesa, ne bi mogao koristiti metod linearnog programiranja, niti bi se dobili kvalitetni rezultati pri rešavanju zadataka. Kod modela bilo kog zadatka teži se da se održi suština i specifičnost odgovarajuće pojave, tj. procesa. Model svakog zadatka linearnog programiranja sastoji se od tri osnovna dela:

1. Linearna funkcija, kojom se cilj zadatka predstavlja u matematičkom obliku. Ona zavisi od više promenljivih., i naziva se linearna forma, linearna funkcija, funkcija cilja, ekonomska funkcija ili funkcija kriterijuma optimalnosti. Uobičajeno obeležavanje je $F(x)$, $f(x)$, ili $z(x)$.
2. Sistem linearnih jednačina i/ili nejednačina kojima se u matematičkom obliku predstavljaju ograničavajući uslovi. Uobičajeni naziv za ovu vrstu ograničenja je da su to, jednostavno, ograničenja problema LP.
3. Sistem nejednačina koje izražavaju uslov za nenegativnost rešenja, tj. sva rešenja moraju biti pozitivni brojevi računajući i nulu. U literaturi se često sreće da su to prirodna ograničenja problema.

Kod svih problema, zadatak se svodi na to da se pronađe optimalno rešenje između svih mogućih rešenja koja zadovoljavaju neke unapred postavljene uslove, odnosno ograničenja. To znači da, rešavanje zadataka linearnog programiranja predstavlja određivanje takvih nenegativnih vrednosti programa proizvodnje koji ima p_1, p_2,..., p_n proizvoda, a čije su količine x_1, x_2,..., x_n tako da te količine predstavljaju optimalni program uz određeni kriterijum optimalnosti i sa odgovarajućim ograničavajućim uslovima.

Na primer, ukoliko je kriterijum optimalnosti maksimiziranje dobiti, potrebno je, iz obračunskih kalkulacija, utvrditi dobit po jedinici proizvoda, tj. za količine proizvoda x_1, x_2,..., x_n odrediti dobit c_1, c_2,..., c_n izražen u novčanim jedinicama po komadu (*nj*/*kom*).

Za ograničavajuće uslove, obzirom na specifičnost proizvodnog procesa, potrebno je imati podatke vezane za:

- sredstva za rad: vreme obrade po jedinici proizvoda na odgovarajućim mašinama M_i (časovi, minuti,...), kao i njihov kapacitet (maksimalno raspoloživo vreme) za svaku vrstu mašina (*čas*/*godina*, *čas*/*mesec*,...);
- sirovine S_i: norma potrošnje po jedinici proizvoda (*kg*/*kom*, *l*/*kom*,...), kao i njihov kapacitet, tj. ukupna količina za dati program (*kg*/*god*, *kg*/*mes*,...).
- uslovi na tržištu: konzumna moć tržišta, tj. količina planirane prodaje robe u razmatranom periodu (*količina*/*godina*), porudžbine od strane kupaca izražene, takođe, u količini robe u razmatranom periodu (*kol*/*god*).

Zbog lakšeg uočavanja zadatih parametara i matematičkog modeliranja, svi ovi podaci se mogu prikazati tabelarno, kao što je prikazano u tabeli 2.2.1.

Tabela 2.2.1. *Tabelarni prikaz raspoloživih resursa*

Resursi	Proizvodi (p_1, p_2,..., p_n)				**Kapacitet** (raspoloživo vreme, količina,...)
	x_1	x_2	...	x_n	
M_1					
M_2					
...					
M_p					
S_1					
S_2					
...					
S_k					

Ograničavajući uslovi se mogu proširiti i na ostala područja, zavisno od poslovne politike preduzeća, kao i na spoljne faktore koji utiču na obim proizvodnje.

Opšti zadatak linearnog programiranja formuliše se na sledeći način: Traže se vrednosti promenljivih x_1, x_2,..., x_n koje odgovaraju uslovima nenegativnosti

$$x_1 \geq 0,\ x_2 \geq 0,\ ...,\ x_n \geq 0 \tag{2.2.1.}$$

sa kojima se postiže ekstremna (maksimalna ili minimalna) vrednost linearne funkcije cilja

$$F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.2.2.}$$

i zadovoljava sistem ograničenja

$$\begin{matrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{matrix} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \begin{matrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{matrix} \tag{2.2.3.}$$

$$i = 1, 2, ..., m; \quad j = 1, 2, ..., n$$

gde su:

– m i n proizvoljni prirodni brojevi;
– koefisijenti uz promenljive u funkciji cilja c_j, i sistemu ograničenja a_{ij} su poznati, proizvoljni realni brojevi;
– slobodni članovi b_i, sa desne strane sistema ograničenja, su proizvoljni nenegativni brojevi (u suprotnom slučaju relacije se množe sa – 1).

Zadatak linearnog programiranja može da se prikaže i u sažetom obliku, kao:

$$F(x) = \sum_{j=1}^{n} c_j x_j \tag{2.2.4.}$$

$$\sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ = \\ \geq \end{Bmatrix} b_i \quad i = 1, 2, \ldots, m \qquad (2.2.5.)$$

$$x_j \geq 0 \qquad j = 1, 2, ..., n \qquad (2.2.6.)$$

Nejednačine (2.2.3,) ili (2.2.5.) se mogu transformisati u ekvivalentne jednačine dodavanjem dopunskih i/ili veštačkih promenljivih (x_{n+1}, x_{n+2},..., x_{n+m}) sa leve strane znaka nejednakosti. Postupak prevođenja nejednačina u jednačine je prikazan kao zasebna celina (videti poglavlje 2.4.2.).

2.3. OBLICI REŠENJA LINEARNOG PROGRAMIRANJA

Rešenje x=(x_1, x_2,..., x_n) u primenama najčešće predstavlja plan ili program proizvodnje, prevoza, i sl., pa je otuda ovaj zadatak dobio naziv „programiranje“, a naziv „linearno programiranje“ označava da su ograničenja promenljivih (2.2.3.) i funkcije cilja (2.2.2.) linearnog karaktera. Postoje tri oblika rešenja zadataka linearnog programiranja: nedopustivo, dopustivo i optimalno rešenje.

Nedopustivo rešenje sistema linearnog problema naziva se vektor x=(x_1,x_2,...,x_n) čije komponente ne zadovoljavaju uslove nenegativnosti sistema jednačina (2.2.3.).

Proizvoljno rešenje sistema nejednačina (2.2.3.) može se zamisliti kao vektor x= (x_1, x_2,..., x_n), koji u geometrijskoj interpretaciji predstavlja tačku k-dimenzionalnog prostora R^k (realnih brojeva). Vektor x=(x_1, x_2,..., x_n) predstavlja ***dopustivo*** **(*moguće*)** ***rešenje*** linearnog problema ukoliko zadovoljava uslove sistema ograničenja (2.2.3.) i uslov nenegativnosti (2.2.1.).

Uočava se da vektor x=(x_1, x_2,..., x_n) ima n elemenata, što znači da dopustivo rešenje ima n promenljivih. Broj nezavisnih jednačina u sistemu je m (broj ograničenja). Kako je broj promenljivih veći od broja nezavisnih jednačina ($n > m$), neke od promenljivih dopustivog rešenja biće jednake nuli, a neke će biti pozitivne. U zavisnosti od broja pozitivnih promenljivih (r) u dopustivom rešenju razlikujemo:

– ***bazično dopustivo rešenje*** – to je dopustivo rešenje koje nema više od m pozitivnih promenljivih ($r \leq m$),
  - ✓ ***nedegenerisano rešenje*** – to je dopustivo rešenje koje ima tačno m pozitivnih promenljivih ($r = m$),
  - ✓ ***degenerisano rešenje*** – to je dopustivo rešenje koje sadrži manje od m pozitivnih promenljivih ($r < m$),

– ***nebazično dopustivo rešenje*** – to je dopustivo rešenje koje ima više od m pozitivnih promenljivih ($r > m$).

Optimalno rešenje je ono dopustivo rešenje koje pored zadovoljenja uslova sistema ograničenja (2.2.3.) i uslov nenegativnosti (2.2.1.), zadovoljava i da funkcija cilja (2.2.2.) dostigne svoju ekstremnu vrednost (maksimum ili minimum). U literaturi je uobičajen zapis x^*. Problem linearnog programiranja svodi se na rešavanje optimalnog (najboljeg) rešenja pri zadatim ograničenjima.

Ukoliko sa Ω_P označimo skup dopustivih rešenja, onda je $\Omega_P \subset R^k$. Za skup Ω_P pretpostavljamo da nije prazan i da sadrži najmanje jedan element (slučaj kada zadatak 2.2.1 – 2.2.3 ima rešenje). Optimalno rešenje $x^*=(x_1^*,..., x_n^*)\in\Omega_P$ zadatka linearnog programiranja je ono dopustivo rešenje za koje funkcija cilja $F(x)=F(x_1, x_2,..., x_n)$ dostiže maksimum (minimum). U slučaju maksimizacije funkcije cilja ispunjen je uslov

$$F(x^*) = \max_{x\in\Omega_P} F(x) \tag{2.3.1.}$$

ili

$$F(x^*) \geq F(x) \quad \forall x \in \Omega_P$$

Često se u praksi traži da optimalna vrednost funkcije cilja bude najmanja na skupu dopustivih rešenja Ω_P. U ovom slučaju optimalno rešenje $x^*\in\Omega_P$ je ono dopustivo rešenje za koje je ispunjeno

$$F(x^*) = \min_{x\in\Omega_P} F(x) \tag{2.3.2.}$$

ili

$$F(x^*) \leq F(x) \quad \forall x \in \Omega_P$$

U jednom konkretnom zadatku rešava se samo problem maksimizacije, odnosno problem minimizacije. Problem minimuma se može transformisati u problem maksimuma, i obratno, jednostavnim množenjem ciljne funkcije sa – 1.

Propozicija 2.3.1. *Dati kriterijum optimizacije može da se zameni suprotnim, pri čemu ta zamena ne utiče na optimalno rešenje, tj. ako je za $x^*\in\Omega_P$ ispunjeno*

$$F(x^*) = \max_{x\in\Omega_P} F(x)$$

onda je

$$-F(x^*) = \min_{x\in\Omega_P}\left[-F(x)\right] \tag{2.3.3.}$$

i obrnuto.

Dokaz. Prema pretpostavci teoreme ispunjeno je

$$F(x^*) \geq F(x) \quad \forall x \in \Omega_P$$

ukoliko ovu nejednakost pomnožimo sa – 1 dobija se

$$-F(x^*) \leq -F(x) \quad \forall x \in \Omega_P$$

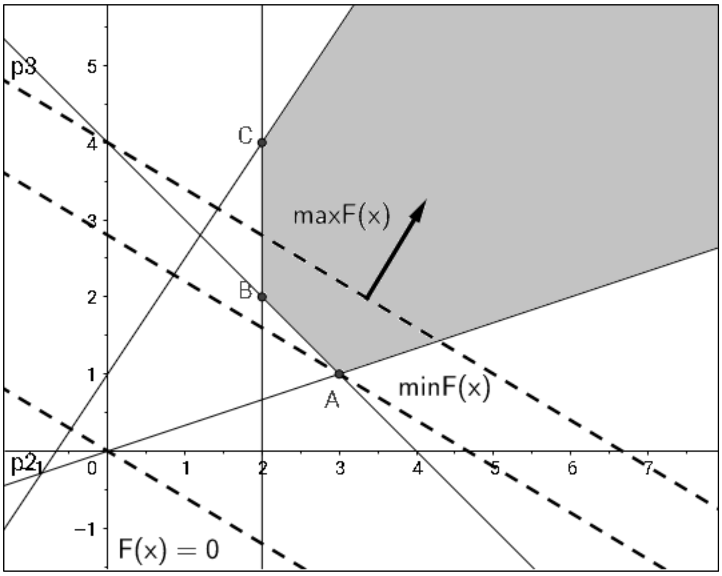
čime je teorema dokazana.

Zadatak linearnog programiranja ima rešenje ako veličina F_{max} (F_{min}) ima konačnu vrednost na skupu Ω_P dopustivih rešenja. Zadatak linearnog programiranja nema rešenja ako sistem nejednačina (2.2.3.) nema nenegativnih rešenja ili ako veličina F_{max} (F_{min}) nema konačnu vrednost.

Ograničenja (2.2.3.) i (2.2.1.) određuju u k-dimenzionalnom prostoru konveksnu oblast Ω_P ograničenu skupom hiperravni

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1, 2, \ldots, m \,.$$

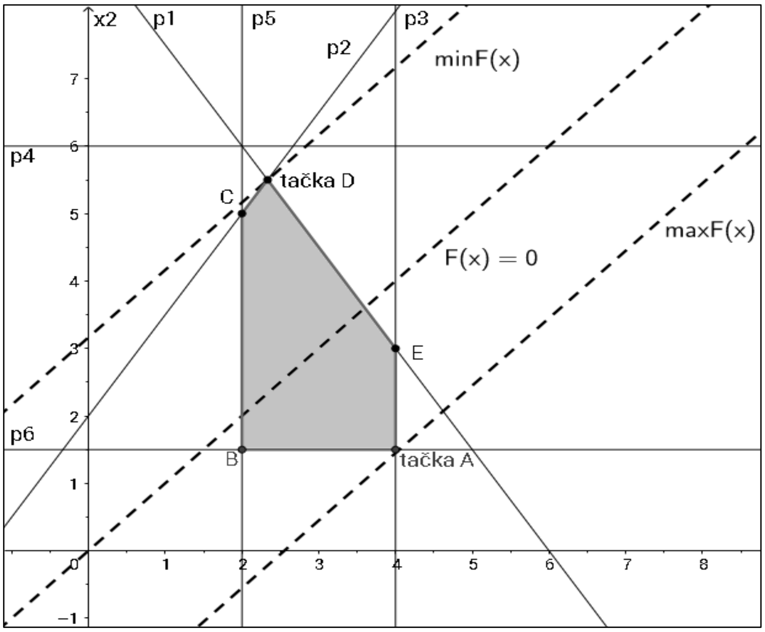
Skup dopustivih rešenja Ω_P se može nazvati poliedrom, mada u izvesnim slučajevima ona može biti i beskonačna. (slika 2.3.1.). Ta oblast se naziva *oblast dopustivih rešenja*. Kako su funkcije koje treba maksimizirati ili minimizirati linearne, klasični matematički metodi pokazuju da se maksimumi (minimumi) funkcije cilja $F(x)=F(x_1, x_2,..., x_n)$ dostižu na granicama oblasti Ω_P koja je određena datim ograničenjima.



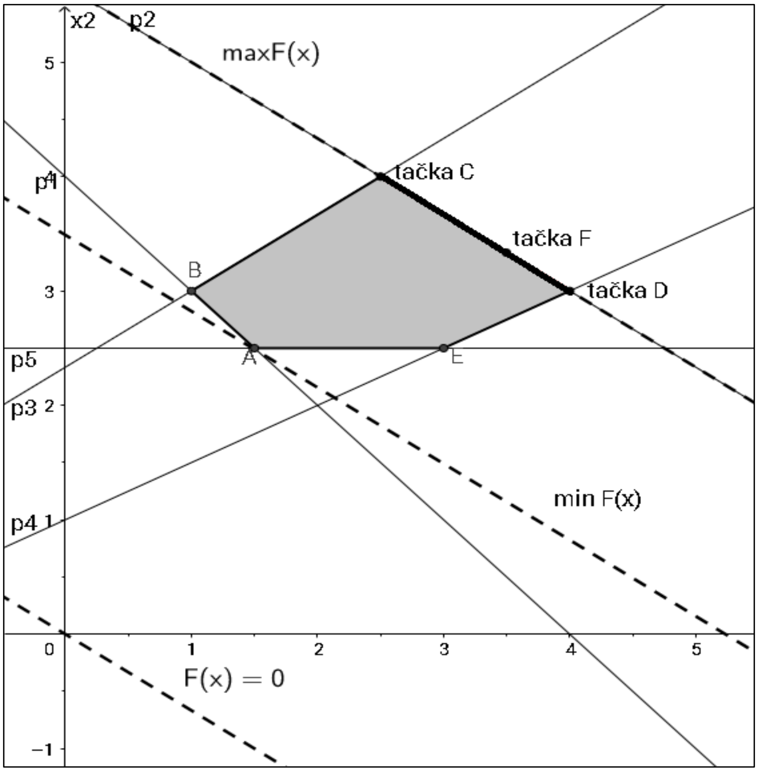
Slika 2.3.1. *Beskonačna oblast dopustivih rešenja*

Ukoliko hiperravan $F(x)=const$ nije paralelna ni sa jednom od pomenutih hiperravni, koje predstavljaju strane poliedra, onda će ciljna funkcija $F(x)$ dostići maksimum (minimum) u jednom od temena poliedra. Tada funkcija cilja ima jedinstveno rešenje. (slika 2.3.2.).

Ako je hiperravan $F(x)=const$ paralelna sa jednom od pomenutih hiperravni, koje predstavljaju strane poliedra, onda će funkcija cilja $F(x)$ dostići ekstremnu vrednost na toj, paralelnoj, strani poliedra. Tada funkcija cilja nema jedinstveno rešenje. (slika 2.3.3.).



Slika 2.3.2. *Oblast dopustivih rešenja u obliku poliedra sa ciljnom finkcijom koja dostiže ekstremne vrednosti u temenima poliedra*



Slika 2.3.3. *Oblast dopustivih rešenja* u obliku poliedra sa ciljnom finkcijom paralelnom *sa jednom stranom poliedra, gde funkcija cilja dostiže ekstremnu vrednost na toj, paralelnoj, strani poliedra*

2.4. PRIKAZIVANJE OPŠTEG ZADATKA LINEARNOG PROGRAMIRANJA

Opšti problem linearnog programiranja se može prikazati u tri oblika:

- skalarni (koordinatni),
- kanonični, i
- vektorsko – matrični.

2.4.1. Skalarni oblik problema linearnog programiranja

Zadatak linearnog programiranja se može iskazati u formalizovanom razvijenom obliku. Finkcija cilja je prikazana relacijom (2.4.1.), sistem linearnih ograničenja relacijama (2.4.2.), a uslov o nenegativnosti rešenja relacijom (2.4.3.).

$$F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \qquad (2.4.1.)$$

pri ograničenjima (p.o.)

$$\begin{matrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{matrix} \begin{Bmatrix} \le \\ = \\ \ge \end{Bmatrix} \begin{matrix} b_1 \\ \cdots \\ b_m \end{matrix} \qquad (2.4.2.)$$

$$x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0 \qquad (2.4.3.)$$

ili sažetije, funkcija cilja relacijom (2.4.4.), ograničenja (2.4.5.) i prirodna ograničenja relacijom (2.4.6.), kao:

$$F(x) = \sum_{j=1}^{n} c_j x_j \qquad (2.4.4.)$$

p.o.

$$\sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \le \\ = \\ \ge \end{Bmatrix} b_i \quad i = 1, 2, \ldots, m \qquad (2.4.5.)$$

$$x_j \ge 0 \qquad j = 1, 2, \ldots, n \qquad (2.4.6.)$$

2.4.1.1. Opšti problem maksimuma

Potrebno je pronaći takve vrednosti vektora promenljivih $x = (x_1, x_2, ..., x_n)$ tako da funkcija cilja dostigne svoju maksimalnu vrednist, kao što je prikazano relacijom (2.4.7.), pri ograničenjima problema koja su sva izražena nejednačinama tipa $\le$, prikazano relacijom (2.4.8.), uz uslov nenegativnosti promenljivih, relacija (2.4.9.).

$$\max F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.4.7.}$$

p.o.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\le b_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\le b_m \end{aligned} \tag{2.4.8.}$$

$$x_1 \ge 0,\ x_2 \ge 0, \ldots, x_n \ge 0 \tag{2.4.9.}$$

u skraćenom obliku, kao:

$$\max F(x) = \sum_{j=1}^{n} c_j x_j \tag{2.4.10.}$$

p.o.

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1, 2, \ldots, m \tag{2.4.11.}$$

$$x_j \ge 0 \qquad j = 1, 2, \ldots, n \tag{2.4.12.}$$

2.4.1.2. Opšti problem minimuma

Treba naći takve vrednosti vektora promenljivih $x = (x_1, x_2, ..., x_n)$ tako da funkcija cilja dostigne svoju minimalnu vrednost, kao što je prikazano relacijom (2.4.13.), pri ograničenjima problema koja su sva izražena nejednačinama tipa $\ge$, prikazano relacijom (2.4.14.), uz uslov nenegativnosti promenljivih, relacija (2.4.15.).

$$\min F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.4.13.}$$

p.o.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\ge b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\ge b_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\ge b_m \end{aligned} \tag{2.4.14.}$$

$$x_1 \ge 0,\ x_2 \ge 0, \ldots, x_n \ge 0 \tag{2.4.15.}$$

u skraćenom obliku, kao:

$$\min F(x) = \sum_{j=1}^{n} c_j x_j \tag{2.4.16.}$$

p.o.

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad i = 1, 2, \ldots, m \qquad (2.4.17.)$$

$$x_j \geq 0 \qquad j = 1, 2, \ldots, n \qquad (2.4.18.)$$

2.4.2. Transformacija nejednačina u jednačine

Primena bilo kog algoritma simpleks metoda zahteva prilagođavanje početnog matematičkog modela. Transformacija postavljenog modela vrši se prevođenjem svih nejednačina u jednačine uvođenjem novih promenljivih.

2.4.2.1. Transformacija nejednačina tipa ≤

Kod ograničenja tipa ≤ uvode se promenljive koje se nazivaju ***dopunske*** ili ***izravnavajuće*** promenljive. U stranoj literaturi se sreće naziv ***slack*** promenljive. Postupak uvođenja dopunskih promenljivih ilustrovan je na primeru ograničenja $2x_1 + 3x_2 \leq 100$. Iz priloženog ograničenja uočava se da je leva strana nejednačine manja ili jednaka desnoj strani. Zbog toga je potrebno dodati jednu promenljivu sa leve strane kako bi dopunili (izravnali) nejednačinu i preveli je u jednačinu. Otuda i potiče naziv promenljivih. Levoj strani dodaje se dopunska promenljiva x_3, i dobija se jednačina: $2x_1 + 3x_2 + x_3 = 100$.

Promenljiva x_3 je, takođe, nenegativna veličina i predstavlja dopunu manje, u ovom slučaju leve, strane nejednačine do potpune jednačine. Uvođenjem svake dopunske promenljive povećava se dimenzija problema koji ima *n* promenljivih za 1. U ovom primeru sistem je imao dve promenljive, a nakon uvođenja dopunske promenljive ima tri. Koeficijenti koji stoje uz dopunske promenljive su jedinice (1). U sistemu ograničenja pomoću dopunskih promenljivih formira se jedinična matrica. Formiranje jedinične matrice je jedan od uslova za primenu sipleks algoritma.

Linearni model je prilagođen za primenu simpleks algoritma kada su ispunjena dva uslova:

1. sve nejednačine moraju biti pretvorene u jednačine, i
2. u tako formiranom prilagođenom modelu mora postojati jedinična matrica.

Primer 2.4.1. Date nejednačine transformisati u jednačine.

$$\begin{aligned} 2x_1 + 3x_2 &\leq 100 \\ 3x_1 \qquad &\leq 200 \\ x_2 &\leq 50 \\ x_1, x_2 \geq 0 & \end{aligned}$$

Rešenje. Leve strane nejednačina su manje ili jednake desnim stranama. Dodavanjem dopunskih promenljivih levim stranama nejednačina dobija se sledeći sistem jednačina:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 \quad\quad\quad &= 100 \\ 3x_1 \quad\quad + x_4 \quad\quad &= 200 \\ x_2 \quad\quad + x_5 &= \ \ 50 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 & \end{aligned}$$

U sažetom obliku sistem jednačina transformisan iz nejednačina tipa $\leq$ u jednačine prikazan je relacijom (2.4.19.).

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \Rightarrow \quad \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i \qquad (2.4.19.)$$
$$i = 1, 2, \ldots, m$$

Pored toga što dovode matematički model u oblik pogodan za primenu simpleks algoritma, dopunske promenljive imaju i odgovarajuće ekonomsko značenje. U primeru 2.4.1., promenljive x_1 i x_2 potiču iz osnovnog problema i označavaju količine proizvoda P_1 i P_2 koje treba proizvoditi. Promenljiva x_1 predstavlja količinu proizvedenog proizvoda P_1, a x_2 količinu proizvedenog proizvoda P_2. Značenje dopunskih promenljivih x_3, x_4 i x_5 vezuju se za prirodu i značenje samih ograničenja kome je promenljiva dodeljena. Ukoliko se prvo ograničenje odnosi na kapacitet radnog mesta RM_1, pri čemu leva strana označava iskorišćeni kapacitet tog radnog mesta za proizvodnju proizvoda P_1 i P_2, onda će dopunska promenljiva x_3 predstavljati neiskorišćeni kapacitet radnog mesta RM_1. Na sličan način određujemo i značenje promenljivih x_4 i x_5. Ukoliko drugo ograničenje označava utrošak sirovine S_1, koja je potrebna za proizvodnju proizvoda P_1, onda će promenljiva x_4 predstavljati neiskorišćeni kapacitet sirivine S_1. Ako je treće ograničenje posledica utroška sirovine S_2, potrebne za proizvodnju proizvoda P_2, onda promenljiva x_5 predstavlja neikorišćeni kapacitet sirovine S_2.

2.4.2.2. *Transformacija nejednačina tipa* $\geq$

Ukoliko je dat hipotetički primer ograničenja $4x_1 + 5x_2 \geq 20$, uočava se da je desna strana nejednačine manja ili jednaka levoj strani. Zbog toga je potrebno dodati jednu dopunsku promenljivu sa desne strane, koja je nenegativna, kako bi dopunili nejednačinu i preveli je u jednačinu. Desnoj strani se dodaje promenljiva x_3, i dobija se jednačina: $4x_1 + 5x_2 = 20 + x_3$.

Sređivanjem jednačine, tako da se leve strane budu promenljive sa svojim koeficijentima a sa desne slobodni član, dopunska promenljiva x_3 dobija negativni koeficijent (-1). Predhodna jednačina dobija oblik: $4x_1 + 5x_2 - x_3 = 20$.

Koeficijent uz dopunsku promenljuvu x_3 je negativan, zbog toga nije moguće da se odredi početno nenegativno bazno rešenje u početnom koraku simpleks algoritma. Zato se uvodi nova promenljiva x_4 u model koja se naziva ***veštačka promenljiva***. Ove promenljive ne pripadaju sistemu ograničenja i nemaju konkretno ekonomsko značenje. Veštačke promenljive služe kao kalkulativno sredstvo. U postupku rešavanja problema one se eliminišu iz rešenja, tako da se u optimalnom rešenju nemože pojaviti ni jedna veštačka promenljiva. Ukoliko se u nenegativno bazno rešenje problema pojavi bar jedna veštačka promenljiva, koje se ne možemo osloboditi, onda ne postoji dopustivo rešenje problema.

Levoj strani jednačine dodaje se veštačka promenljiva x_4 sa pozitivnim (+1) koeficijentom, tako da sada jednačina ima oblik: $4x_1 + 5x_2 - x_3 + x_4 = 20$.

Promenljive x_1 i x_2 potiču iz osnovnog problema, x_3 je dopunska, a x_4 veštačka promenljiva.

Primer 2.4.2. Date nejednačine transformisati u jednačine.

$$\begin{aligned} 2x_1 + 3x_2 &\geq 100 \\ 3x_1 \quad\quad &\geq 200 \\ x_2 &\geq 50 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Rešenje. Leve strane nejednačina su manje ili jednake desnim stranama. Dodavanjem dopunskih promenljivih levim stranama nejednačina dobija se sledeći sistem jednačina:

$$\begin{aligned} 2x_1 + 3x_2 - x_3 \quad\quad\quad + x_6 \quad\quad\quad &= 100 \\ 3x_1 \quad\quad - x_4 \quad\quad\quad + x_7 \quad &= 200 \\ x_2 \quad\quad - x_5 \quad\quad\quad + x_8 &= 50 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 &\geq 0 \end{aligned}$$

U sažetom obliku sistem jednačina transformisan iz nejednačina tipa $\geq$ u jednačine prikazan je relacijom (2.4.20.).

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \quad \Rightarrow \quad \sum_{j=1}^{n} a_{ij}x_j - x_{n+i} + x_{n+i+i} = b_i \qquad (2.4.20.)$$
$$i = 1, 2, \ldots, m$$

2.4.2.3. Transformacija funkcije cilja

Pored transformacije nejednačina neophodno je izvršiti i proširenje funkcije cilja. Polazeći od konstatacije da dopunske (izravnavajuće) promenljive označavaju neiskorišćene kapacitete, logično je da uz njih u funkciji cilja stoje koeficijenti jednaki nuli (0). Kako se veštačke promenljive ne bi pojavljivale u optimalnom rešenju, jer su nepoželjne (nemaju ekonomskog značenja), u funkciji cilja uz ove promenljive uvodimo koeficijente obeležene sa M, gde je M proizvoljno veliki broj. Kada se zahteva maksimalna vrednost funkcije cilja uz veštačke promenljive se uvode koeficijenti $-M$, tj. beskonačno mali broj ($M<<0$), a kada se izračunava minimalna vrednost ciljne funkcije onda se uz veštačke promenljive uvode koeficijenti $+M$, tj. beskonačno veliki broj ($M>>0$).

Pravila za prilagođavanje modela, transformacija nejednačina u jednačine i proširenje ciljne funkcije, uvođenjem dopunskih i veštačkih promenljivih, kako za problem maksimuma tako i za problem minimuma, mogu se pregledno prikazati na tabelaran način, u tabeli 2.4.1.

Tabela 2.4.1. *Pravila za izbor dopunskih i veštačkih promenljivih*

Tabela za izbor dopunskih i veštačkih promenljivih			***Tip ograničenja***		
			$\leq$	=	$\geq$
Kod pretvaranje nejednačina u jednačine	dopunska		DA	NE	DA
	veštačka		NE	DA	DA
Koeficijenti u jednačinama ograničenja	uz dopunsku		+1	/	–1
	uz veštačku		/	+1	+1
Koeficijenti u funkciji cilja	uz dopunsku		0	/	0
	uz veštačku	max	/	$-M$	$-M$
		min	/	$+M$	$+M$

Proširena funkcija cilja čija se maksimalna vrednost traži, sa svim ograničenjima tipa $\leq$, u sažetom obliku, prikazana je relacijom (2.4.21.).

$$F(x)=\sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j + 0\cdot(x_{n+1}+x_{n+1}+\ldots+x_{n+m}) \qquad (2.4.21.)$$

Primer 2.4.3. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{aligned} & \max F(x) = 10x_1 + 20x_2 \\ \text{p.o.}\quad & 3x_1 + 2x_2 \leq 300 \\ & 5x_1 + 5x_2 \leq 200 \\ & \quad\;\; x_2 \leq 100 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje:

$$\begin{aligned} & \max F(x) = 10x_1 + 20x_2 + 0x_3 + 0x_4 + 0x_5 \\ & 3x_1 + 2x_2 + x_3 = 300 \\ & 5x_1 + 5x_2 + x_4 = 200 \\ & \quad\;\; x_2 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Proširena funkcije cilja čija se maksimalna vrednost traži sa svim ograničenjima tipa =, u sažetom obliku, prikazana je relacijom (2.4.22.).

$$F(x)=\sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j - M\cdot(x_{n+1}+x_{n+1}+\ldots+x_{n+m}) \qquad (2.4.22.)$$

Primer 2.4.4. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{aligned} & \max F(x) = 10x_1 + 20x_2 \\ \text{p.o.}\quad & 3x_1 + 2x_2 = 300 \\ & 5x_1 + 5x_2 = 200 \\ & \quad\;\; x_2 = 100 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje:

$$\begin{aligned} & \max F(x) = 10x_1 + 20x_2 - M\,(x_3 + x_4 + x_5) \\ & 3x_1 + 2x_2 + x_3 = 300 \\ & 5x_1 + 5x_2 + x_4 = 200 \\ & \quad\;\; x_2 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Proširena funkcije cilja čija se maksimalna vrednost traži sa svim ograničenjima tipa $\geq$, u sažetom obliku, prikazana je relacijom (2.4.23.).

$$F(x)=\sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j - 0\cdot(x_{n+1}+...+x_{n+m}) - M\cdot(x_{n+m+1}+...+x_{n+m+m}) \quad (2.4.23.)$$

Primer 2.4.5. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{array}{ll} & \max F(x) = 10x_1 + 20x_2 \\ \text{p.o.} & 3x_1 + 2x_2 \geq 300 \\ & 5x_1 + 5x_2 \geq 200 \\ & \quad\quad\; x_2 \geq 100 \\ & x_1,\, x_2 \geq 0 \end{array}$$

Rešenje:

$$\begin{array}{l} \max F(x) = 10x_1 + 20x_2 - 0(x_3 + x_4 + x_5) - M\,(x_6 + x_7 + x_8) \\ 3x_1 + 2x_2 - x_3 + x_6 = 300 \\ 5x_1 + 5x_2 - x_4 + x_7 = 200 \\ x_2 - x_5 + x_8 = 100 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \geq 0 \end{array}$$

Proširena funkcije cilja čija se maksimalna vrednost traži sa mešovitim ograničenjima (sva tri tipa $\leq,=,\geq$), u sažetom obliku, prikazana je relacijom (2.4.24.).

$$F(x)=\sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j + 0\cdot(x_{n+1}+...+x_{n+k}) - M\cdot(x_{n+k+1}+...+x_{n+k+p}) \quad (2.4.24.)$$

gde je

- k – broj dopunskih promenljivih,
- p – broj veštačkih promenljivih.

Primer 2.4.6. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{array}{ll} & \max F(x) = 10x_1 + 20x_2 \\ \text{p.o.} & 3x_1 + 2x_2 \geq 300 \\ & 5x_1 + 5x_2 = 200 \\ & \quad\quad\; x_2 \leq 100 \\ & x_1,\, x_2 \geq 0 \end{array}$$

Rešenje:

$$\begin{array}{l} \max F(x) = 10x_1 + 20x_2 + 0(-x_3 + x_4) - M\,(x_5 + x_6) \\ 3x_1 + 2x_2 - x_3 + x_5 = 300 \\ 5x_1 + 5x_2 + x_6 = 200 \\ x_2 + x_4 = 100 \\ x_1, x_2, x_3, x_4, x_5, x_6, \geq 0 \end{array}$$

Proširena funkcije cilja čija se minimalna vrednost traži sa svim ograničenjima tipa $\leq$, u sažetom obliku, prikazana je relacijom (2.4.25.).

$$F(x)=\sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j + 0\cdot(x_{n+1}+x_{n+1}+...+x_{n+m}) \quad (2.4.25.)$$

Primer 2.4.7. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{array}{ll} & \min F(x) = 10x_1 + 20x_2 \\ \text{p.o.} & 3x_1 + 2x_2 \leq 300 \\ & 5x_1 + 5x_2 \leq 200 \end{array}$$

$$\begin{aligned} & x_2 \leq 100 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje:

$$\begin{aligned} & \min F(x) = 10x_1 + 20x_2 + 0(x_3 + x_4 + x_5) \\ & 3x_1 + 2x_2 + x_3 = 300 \\ & 5x_1 + 5x_2 + x_4 = 200 \\ & x_2 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Proširena funkcije cilja čija se minimalna vrednost traži sa svim ograničenjima tipa =, u sažetom obliku, prikazana je relacijom (2.4.26.).

$$F(x) = \sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j + M \cdot (x_{n+1} + x_{n+1} + \ldots + x_{n+m}) \qquad (2.4.26.)$$

Primer 2.4.8. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{aligned} & \min F(x) = 10x_1 + 20x_2 \\ \text{p.o.}\quad & 3x_1 + 2x_2 = 300 \\ & 5x_1 + 5x_2 = 200 \\ & x_2 = 100 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje:

$$\begin{aligned} & \min F(x) = 10x_1 + 20x_2 + M(x_3 + x_4 + x_5) \\ & 3x_1 + 2x_2 + x_3 = 300 \\ & 5x_1 + 5x_2 + x_4 = 200 \\ & x_2 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Proširena funkcije cilja čija se minimalna vrednost traži sa svim ograničenjima tipa ≥, u sažetom obliku, prikazana je relacijom (2.4.27.).

$$F(x) = \sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j - 0 \cdot (x_{n+1} + \ldots + x_{n+m}) + M \cdot (x_{n+m+1} + \ldots + x_{n+m+m}) \qquad (2.4.27.)$$

Primer 2.4.9. Prilagoditi matematički model za primenu simpleks algoritma.

$$\begin{aligned} & \min F(x) = 10x_1 + 20x_2 \\ \text{p.o.}\quad & 3x_1 + 2x_2 \geq 300 \\ & 5x_1 + 5x_2 \geq 200 \\ & x_2 \geq 100 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje:

$$\begin{aligned} & \min F(x) = 10x_1 + 20x_2 - 0(x_3 + x_4 + x_5) + M(x_6 + x_7 + x_8) \\ & 3x_1 + 2x_2 - x_3 + x_6 = 300 \\ & 5x_1 + 5x_2 - x_4 + x_7 = 200 \\ & x_2 - x_5 + x_8 = 100 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \geq 0 \end{aligned}$$

Proširena funkcije cilja čija se minimalna vrednost traži sa mešovitim ograničenjima (sva tri tipa ≤,=,≥), u sažetom obliku, prikazana je relacijom (2.4.28.).

$$F(x) = \sum_{j=1}^{n} c_j x_j \Rightarrow \sum_{j=1}^{n} c_j x_j + 0 \cdot (x_{n+1} + \ldots + x_{n+k}) + M \cdot (x_{n+k+1} + \ldots + x_{n+k+p}) \qquad (2.4.28.)$$

gde je

- k – broj dopunskih promenljivih,
- p – broj veštačkih promenljivih.

Primer 2.4.10. Prilagoditi matematički model za primenu simpleks algoritma.

$$\min F(x) = 10x_1 + 20x_2$$

$$\text{p.o.}\quad \begin{aligned} 3x_1 + 2x_2 &\geq 300 \\ 5x_1 + 5x_2 &= 200 \\ x_2 &\leq 100 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Rešenje:

$$\min F(x) = 10x_1 + 20x_2 + 0(-x_3 + x_4) + M(x_5 + x_6)$$

$$\begin{aligned} 3x_1 + 2x_2 - x_3 \quad + x_5 \quad &= 300 \\ 5x_1 + 5x_2 \quad + x_6 &= 200 \\ x_2 \quad + x_4 \quad &= 100 \\ x_1, x_2, x_3, x_4, x_5, x_6, &\geq 0 \end{aligned}$$

2.4.3. Kanonični oblik problema linearnog programiranja

Osnovne osobine kanoničnog oblika linearnog programiranja su:

- funkcija cilja se maksimizira ili minimizira,
- sva ograničenja izražena su kao jednačine,
- sve nepoznate promenljive su nenegativne, i
- slobodni članovi sa desne strane jednakosti svakog ograničenja su nenegativni.

Kanonični oblik zahteva ograničenja u obliku jednačina, zbog toga se nejednačine moraju transformisati u jednačine, i proširenu funkciju cilja. Nakon prilagođavanja matematičkog modela, opšti problema maksimuma prikazan relacijama (2.4.7.) – (2.4.9.), u kanoničkom obliku je prikazan relacijama (2.4.29.) – (2.4.31.).

$$\max F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n + 0x_{n+1} + 0x_{n+2} + \ldots + 0x_{n+m} \qquad (2.4.29.)$$

p.o.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} \qquad\qquad &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \qquad + x_{n+2} \qquad &= b_2 \\ \ldots \qquad\qquad & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \qquad\qquad + x_{n+m} &= b_m \end{aligned} \qquad (2.4.30.)$$

$$x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_{n+m} \geq 0 \qquad (2.4.31.)$$

U sažetom obliku, kanonični oblik funkcije cilja prikazan je relacijom (2.4.32.), ograničenja sistema jednačina relacijom (2.4.33.), a prirodna ograničenja (2.4.34.).

$$\max F(x) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} c_{n+i} x_{n+i} \qquad (2.4.32.)$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i \quad i = 1, 2, \ldots, m \tag{2.4.33.}$$

$$x_j \geq 0 \qquad j = 1, 2, ..., n, n+1, ..., n+m \tag{2.4.34.}$$

gde su:

- x_{n+i} (i = 1, 2, ..., m) dopunske promenljive,
- koeficijenti u funkciji cilja c_{n+i} (i = 1, 2, ..., m), koji odgovaraju dopunskim promenljivima, jednaki nuli ($c_{n+i} = 0$).

Postavljeni linearni model odnosi se na probleme u kome je potrebno odrediti maksimalnu vrednost funkcije cilja. Postupak za prikazivanje u kanoničnom obliku ostalih varijanti problema linearnog programiranja (max/min funkcije cilja u kombinaciji sa svim tipovima ograničenja) potpuno je analogan prikazanom. Potrebno je poštovati pravila za izbor dopunskih i veštačkih promenljivih i proširenja funkcije cilja, kao što je prikazano u predhodnom izlaganju.

2.4.4. Standardni oblik problema linearnog programiranja

Osnovne osobine standardnog oblika problema linearnog programiranja su:

- funkcija cilja se maksimizira, tj. max $F(x)$,
- sva ograničenja izražena su kao jednačine (tip ograničenja =),
- sve nepoznate promenljive su nenegativne, tj. $x_j \geq 0$, j=1,2,...,n,
- svi slobodni članovi b_i, i=1,2,...,m, sa desne strane jednakosti svakog ograničenja, su nenegativni, tj $b_i \geq 0$.
- ukupan broj ograničenja m je manji od broja promenljivih

Standardni oblik problema zahteva ograničenja u obliku jednačina, i proširenu maksimiziranu funkciju cilja. Zato se nejednačine, uslovljene ograničenjima, i funkcija cilja moraju transformisati korišćenjem pravila za transformaciju, koja su opisana u poglavlju 2.4.2. Nakon prilagođavanja matematičkog modela, standardni oblik problema, definisan relacijama (2.4.7.) – (2.4.9.), prikazan je relacijama (2.4.35.) – (2.4.37.).

$$\max F(x) = c_1x_1 + ... + c_nx_n + 0\cdot(x_{n+1} + ... + x_{n+k}) - M\cdot(x_{n+k+1} + ... + x_{n+k+p}) \tag{2.4.35.}$$

p.o.

$$\begin{array}{llllll}
a_{11}x_1 + ... + a_{1n}x_n & + x_{n+1} & & + x_{n+k+1} & & = b_1 \\
a_{21}x_1 + ... + a_{2n}x_n & & + x_{n+2} & & + x_{n+k+2} & = b_2 \\
\cdots & & & & & \\
a_{m1}x_1 + ... + a_{mn}x_n & & + x_{n+k} & & + x_{n+k+p} & = b_m
\end{array} \tag{2.4.36.}$$

$$x_1, x_2, ..., x_n, x_{n+1}, ..., x_{n+k}, x_{n+k+1}, ..., x_{n+k+p} \geq 0 \tag{2.4.37.}$$

U sažetom obliku, standardni oblik prikazan je relacijama (2.4.38.) – (2.4.40.).

$$\max F(x) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{k} c_{n+i} x_{n+i} - \sum_{l=1}^{p} c_{n+i+l} \cdot x_{n+i+l} \qquad (2.4.38.)$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} + x_{n+i+l} = b_i,\ i = 1,\ldots,k,\ l = 1,\ldots,p \qquad (2.4.39.)$$

$$x_j \geq 0 \qquad j=1,2,...,n,...,n+k,...,n+k+p \qquad (2.4.40.)$$

gde su:

- k – broj dopunskih promenljivih
- p – broj veštačkih promenljivih
- x_{n+i} (i = 1, 2, ..., k) dopunske promenljive,
- x_{n+i+l} (l = 1, 2, ..., p) veštačke promenljive,
- koeficijenti u funkciji cilja c_{n+i} (i=1,2,...,k), koji odgovaraju dopunskim promenljivima, jednaki su nuli ($c_{n+i} = 0$),
- koeficijenti u funkciji cilja c_{n+i+l} (i=1,2,...,k; l=1,2,...,p), koji odgovaraju veštačkim promenljivima, jednak je velikom broju M, tj. M>>0 ($c_{n+i+l} = M$).

Primer 2.4.11. Dati matematički model svesti na standardni oblik problema LP.

$$\begin{array}{ll} & \min F(x) = -5x_1 + 4x_2 - 3x_3 \\ \text{p.o.} & 2x_1 + 4x_2 - 5x_3 = 10 \\ & 3x_1 + 6x_2 - 2x_3 \leq 20 \\ & -3x_1 + 7x_2 - 5x_3 \leq -15 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Rešenje: Problem minimizacije funkcije cilja se može svesti na problem maksimizacije, i to – min $F(x)$ = max $F(-x)$, na osnovu relacije (3.3), tako da je:

$$\max F(x) = 5x_1 - 4x_2 + 3x_3$$

U trećem ograničenju, slobodni član b_3 je negativan, $b_3 = -15$. Množenjem ovog ograničenja sa – 1 slobodni član postaje pozitivan, a ograničenje izgleda:

$$3x_1 - 7x_2 + 5x_3 \geq 15$$

Ovako transformisan problem dobija oblik:

$$\begin{array}{ll} & \max F(x) = 5x_1 - 4x_2 + 3x_3 \\ \text{p.o.} & 2x_1 + 4x_2 - 5x_3 = 10 \\ & 3x_1 + 6x_2 - 2x_3 \leq 20 \\ & 3x_1 - 7x_2 + 5x_3 \geq 15 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Uvođenjem dopunskih i veštačkih promenljivih, na način opisan u poglavlju 4.2, problem se svodi na standardni oblik sa m = 3 ograničenja i n = 7 promenljivih, kao:

$$\begin{array}{l} \max F(x) = 5x_1 - 4x_2 + 3x_3 + 0(-x_4 + x_5) - M(x_6 + x_7) \\ 2x_1 + 4x_2 - 5x_3 + x_6 = 10 \\ 3x_1 + 6x_2 - 2x_3 + x_5 = 20 \\ 3x_1 - 7x_2 + 5x_3 - x_4 + x_7 = 15 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \end{array}$$

2.4.5. Vektorsko – matrični oblik problema linearnog programiranja

Prilagođeni model linearnog programiranja koji je prikazan relacijama (2.4.29.) – (2.4.31.) može se iskazati u vektorskom i matričnom obliku.

Funkcija cilja u matričnoj formi izgleda:

$$\max \quad F(x) = \begin{bmatrix} c_1 & c_2 & \cdots & c_n & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} \tag{2.4.35.}$$

Sistem jednačina ograničenja je:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ a_{31} & a_{32} & \cdots & a_{3n} & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_m \end{bmatrix} \tag{2.4.36.}$$

Uslov nenegativnosti promenljivih:

$$x_j \geq 0, \qquad j = (1, 2, ..., n+m) \tag{2.4.37.}$$

Zbog potrebe izražavanja problema u matričnom obliku, uvode se sledeće oznake za vektore i matrice:

$$C^T = \begin{bmatrix} c_1 & c_2 & \cdots & c_n & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ \vdots \\ x_{n+m} \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ a_{31} & a_{32} & \cdots & a_{3n} & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix}$$

Problem linearnog programiranja se može izraziti u matričnom obliku na sledeći način: Potrebno je odrediti nenegativni vektor X tako da funkcija cilja dostigne svoju maksimalnu vrednost, tj.

$$\max F(x) = C^T \cdot X \tag{2.4.38.}$$

uz linearna ograničenja:

$$A \cdot X = b \tag{2.4.39.}$$

$$X \geq 0 \tag{2.4.40.}$$

Uočava se da je C^T po formi vektor red $n+m$-tog reda, X vektor kolone, takođe, $n+m$-tog reda, b vektor kolone m-tog reda, A matrica koeficijenata sistema ograničenja reda ($m \times n+m$).

Često je u upotrebi još jedan oblik izražavanja linearnih problema. Kolone matrice A mogu se posmatrati kao vektori kolona m-tog reda, a označavaju se kao:

$$A_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}, A_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}, \cdots, A_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}, A_{n+1} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, A_{n+2} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, A_{n+m} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{2.4.41.}$$

Sistem linearnih jednačina ograničenja se sada predstavlja kao:

$$A_1 \cdot x_1 + A_2 \cdot x_2 + \ldots + A_n \cdot x_n + A_{n+1} \cdot x_{n+1} + \ldots + A_{n+m} \cdot x_{n+m} = b \tag{2.4.42.}$$

ili u sažetom obliku:

$$\sum_{j=1}^{n+m} A_j x_j = b \tag{2.4.43.}$$

2.4.6. Zavisne i nezavisne promenljive

Sistem jednačina $Ax = b$ ima rešenje ako je rang matrice sistema jednačina

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & & & & & \\ a_{m1} & a_{m2} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix}$$

jednak rangu proširene matrice $[A|b]$, koja se dobija dodavanjem kolone slobodnih članova b_1, b_2, ..., b_m matrici A (teorema Kronecker-Capellija). Rang r matrice A naziva se rangom sistema i predstavlja broj linearno nezavisnih jednačina među datim ograničenjima. Ako taj uslov nije ispunjen, skup dopustivih rešenja Ω_P je prazan i problem nema rešenja.

Rang sistema jednačina $Ax = b$ ne može biti veći od broja jednačina m, tj. uvek je ispunjeno $r \leq m$, ali broj r ne može biti veći ni od broja promenljivih n, pa je takođe ispunjeno $r < n$. Ukoliko je $r < m$, tada primenom Gaussovog algoritma dolazimo do zaključka o nesaglasnosti sistema ili eliminišemo $m - r$ suvišnih jednačina u sistemu $Ax = b$. Zato pretpostavimo da je $r = m$, tj. da među ograničenjima (2.4.39) nema linearno zavisnih i da je sistem $Ax = b$ saglasan.

U slučaju da je $r = m = n$, sistem ima jedinstveno rešenje $x = A^{-1}b$ pa ostaje samo da se proveri uslov $x > 0$.

Interesantan slučaj je $r = m < n$, kada je broj linearno nezavisnih jednačina manji od broja promenljivih. Tada, ukoliko je sistem jednačina $Ax = b$ saglasan, postoji beskonačno mnogo rešenja. Svako od tih rešenja dobija se tako što se za $n - r = s$ promenljivih izaberu proizvoljne vrednosti, a zatim se vrednosti preostalih r promenljivih izračunavaju iz sistema jednačina $Ax = b$. Promenljive veličine koje izračunavamo nazivamo *zavisnim* ili *baznim* (ima ih r), a promenljive veličine čije se vrednosti biraju proizvoljno nazivamo *nezavisnim* ili *slobodnim* promenljivim (njih ima $n - r = s$). Praktično se iz sistema jednačina r zavisnih promenljivih izražava pomoću $n - r = s$ nezavisnih promenljivih, pa se za nezavisne promenljive biraju proizvoljne vrednosti. Kako se za nezavisne promenljive može birati beskonačno mnogo različitih vrednosti, to sistem jednačina $Ax = b$ ima beskonačno mnogo rešenja.

Definicija 2.4.1 *Nenegativno rešenje, koje se dobija tako što se za nezavisne (slobodne) promenljive izaberu vrednosti jednake nuli, naziva se bazično rešenje zadatka linearnog programiranja.*

Primer 2.4.12. Transformisati dati, opšti, oblik zadatka linearnog programiranja u standardni oblik, naći broj zavisnih i nezavisnih promenljivih, izraziti zavisne promenljive pomoću nezavisnih i naći jedno bazično rešenje.

$$\max F(x) = 2x_1 + 5x_2$$
$$\text{p.o.}\quad \begin{aligned} x_1 + 4x_2 &\le 24 \\ 2x_1 + x_2 &\le 21 \\ x_1 + x_2 &\le 9 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Rešenje. Levim stranama nejednačina dodajemo nenegativne promenljive x_3, x_4 i x_5 tako da one postanu jednačine:

$$\begin{aligned} x_1 + 4x_2 + x_3 \qquad\qquad &= 24 \\ 2x_1 + x_2 \qquad + x_4 \qquad &= 21 \\ x_1 + x_2 \qquad\qquad + x_5 &= 9 \end{aligned}$$

Sada se zadatak linearnog programiranja (u kanoničnom obliku sa maksimizacijom funkcije cilja) može formulisati na sledeći način: Naći nenegativno rešenje $x = (x_1, x_2, x_3, x_4$ i $x_5)$, $x_i \ge 0$, $i = 1,2,3,4,5$ koje zadovoljava sistem ograničenja u kanoničnom obliku i za koje ciljna funkcija $F(x) = 2x_1 + 5x_2 + 0x_3 + 0x_4 + 0x_5$ dostiže maksimum. Matrica sistema jedačina A i proširena matrica A_p izgledaju:

$$A = \begin{bmatrix} 1 & 4 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad A_p = \begin{bmatrix} 1 & 4 & 1 & 0 & 0 & 24 \\ 2 & 1 & 0 & 1 & 0 & 21 \\ 1 & 1 & 0 & 0 & 1 & 9 \end{bmatrix}$$

Obzirom da je rang matrice A jednak rangu matrice A_p ($r = r_A = r_{Ap} = 3$), tada u sistemu ima r=3 zavisne promenljive i $s = n-r = 5-3 = 2$ nezavisne promenljive. Za nezavisne promenljive izaberimo x_1, x_2 i pomoću njih izrazimo zavisne promenljive x_3, x_4, i x_5:

$$\begin{aligned} x_3 &= 24 - x_1 - 4x_2 \\ x_4 &= 21 - 2x_1 - x_2 \\ x_5 &= 9 - x_1 - x_2 \end{aligned}$$

Bazično rešenje se dobija ukoliko se proglasi da je $x_1=x_2=0$. Odatle sledi da je: $x_3=24$, $x_4=21$ i $x_5=9$, tj. dobili smo bazično rešenje (0,0,24,21,9). Za ovo bazično rešenje ciljna funkcija ima vrednost $F(x)=0$. Naravno, mogu se dobiti i druga bazična rešenja ukoliko se drugi par promenljivih odabere za nezavisne promenljive. Npr. ukoliko se za nezavisne promenljive odabre x_2 i x_5, onda bi zavisne promenljive bile definisane na sledeći način:

$$x_1 = 9 - x_2 - x_5$$
$$x_3 = 24 - (9 - x_2 - x_5) - 4x_2 = 15 - 3x_2 + x_5$$
$$x_4 = 21 - 2(9 - x_2 - x_5) - x_2 = 3 + x_2 + 2x_5$$

Za $x_2=x_5=0$ dobija se drugo bazično rešenje (9,0,15,3,0). Ukoliko se funkcija cilja izrazi pomoću x_2 i x_5 ona dobija vrednost 18.

$$F(x) = 2(9 - x_2 - x_5) + 5x_2 = 18 + 3x_2 - 2x_5 = 18$$

Ova vrednost je veća od vrednosti funkcije cilja u prvom bazičnom rešenju, tj. bazično rešenje (9,0,15,3,0) je bliže optimalnom rešenju od bazičnog rešenja (0,0,24,21,9).

Maksimalna vrednost funkcije cilja iznosi $maxF(x) = 33$. Ovu vrednost ciljna funkcija je dostigla u bazičnom rešenju (4,5,0,13,0), to bazično rešenje predstavlja i optimalno rešenje.

Primer 2.4.13. Fabrika proizvodi dve vrste artikala A_1 i A_2, i to na mašinama M_1 i M_2. Za artikal A_1 mašina M_1 radi 2^h, a mašina M_2 radi 4^h, i za artikal A_2 mašina M_1 radi 4^h, a mašina M_2 radi 2^h. Mašine su na raspolaganju 24 *h*/*dan*. Fabrika dobija 3.500 dinara po jedinici proizvoda A_1, a 4.800 dinara po jedinici proizvoda A_2. Koliko treba proizvoditi artikala A_1 i A_2 i kako iskoristiti rad mašina M_1 i M_2 da dnevna dobit fabrike bude maksimalna?

Rešenje: Ukoliko je x_1 broj proizvedenih artikala A_1, a x_2 broj proizvedenih artikala A_2 u toku dana, tada je dnevna dobit fabrike:

$$\begin{array}{ll} & F(x) = 3.500x_1 + 4.800x_2 \\ \text{p.o.} & 2x_1 + 4x_2 \le 24 \\ & 4x_1 + 2x_2 \le 24 \\ & x_1, x_2 \ge 0 \end{array}$$

Dobijen je simetrični oblik matematičkog modela. Ukoliko se model proširi uvođenjem dopunskih (slack) promenljivih x_3 i x_4 dobija se ekvivalentan problem u standardnom obliku:

$$\begin{array}{ll} & F(x) = 3.500x_1 + 4.800x_2 + 0(x_3 + x_4) \\ \text{p.o.} & 2x_1 + 4x_2 + x_3 \qquad = 24 \\ & 4x_1 + 2x_2 \qquad + x_4 = 24 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{array}$$

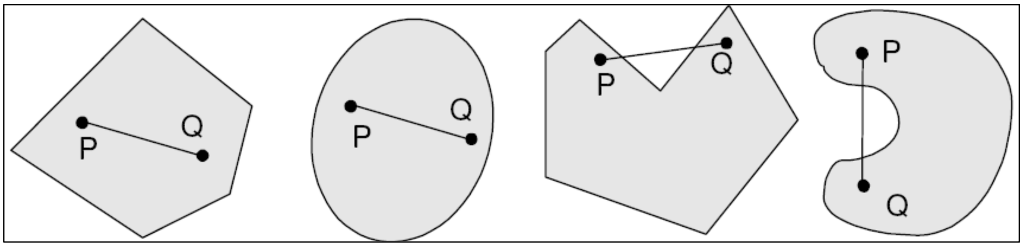
ili u matričnoj formi:

$$A = \begin{bmatrix} 2 & 4 & 1 & 0 \\ 4 & 2 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 24 \\ 24 \end{bmatrix} \quad C^T = \begin{bmatrix} 3500 & 4800 & 0 & 0 \end{bmatrix}$$

2.5. KONVEKSNOST SKUPOVA

Dopustiva (moguća) rešenja linearnog programiranja se zasnivaju na nekim osobinama konveksnih skupova. Za geometrijski skup K se kaže da je konveksan ukoliko sadrži svaku konveksnu kombinaciju svojih elemenata, tj., ukoliko sadrži za bilo koje svoje dve tačke (a^1 i $a^2 \in K$)i čitavu duž među njima (slika 2.5.1.). Zbog toga je moguće povezivanje linearnog programiranja i teorije konveksnih skupova. Linearna ograničenja (2.4.39.) i (2.4.40.) određuju konveksni skup tačaka u m

dimenzionalnom vektorskom prostoru. Zadatak je da se od svih tačaka konveksnog skupa odabere ona koja će obezdediti da funkcija cilja (2.4.38.) dostigne svoju ekstremnu vrednost (minimum/maksimum). Zadatak je moguće rešiti jer, među mogućim rešenjima linearnog programiranja posebno su značajna ona koja odgovaraju ekstremnim tačkama konveksnog skupa. Optimalno rešenje problema se nalazi u nekoj ekstremnoj tački ili svakoj konveksnoj kombinaciji više ekstremnih tačaka.



a) *konveksni skupovi* *b*) *nekonveksni skupovi*

Slika 2.5.1. *Oblast dopustivih rešenja u obliku poliedra sa ciljnom finkcijom paralelnom sa jednom stranom poliedra, gde funkcija cilja dostiže ekstremnu vrednost na toj, paralelnoj, strani poliedra*

Najmanji konveksan skup koji u prostoru R^n sadrži $n + 1$ različitih tačaka naziva se *simpleks* u R^n. U prostoru R^2 svaki trougao je simpleks, a u R^3 svaki tetraedar je simpleks.

Navedene osobine i tvrdnje, neophodne za izučavanje simpleks metoda kao i metoda unutrašnjih tačaka, dokazuju se pomoću nekoliko definicija i teorema.

Teorema 2.5.1. *Skup svih dopustivih* (*mogućih*) *rešenja linearnog programiranja čini konveksni skup K, tj. skup* $\Omega_P = \{x | Ax = b, x \geq 0\}$ *je konveksan.*

Teorema 2.5.2. Ako je *skup dopustivih rešenja* $\Omega_P = \{x | Ax = b, x \geq 0\} \neq 0$, *tada on sadrži bar jedno osnovno (bazno) rešenje.*

Teorema 2.5.3. *Ako je* $x = (x_1,...x_n) \in \Omega_P$ *bazično rešenje sistema* $Ax = b$, $x > 0$, *tada je x ekstremna tačka skupa* Ω_P. *Obratno, ako je x ekstremna tačka skupa* Ω_P, *tada je x bazično rešenje.*

Teorema 2.5.4. *Funkcija cilja* $F(x) = C \cdot X$ *definisana na konveksnom skupu mogućih rešenja K, postiže svoju ekstremnu vrednost (minimum ili maksimum) u nekoj ekstremnoj tački konveksnog skupa K, ili u svakoj konveksnoj kombinaciji više ekstremnih tačaka skupa K.*

Tačka *X* iz konveksnog skupa *K* zove se ekstemna tačka ukoliko se ne može izraziti kao konveksna kombinacija nekih drugih tačaka iz skupa *K*. Može se reći da ekstremna tačka ne leži između dve tačke konveksnog skupa.

Teorema 2.5.5. *Ukoliko postoji m linearno nezavisnih vektora tako da važi vektorska jednačina* $A_1 \cdot x_1 + A_2 \cdot x_2 + ... + A_{n+m} \cdot x_{n+m} = B$, *onda je odgovarajuće* *bazično rešenje $X = (x_1, x_2, ..., x_n, 0, 0, ..., 0)$ ekstremna tačka konveksnog skupa dopustivih rešenja K.*

Teorema 2.5.6. *Vektori $A_1, A_2, ..., A_m$ koji odgovaraju pozitivnim koordinatama nekog bazno dopustivog rešenja, tj. neke ekstremne tačke konveksnog skupa K, linearno su nezavisni. Iz ovog stava proizilazi da ekstremna tačka može da ima najviše m pozitivnih koordinata.*

2.6. GEOMETRIJSKI METOD

Svakom uslovu nenegativnosti odgovara u vektorskom prostoru R^n poluprostor u kome je odgovarajuća promenljiva nenegativna. Svakoj uslovnoj jednačini u R^n odgovara jedna hiperravan. Svakoj uslovnoj nejednačini odgovara poluprostor omeđen hiperravni pridruženoj odgovarajućoj jednačini. Skup svih dopustivih vektora x je presek svih datih poluprostora i datih hiperravni, dakle čini jedan konveksni poliedar Ω_P. Jednačina $c^Tx=k$ za neko k predstavlja hiperravan paralelnu sa prostorom R^{n-1} koji je normalan na c. Projekcija poliedra Ω_P na pravac određen vektorom c je zatvoren skup $[l, L]$ realnih brojeva, gde je l minimum a L maksimum ciljne funkcije. Odgovarajuće hiperravni normalne na c su dodirne hiperravni poliedra Ω_P. Zajedničke tačke tih dodirnih hiperravni sa poliedrom Ω_P daju vrednosti u kojima funkcija cilja dostiže ekstremnu vrednost.

Geometrijski metod se može iskoristiti kod problema koji sadrže $n=2$, a najviše $n=3$ promenljive. Zadatak linearnog programiranja dat u osnovnom obliku koji ispunjava uslov $n–m=2$ (a najviše $n–m=3$) takođe se može rešavati geometrijskim metodom. Geometrijski metod, iako ne baš pristupačan, koristi se zato što olakšava pristup opštoj algebarskoj metodi.

Neka je dat linearni problem u obliku:

$$\max F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

p.o.

$$\begin{gathered}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\
\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \\
x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0
\end{gathered}$$

Za dati sistem se zna da je svako rešenje sistema nejednačina jedna tačka prostora R^n, a skup nenegativnih dopustivih rešenja Ω_P je podskup prostora R^n. Svaka od nejednačina

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i \quad i = 1, 2, \ldots, m$$

određuje podskup $D_i \subset R^n$, $i=1,..., m$ koji predstavlja skup tačaka s jedne strane hiperravni

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

pa je oblast dopustivih rešenja (poliedar u R^n) određena presekom skupova

$$\Omega_P = D_1 \cap D_2 \cap \cdots \cap D_m \cap D_{m+1} \cap \cdots \cap D_{m+n}$$

gde se podskupovi $D_{m+1},...,D_{m+n}$ dobijaju iz uslova nenegativnosti promenjivih $x_1 \geq 0,...,x_n \geq 0$. Skup dopustivih rešenja geometrijski predstavlja poliedar (simplicijalni kompleks).

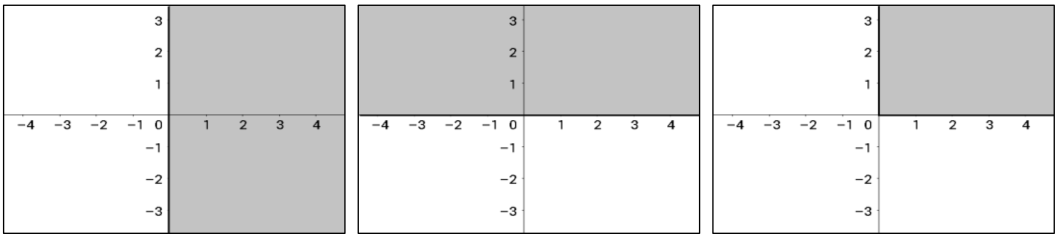
Na sledećem primeru je prikazan postupak rešavanja zadatka linearnog programiranja primenom geometrijskog metoda.

Primer 2.6.1. Potrebno je rešiti problem linearnog programiranja: Odrediti nenegativne vrednosti promenljivih x_1 i x_2 za koje funkcija kriterijuma (cilja) postiže maksimalnu vrednost i da istovremeno zadovoljava zadata ograničenja:

$$\max F(x) = x_1 + 4x_2$$

$$\text{p.o.} \quad \begin{aligned} 2x_1 + 4x_2 &\leq 20 \\ -x_1 + x_2 &\leq 2 \\ x_1 + x_2 &\leq 7 \\ x_1 &\leq 7 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Rešenje: Ovaj problem ima samo dve promenljive, pa se optimalno rešenje može utvrditi grafičkom metodom. Problem se rešava tako se sve nejednačine iz sistema ograničenja predstave grafički u pravouglom koordinatnom sistemu. Pošto promenljive x_1 i x_2 nemogu biti negativne, za grafičko prikazivanje sistema ograničenja se uzima samo prvi kvadrant koordinatnog sistema. Prirodno ograničenje $x_1=0$ je zapravo ordinata x_2, a uslov nenegativnosti $x_1 \geq 0$ desna poluravan koordinatnog sistema, prikazano na slici 2.6.1. *a*). Prirodno ograničenje $x_2=0$ je apcisa x_1, a uslov nenegativnosti $x_2 \geq 0$ gornja poluravan koordinatnog sistema, prikazano na slici 2.6.1. *b*). Uslov nenegativnosti $x_1,x_2 \geq 0$ predstavlja prvi kvadrant koordinatnog sistema, kao što je prikazano na slici 2.6.1. *c*).



a) $x_1 \geq 0$ *b*) $x_2 \geq 0$ *c*) $x_1, x_2 \geq 0$

Slika 2.6.1. *Uslov nenegativnosti*

Svaku linearnu nejednačinu iz sistema ograničenja grafički predstavljamo tako što je transformišemo u odgovarajuću jednačinu. Prvoj nejednačini odgovara sledeća jednačina:

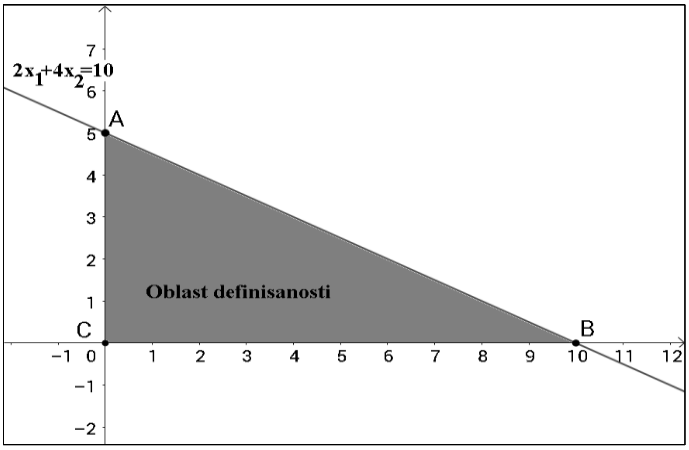
$$2x_1 + 4x_2 = 20$$

Ova jednačina predstavlja jednu pravu. Kako se razmatraju samo nenegativne vrednosti promenljivih x_1 i x_2 to je značajna samo duž ove prave koja se nalazi u prvom kvadrantu pravouglog koordinatnog sistema. Ova duž se najjednostavnije unosi u koordinatni sistem pošto se odrede presečne tačke prave sa koordinatnim osama. Sledi da je:

– za $x_1 = 0 \Rightarrow x_2 = 5$, pa je presečna tačka sa ordinatom tačka A sa koordinatama (0,5),

– za $x_2 = 0 \Rightarrow x_1 = 10$, pa je presečna tačka sa apcisom tačka *B* sa koordinatama (10,0).

Smer nejednačine određujemo tako što proizvoljno odaberemo jednu tačku van ucrtane duži (prave). Ako koordinate odabrane tačke zadovoljavaju datu nejednačinu onda je smer te nejednačine u pravcu te proizvoljno odabrane tačke, u protivnom smer je u suprotnom pravcu. Tako npr. ako uzmemo tačku sa koordinatama (2,2) i zamenimo je u nejednačini ($2 \cdot 2+4 \cdot 2<10$), uočava se da koordinate te tačke zadovoljavaju ograničenje pa je smer ograničenja u pravcu tačke (2,2). Na slici 2.6.2. grafički je predstavljena prva jednačina i određen smer prvog ograničenja. Sve tačke iz osenčanog područja, trougla *ABC*, zadovoljavju prvu nejednačinu i predstavljaju oblast definisanosti *D*.



Slika 2.6.2. *Ucrtavanje prvog ograničenja*

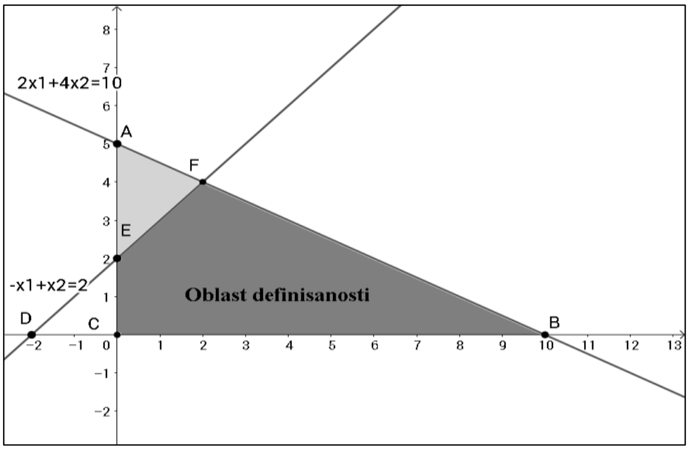
Na isti način se unose i ostala ograničenja. Drugoj nejednačini odgovara jednačina:

$$-x_1 + x_2 = 2$$

za koju su presečne tačke sa koordinatnim osama:

– za $x_1 = 0 \Rightarrow x_2 = 2$, pa je presečna tačka sa ordinatom tačka *D* sa koordinatama (0,2),
– za $x_2 = 0 \Rightarrow x_1 = -2$, pa je presečna tačka sa apcisom tačka *E* sa koordinatama (-2,0).

Smer nejednačine se određujemo na potpuno isti način kao i kod predhodnog ograničenja. Uočava se da je oblast definisanosti *D* smanjena jer je na nju uticalo, osim prvog, i drugo ograničenje. Sada oblast *D* sačinjava četvorougao *BCEF*, kao što je prikazano na slici 2.6.3.



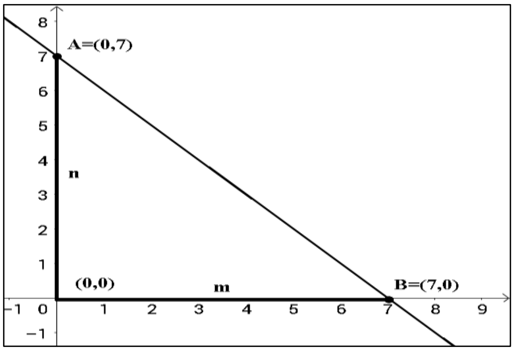
Slika 2.6.3. *Ucrtavanje drugog ograničenja*

Konstrukcija duži (prave) može da se uradi i tako što se konkretna jednačina prave prevede u segmentni oblik:

$$\frac{x_1}{m}+\frac{x_2}{n}=1$$

gde su:
- m – odsečak na apcisi,
- n – odsečak na ordinati,

kao što je prikazano na slici 2.6.4.



***Slika 2.6.4.** Odsečci na koordinatnim osama*

Trećoj nejednačini odgovara jednačina:

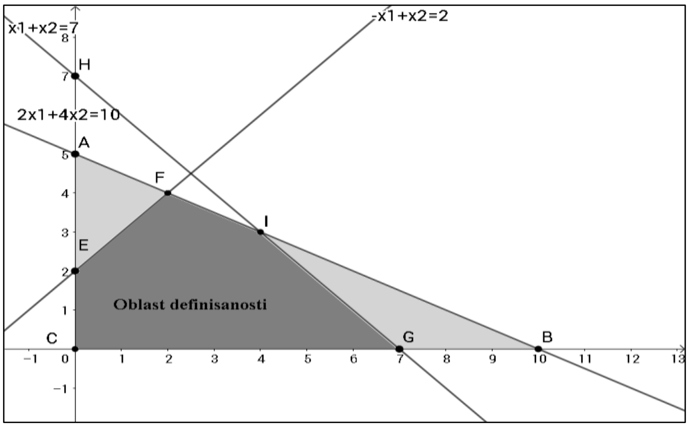
$$x_1 + x_2 = 7$$

prevođenjem u segmentni oblik, jednačina prave izgleda:

$$\frac{x_1}{7}+\frac{x_2}{7}=1$$

- $m = 7 \Rightarrow$ to je odsečak na apcisnoj osi 7, tj. tačka G sa koordinatama (7,0).
- $n = 7 \Rightarrow$ to je odsečak na ordinatnoj osi 7, tj. tačka H sa koordinatama (0,7),

Nakon određivanja smera nejednačini, oblast definisanosti D se opet smanjila jer je na nju uticalo, osim prve dve, i treće ograničenje. Sada oblast D sačinjava petougao *CEFIG*, kao što je prikazano na slici 2.6.5.

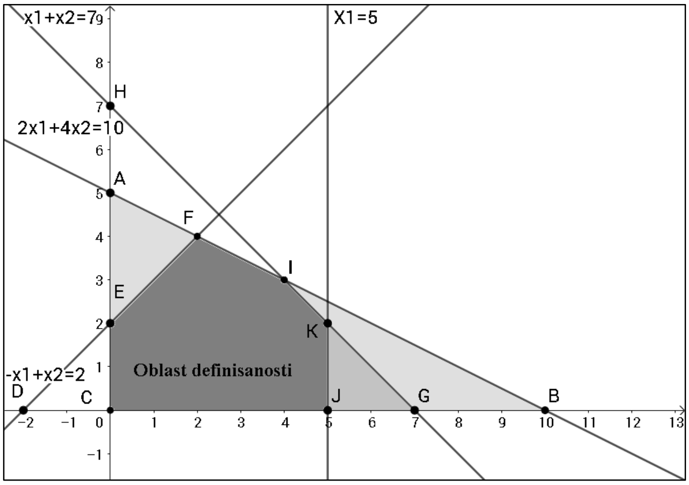


***Slika 2.6.5.** Ucrtavanje trećeg ograničenja*

U modelu postoji i četvrto ograničenje kome odgovara jednačina:

$$x_1 = 5$$

Ova prava je paralelna ordinatnoj osi x_2, i prolazi kroz tačku $J = (5,0)$. Nakon određivanja smera nejednačini, oblast D se opet smanjila jer je na nju uticalo i četvrto ograničenje. Oblast definisanosti D sačinjava šestougao *CEFIKJ*, kao što je prikazano na slici 2.6.6.



Slika 2.6.6. *Ucrtavanje četvrtog ograničenja*

Nakon iscrtavanja svih ograničenja, na slici 2.6.6., obeleženo je (tamno osenčano) područje mogućih rešenja. Zapravo, koordinate svake tačke šrafiranog područja zadovoljavaju sva ograničenja linearnog modela i uslov nenegativnosti. Može se uočiti da se radi o zatvorenom zajedničkom području i da je to područje jedan konveksni skup.

Rešenje problema se sastoji u tome da se od svih mogućih rešenja odabere ono koje će obezbediti da funkcija cilja postigne najveću moguću vrednost. Da bi se grafički odredilo optimalno rešenje mora se uneti i prava koja će reprezentovati funkciju cilja. Funkcija kriterijuma (cilja):

$$F(x) = x_1 + 4x_2$$

predstavlja familiju paralelnih pravih. Raznim vrednostima $F(x)$ odgovaraju razne međusobno paralelne prave. Svejedno je koja će se prava iz familije pravih grafički predstaviti i uzeti za početno razmatranje. Najčešće se koristi jedan od sledeća dva postupka:

1. Iz familije pravih uzima se prava koja prolazi kroz koordinatni početak. To je prava koja predstavlja linearnu jednačinu:

$$F(x) = x_1 + 4x_2 = 0$$

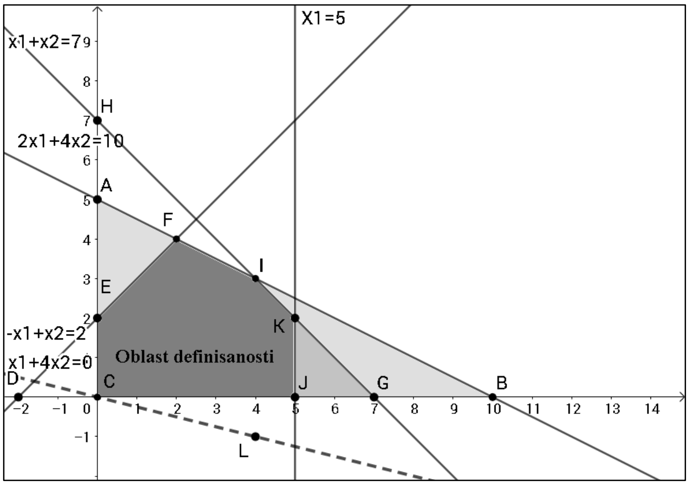
iz koje sledi da je:

$$x_2 = -\frac{1}{4}x_1$$

– za $x_1 = 0 \implies x_2 = 0$, to je tačka C u koordinatnom početku (0,0),

– za $x_1 = 4 \implies x_2 = -1$, to je tačka L sa koordinatama (4,-1).

Prava koja prolazi kroz koordinatani početak u tačci $C = (0,0)$ i tačci $L = (4,-1)$ ucrtana je isprekidanom linijom, i prikazana na slici 2.6.7.

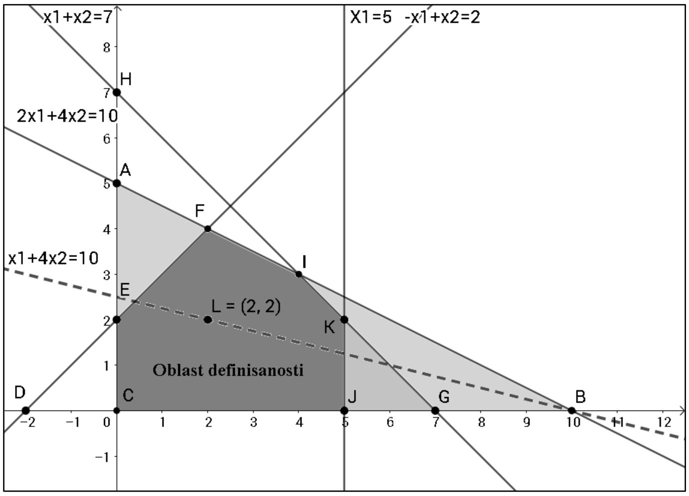


Slika 2.6.7. *Ucrtavanje F(x)=0*

2. Može se, takođe, iz familija pravih uzeti prava koja prolazi kroz tačku iz područja mogućih rešenja. Na primer, tačka L = (2,2) i odredi se vrednost funkcije cilja:

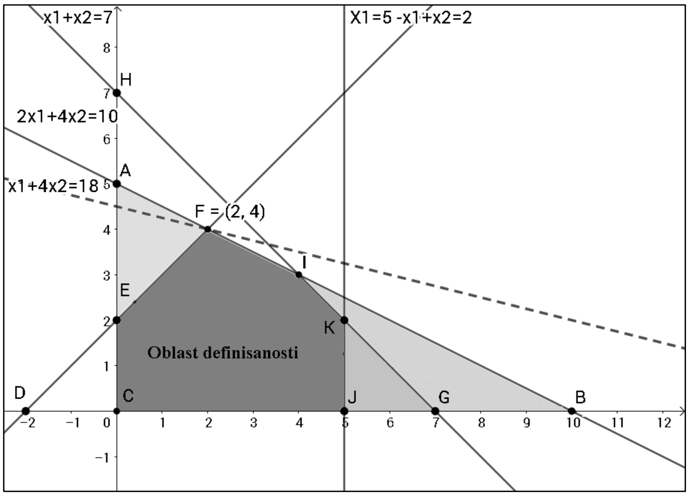
$$F(x) = x_1 + 4x_2 = 1 \cdot 2 + 4 \cdot 2 = 10$$

Iz ove jednačine, za x_2=0 $\Rightarrow$ x_1=10, a za x_1=0 $\Rightarrow$ x_2=2,5, pa su presečne tačke sa apcisom (10,0) i ordinatom (0,2,5). Prava koja prolazi kroz tačku L = (2,2) ucrtana je isprekidanom linijom i prikazana na slici 2.6.8.



Slika 2.6.8. *Ucrtavanje F(x)=10*

Kako prava koja prolazi kroz tačku (2,2) daje veću vrednost funkcije cilja, zaključuje se da što je prava udaljenija od koordinatnog početka dobija se veća vrednost funkcije cilja. Potrebno je odrediti maksimalna vrednost funkcije cilja. Zbog toga se prava, kojom je predstavljena funkcije cilja, pomera paralelno u smeru povećanja njene vrednosti, odnosno što je moguće dalje od koordinatnog početka ali tako da bar u jednoj tačci dodiruje područje mogućih rešenja. Pomeranje se vrši translatorno (paralelno) u odnosu na pravu $F(x)$=0 ili $F(x)$=10 ili bilo koje druge konstruisane početne prave koja definiše funkciju cilja. U ovom primeru, funkcija cilja dostiže svoju maksimalnu vrednost u tačci F, kao što je prikazano na slici 2.6.9.



Slika 2.6.9. *Ucrtavanje optimalnog rešenja maxF(x)=18*

Da bi se dobila vrednost funkcije cilja potrebno je odrediti koordinate tačke *F*, koja je posledica preseka dve prave p_1 i p_2 (prvo i drugo ograničenje):

$$2x_1 + 4x_2 = 20$$
$$-x_1 + \ \ x_2 = 2$$

Rešavanjem sistema dve jednačine sa dve nepoznate dobijaju se koordinate tačke *F*. Sistem jednačina se može izračunati na više načina:

1. Iz druge jednačine se izrazi x_2: $x_2 = x_1 + 2$ i zameni u prvoj, koja se rešava po pormenljivoj x_1:

$$2x_1 + 4(x_1 + 2) = 20$$
$$2x_1 + 4x_1 = 20 - 8$$
$$6x_1 = 12$$
$$x_1 = 2$$

Dobijeno rešenje se vrati u drugu jednačinu i reši po x_2:

$$x_2 = x_1 + 2$$
$$x_2 = 2 + 2 = 4$$

2. Drugi način za izračunavanje koordinate presečne tačke je sabiranjem obe jednačine. Prethodno drugu jednačinu pomnoćimo sa koeficijentom 2.

$$2x_1 + 4x_2 = 20$$
$$\underline{-x_1 + \ \ x_2 = 2 \quad /\cdot 2}$$
$$2x_1 + 4x_2 = 20$$
$$\underline{-2x_1 + 2x_2 = 4 \quad /+}$$
$$6x_2 = 24$$
$$x_2 = 4$$

dobijeno rešenje se vrati u bilo koju jednačinu i reši po promenljivoj x_1:

$$-x_1 + \ \ x_2 = 2$$
$$x_1 = x_2 - 2 = 2$$

3. Sistem jednačina se može rešiti korišćenjem Kramerovih formula:

$$x_1 = \frac{\Delta x_1}{\Delta},\ x_2 = \frac{\Delta x_2}{\Delta}, \cdots,\ x_i = \frac{\Delta x_i}{\Delta}, \quad i = (1,2,\ldots,n)$$

gde su:
- Δ – determinanta sistema jednačina,
- Δx_i – determinanta po *i*-toj promenljivoj.

$$\Delta = \begin{vmatrix} 2 & 4 \\ -1 & 1 \end{vmatrix} = 2\cdot 1 - 4\cdot(-1) = 6$$

$$\Delta x_1 = \begin{vmatrix} 20 & 4 \\ 2 & 1 \end{vmatrix} = 20\cdot 1 - 4\cdot 2 = 12$$

$$\Delta x_2 = \begin{vmatrix} 2 & 20 \\ -1 & 2 \end{vmatrix} = 2\cdot 2 - 20\cdot(-1) = 24$$

$$x_1 = \frac{\Delta x_1}{\Delta} = \frac{12}{6} = 2 \quad x_2 = \frac{\Delta x_2}{\Delta} = \frac{24}{6} = 4$$

Dodirna tačka *F* sa koordinatama (2,4) je ekstremna tačka konveksnog skupa dopustivih (mogućih) rešenja. Tačka *F*=(2,4) zadovoljava sva ograničenja iz matematičkog modela i obezbeđuje ekstremnu (maksimalnu) vrednost funkcije cilja. Zato kažemo da je to optimalno rešenje problema, koje se može prikazati na sledeći način:

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

a maksimalna vrednost funkcije cilja iznosi:

$$\max F(x) = x_1 + 4x_2 = 1\cdot 2 + 4\cdot 4 = 18.$$

Geometrijski metod, za slučaj $n = 2$ je implementiran u programskom jeziku `MATHEMATICA`. Tako je nastao program `GEOM`, koji za unet problem linearnog programiranja u simetričnom obliku od dve promenljive pronalazi optimalno rešenje geometrijskim metodom i pri tome grafički prikazuje sve međukorake. Za grafičko prikazivanje skupa Ω_P dopustivih rešenja, korišćene su standardne funkcije programskog jezika `MATHEMATICA`: `InequalityPlot`, `InequalitySolve`, `FindInstance`, itd. Ove funkcije se nalaze u standardim paketima `Graphics'InequalityGraphics` i `Algebra' InequalitySolve`.

Primer 2.6.2. Rešiti problem linearnog programiranja korišćenjem programa GEOM.

$$\max F(x) = 8x_1 + 12x_2$$

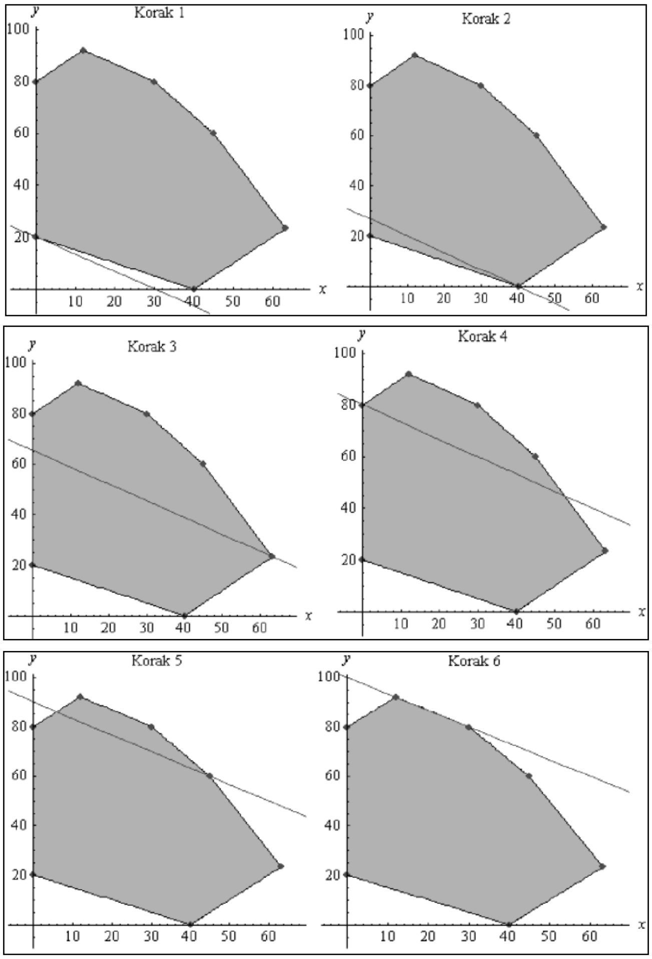
po

$$\begin{aligned} 8x_1 + 4x_2 &\le 600 \\ 2x_1 + 3x_2 &\le 300 \\ 4x_1 + 3x_2 &\le 360 \\ 5x_1 + 10x_2 &\ge 600 \\ x_1 - x_2 &\ge -80 \\ x_1 - x_2 &\le 40 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Rešenje. Problem se rešava sledećom naredbom:

```
Geom[8x+12y,{8x+4y<=600,2x+3y<=300,4x+3y<=360,5x+10y>=600,x-y>=-80,x-
y<=40,x>=0,y>=0}]
```

Program daje grafički prikaz skupa dopustivih rešenja Ω_P, kao i prave koja odgovara ciljnoj funkciji. Pravu pomeramo "naviše", u pravcu vektora c^T=[8,12], sve dok postoji presek sa poligonom dopustivih rešenja. Na slici 2.6.10. su prikazani položaji prave kada prolazi kroz temena poligona, tj kroz ekstremne tačke oblasti dopustivih rešenja. Prikazani su različiti položaji funkcije cilja.



Slika 2.6.10. *Rešenje problema zadatka 2.6.2.*

Program takođe daje optimalno rešenje (ili izraz koji opisuje sva optimalna rešenja). U ovom slučaju to je:

$$x^* = q\begin{bmatrix}12\\92\end{bmatrix} + (1-q)\begin{bmatrix}30\\80\end{bmatrix}, \quad 0 \le q \le 1, \quad \max F(x) = 1.200$$

Grafičko rešavanje problema LP nema posebnog praktičnog značaja jer nema realnih problema koji imaju samo dve promenljive. Međutim, ova metoda omogućava da se na očigledan način prikaže suština linearnog programiranja i neke karakteristike njegovih rešenja. Na osnovu tih rešenja moguće je izvesti zaključke o problemima linearnog programiranja, a to su:

I. Problem linearnog programiranja se formuliše kao problem u kome se treži minimalna ili maksimalna vrednost funkcije cilja. Oblik i struktura matematičkih modela se bitnije na razlikuje.

II. Skup ograničavajućih faktora, izražen sistemom ograničenja, formira zajedničko područje, tj. oblast dopustivih rešenja *D*.
III. Oblast dopustivih rešenja je konveksan skup.
IV. Ukoliko je oblast dopustivih rešenja *D* konačna onda na njenoj granici postoji najmanje jedna tačka koja predstavlja optimalno rešenje.
V. Kada je skup dopustivih rešenja D prazan onda ne postoji optimalno rešenje linearnog problema. Kod realnih problema ovaj slučaj se ne pojavljuje jer će ograničavajući faktori uvek formirati sistem neprotivurečnih nejednačina.
VI. Optimalno rešenje linearnog problema ne postoji i u slučaju kada je oblast dopustivih rešenja *D* neograničena a istovremeno ciljna funkcija, koja se maksimizira ili minimizira, beskonačno raste odnosno opada. Ovo, takođe, nije karakteristika realnih problema.
VII. Ukoliko su dobijena dva optimalna rešenja, tada su sve tačke duži između tih tačaka optimalna rešanja. Dakle ako su $x^{(1)}\in D$ i $x^{(2)}\in D$ optimalna rešenja linearnog problema, onda je i svaka konveksna kombinacuja $x_q^{(0)}=q\cdot x^{(1)}+(1-q)\cdot x^{(2)}$, $0\leq q\leq 1$ optimalno rešenje.

Iz pomenutih, zaključnih, razmatranja uočava se da ekstremne tačke imaju poseban značaj, jer je njihov broj uvek konačan, a optimalno rešenje problema se nalazi preko ekstremnih tačaka u konačnom broju iteracija.

2.7. MATRIČNI METOD

Problem linearnog programiranja, izražen u matričnom obliku, gde je potrebno odrediti nenegativni vektor *X* tako da funkcija cilja dostigne svoju maksimalnu vrednost, je predstavljen kao:

$$\max F(x) = C^T\cdot X \tag{2.7.1.}$$

uz linearna ograničenja:

$$A\cdot X = b \tag{2.7.2.}$$

$$X \geq 0 \tag{2.7.3.}$$

Vektor C^T je po formi vektor red $n+m$-tog reda, *X* vektor kolone, takođe, $n+m$-tog reda, *b* vektor kolone *m*-tog reda, *A* matrica koeficijenata sistema ograničenja reda $(m\times n+m)$. Kolone matrice *A* mogu se posmatrati kao vektori kolona *m*-tog reda, tj. $A = [A_1\ A_2 \dots A_n \dots A_{n+m}]$, a označavaju se kao [42]:

$$A_1=\begin{bmatrix}a_{11}\\a_{21}\\\vdots\\a_{m1}\end{bmatrix},\ A_2=\begin{bmatrix}a_{12}\\a_{22}\\\vdots\\a_{m2}\end{bmatrix},\cdots,A_n=\begin{bmatrix}a_{1n}\\a_{2n}\\\vdots\\a_{mn}\end{bmatrix},\ A_{n+1}=\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix},\ A_{n+2}=\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix},\cdots,A_{n+m}=\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix} \tag{2.7.4.}$$

Sistem linearnih jednačina ograničenja se sada može predstaviti kao:

$$A_1\cdot x_1 + A_2\cdot x_2 + \dots + A_n\cdot x_n + A_{n+1}\cdot x_{n+1} + \dots + A_{n+m}\cdot x_{n+m} = b \tag{2.7.5.}$$

ili u sažetom obliku:

$$\sum_{j=1}^{n+m} A_j x_j = b \qquad (2.7.6.)$$

Pošto su vektori A_j i b vektori m-tog reda, to znači da između $n+m$ vektora tipa A_j može postojati najviše m nezavisnih. Ukoliko je u jednačini (2.7.5.) prvih m vektora linearno nezavisno, vektor b se može izraziti kao:

$$A_1 \cdot x_1 + A_2 \cdot x_2 + ... + A_m \cdot x_m = b \qquad (2.7.7.)$$

Izbor m vektora se može realizovati na C_m^n kombinacija, kao što je prikazano relacijom (2.5.1.). Svaka kombinacija daje nov rezultat u funkciji cilja, tj. novu vrednost vektora X. Najbolja kombinacija donosi optimalno rešenje.

Od baznih vektora $A = [A_1 \;\; A_2 \; ... \; A_n \; ... \; A_{n+m}]$ može se formirati kvadratna matrica Ab, dimenzija $m \times m$. Ona je podmatrica matrice A i može se izraziti kao:

$$Ab = [A_1 \; A_2 \; ... \; A_m] \qquad (2.7.8.)$$

regularna je, tj. važi:

$$\text{dat}Ab \neq 0 \;\; \text{ili} \;\; |Ab| \neq 0 \qquad (2.7.9.)$$

Promenljive koje odgovaraju kolonama podmatrice Ab zovu se *bazne promenljive*, a preostale su *nebazne promenljive*. Skup svih baznih promenljivih čini *bazu* baznog rešenja, dok je Ab je *matrica baze*.

Od m pozitivnih koordinata vektota X (koje stoje uz bazične vektore) formira se vektor $Xb = [x_1, x_2, ... x_m]$.

Sada se sistem ograničenja (2.7.2.), odnosno vektor b, može izraziti pomoću matrice baze Ab, na sledeći način:

$$Ab \cdot Xb = b \qquad (2.7.10.)$$

Pomoću inverzne matrice Ab^{-1} se mogu odrediti koordinate vektora Xb, tj. rešenje problema za posmatranu iteraciju. Obe strane izraza (2.7.10.) se može inverznom matricom Ab^{-1} sa leve strane, na sledeći način:

$$Ab^{-1} \cdot Ab \cdot Xb = Ab^{-1} \cdot b$$
$$Ab^{-1} \cdot Ab = I$$
$$I \cdot Xb = Ab^{-1} b$$
$$Xb = Ab^{-1} \cdot b \qquad (2.7.11.)$$

Bazna vrednost funkcije cilja za pronađeno rešenje je:

$$F(X) = Cb \cdot Xb$$

zamenom se dobija da je:

$$F(X) = Cb \cdot Ab^{-1} \cdot b \qquad (2.7.12.)$$

gde vektor Cb obuhvata koordinate vektora C koje stoje uz bazne promenljive (vektor Xb) u funkciji cilja.

Linearno zavisni vektori matrice A se mogu izraziti pomoću matrice baze Ab, na sledeći način:

$$Ab \cdot X_j = A_j \qquad (2.7.13.)$$

gde je A_j, $j = (m+1, m+2, ..., n)$, nebazni vektor. Nakon sređivanja dobija se izraz:

$$X_j = Ab^{-1} \cdot A_j \qquad (2.7.14.)$$

U ovom slučaju vrednost funkcije cilja iznosi:

$$F(X_j) = Cb \cdot X_j$$

zamenom se dobija da je:

$$F(X_j) = F(j) = Cb \cdot Ab^{-1} \cdot A_j \qquad (2.7.15.)$$

Relacijom (2.7.14.) izvršena je transformacija koeficijenata matrice A, pa je sada moguće izračunati i koeficijente $(F_j - c_j)$. Ovi koeficijenti omogućavaju ocenu optimalnosti pronađenog rešenja a nakon toga i izbor vektora koji će da uđe u narednu bazu vektorskog prostora. Ti koeficijenti se određuju na osnovu relacije:

$$F_j - c_j = Cb \cdot X_j - c_j, \; j = (m+1, ..., n) \qquad (2.7.16.)$$

Primer 2.7.1. Dat je matematički model

$$F(x) = 2x_1 + x_2$$

po

$$\begin{aligned} x_1 + x_2 &\leq 3 \\ x_1 &\leq 2 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Odrediti:

a) maksimalnu vrednost funkcije cilja, matričnim izračunavanjem,
b) maksimalnu vrednost funkcije cilja, grafičkim metodom.

Rešenje.

a) Matrični proračun

Prošireni oblik matematičkog modela je:

$$\begin{aligned} &\max F(x) = 2x_1 + x_2 + 0 \cdot (x_3 + x_4) \\ &x_1 + x_2 + x_3 = 3 \\ &x_1 + x_4 = 2 \\ &x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

U modelu postoje n=4 promenljive i m=2 ograničenja. Rang matrice sistema je $r(A)$=2. Matrice i vektori sistema su:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 1 & 0 & 0 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Rešiti sistem znači da je potrebno prikazati vektor b kao linearnu kombinaciju vektora A_1,...,A_4, tj. izračunavaju se skalarne vrednosti x_1,...,x_4, takve da je

$$A_1 \cdot x_1 + A_2 \cdot x_2 + A_3 \cdot x_3 + A_4 \cdot x_4 = b$$

Pošto je rang matrice $r(A)=2$, znači da između vektora $A_1,...,A_4$ postoje dva linearno nezavisna vektora koji čine bazu vektorskog prostora R^2. Izbor kombinacije dva nezavisna vektora $A_1,...,A_4$ se može realizovati na C_m^n kombinacija, tj.

$$C_m^n = \frac{n!}{m!(n-m)!} = \frac{4!}{2!(4-2)!} = \frac{3\cdot 4}{(2)!} = \frac{12}{2} = 6$$

Skup mogućih kombinacija bi bio: $\{A_1,A_2\},\{A_1,A_3\},\{A_1,A_4\},\{A_2,A_3\},\{A_2,A_4\},\{A_3,A_4\}$. Za dalji postupak su interesantne linearne kombinacije $A_1\cdot x_1+A_2\cdot x_2+A_3\cdot x_3+A_4\cdot x_4 = b$ takve da je vektor b linearna kombinacija dva linearno nezavisna vektora. U ovom primeru postoje pet kombinacija linearno nezavisnih vektora$\{A_1,A_2\},\{A_1,A_3\},\{A_1,A_4\},\{A_2,A_4\},\{A_3,A_4\}$. Linearna kombinacija $\{A_2,A_3\}$ je kombinacija dva linearno zavisna vektora, tj. njihova $\det A_{23}=0$, to znači da matrica nije regularna i ne postoji A_{23}^{-1} matrice A_{23}.

- Proizvoljno se uzimaju kombinacije dva linearno nezavisna vektora. Uzeta je prva kombinacija$\{A_1,A_2\}$, tj. $Ab=[A_2\ A_3]$.

$$Ab = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\quad \det Ab = 1\cdot 0 - 1\cdot 1 = -1,\quad Ab^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\quad \operatorname{adj} Ab = \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}$$

$$Ab^{-1} = \frac{1}{\det Ab}\cdot \operatorname{adj} Ab = \frac{1}{-1}\cdot \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$$

$$Xb = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = Ab^{-1}\cdot b = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}\cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 0\cdot 3+1\cdot 2 \\ 1\cdot 3+-1\cdot 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix},\quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

- Kombinacija dva linearno nezavisna vektora$\{A_1,A_3\}$, tj. $Ab=[A_1\ A_3]$.

$$Ab = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\quad \det Ab = 1\cdot 0 - 1\cdot 1 = -1,\quad Ab^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\quad \operatorname{adj} Ab = \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}$$

$$Ab^{-1} = \frac{1}{\det Ab}\cdot \operatorname{adj} Ab = \frac{1}{-1}\cdot \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$$

$$Xb = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = Ab^{-1}\cdot b = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}\cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 0\cdot 3+1\cdot 2 \\ 1\cdot 3+(-1)\cdot 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix},\quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- Kombinacija dva linearno nezavisna vektora$\{A_1,A_4\}$, tj. $Ab=[A_1\ A_4]$.

$$Ab = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},\quad \det Ab = 1\cdot 1 - 0\cdot 1 = 1,\quad Ab^T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\quad \operatorname{adj} Ab = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$Ab^{-1} = \frac{1}{\det Ab}\cdot \operatorname{adj} Ab = \frac{1}{1}\cdot \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$Xb = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = Ab^{-1}\cdot b = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}\cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1\cdot 3+0\cdot 2 \\ (-1)\cdot 3+1\cdot 2 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix},\quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

- Kombinacija dva linearno zavisna vektora$\{A_2,A_3\}$, tj. $Ab=[A_2\ A_3]$.

$$\mathrm{Ab}=\begin{bmatrix}1&1\\0&0\end{bmatrix},\quad \det\mathrm{Ab}=1\cdot0-0\cdot1=0,\quad \mathrm{Ab}^{T}=\begin{bmatrix}1&0\\1&0\end{bmatrix},\quad \mathrm{adjAb}=\begin{bmatrix}0&-1\\0&1\end{bmatrix}$$

$$\mathrm{Ab}^{-1}=\frac{1}{\det\mathrm{Ab}}\cdot\mathrm{adjAb}\Rightarrow \text{nije regularna}$$

- Kombinacija dva linearno nezavisna vektora$\{A_2,A_4\}$, tj. $Ab=[A_2\ A_4]$.

$$\mathrm{Ab}=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad \det\mathrm{Ab}=1\cdot1-0\cdot0=1,\quad \mathrm{Ab}^{T}=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad \mathrm{adjAb}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$\mathrm{Ab}^{-1}=\frac{1}{\det\mathrm{Ab}}\cdot\mathrm{adjAb}=\frac{1}{1}\cdot\begin{bmatrix}1&0\\0&1\end{bmatrix}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$\mathrm{Xb}=\begin{bmatrix}x_2\\x_4\end{bmatrix}=\mathrm{Ab}^{-1}\cdot b=\begin{bmatrix}1&0\\0&1\end{bmatrix}\cdot\begin{bmatrix}3\\2\end{bmatrix}=\begin{bmatrix}1\cdot3+0\cdot2\\0\cdot3+1\cdot2\end{bmatrix}=\begin{bmatrix}3\\2\end{bmatrix},\quad X=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}0\\3\\0\\2\end{bmatrix}$$

- Kombinacija dva linearno nezavisna vektora$\{A_3,A_4\}$, tj. $Ab=[A_3\ A_4]$.

$$\mathrm{Ab}=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad \det\mathrm{Ab}=1\cdot1-0\cdot0=1,\quad \mathrm{Ab}^{T}=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad \mathrm{adjAb}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$\mathrm{Ab}^{-1}=\frac{1}{\det\mathrm{Ab}}\cdot\mathrm{adjAb}=\frac{1}{1}\cdot\begin{bmatrix}1&0\\0&1\end{bmatrix}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$\mathrm{Xb}=\begin{bmatrix}x_3\\x_4\end{bmatrix}=\mathrm{Ab}^{-1}\cdot b=\begin{bmatrix}1&0\\0&1\end{bmatrix}\cdot\begin{bmatrix}3\\2\end{bmatrix}=\begin{bmatrix}1\cdot3+0\cdot2\\0\cdot3+1\cdot2\end{bmatrix}=\begin{bmatrix}3\\2\end{bmatrix},\quad X=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}0\\0\\3\\2\end{bmatrix}$$

Bazna rešenja (**BR**) sistema su:

$$\begin{bmatrix}2\\1\\0\\0\end{bmatrix},\begin{bmatrix}2\\0\\1\\0\end{bmatrix},\begin{bmatrix}3\\0\\0\\-1\end{bmatrix},\begin{bmatrix}0\\3\\0\\2\end{bmatrix},\begin{bmatrix}0\\0\\3\\2\end{bmatrix}$$

Bazno moguća rešenja (**BMR**) sistema su.

$$\begin{bmatrix}2\\1\\0\\0\end{bmatrix},\begin{bmatrix}2\\0\\1\\0\end{bmatrix},\begin{bmatrix}0\\3\\0\\2\end{bmatrix},\begin{bmatrix}0\\0\\3\\2\end{bmatrix}$$

Bazno rešenje $BR_3=[3\ \ 0\ \ 0\ \ -1]^T$ nije BMR jer nije ispunjen uslov o nenegativnosti promenljivih $x_1,x_2,x_3,x_4 \geq 0$.

Optimalno rešenje problema se nalazi u nekom od bazno mogućih rešenja, tj. onoj kombinaciji gde funkcija cilja dostiže svoju maksimalnu vrednost, tj:

$\{A_1,A_2\}$; $F(x) = 2x_1 + x_2 = 2\cdot2 + 1\cdot1 = 4 + 1 = 5$

$\{A_1,A_3\}$; $F(x) = 2x_1 + x_2 = 2\cdot2 + 1\cdot0 = 4 + 0 = 4$

$\{A_2,A_4\}$; $F(x) = 2x_1 + x_2 = 2\cdot0 + 1\cdot3 = 0 + 3 = 3$

$$\{A_3,A_4\};\quad F(x) = 2x_1 + x_2 = 2\cdot 0 + 1\cdot 0 = 0 + 0 = 0$$

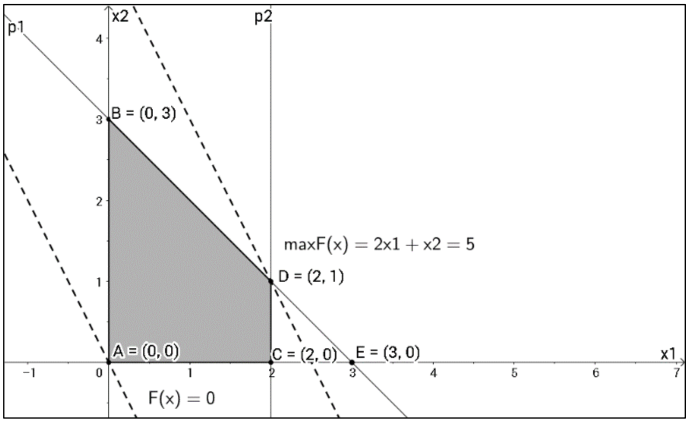
Funkcija cilja dostiže maksimalnu vrednost za kombinaciju $\{A_1,A_2\}$, tj optimalno rešenje je:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \max F(x) = 5$$

Promenljive x_1 i x_2 su bazne, a x_3 i x_4 nebazne. Bazno moguće rešenje je ***nedegenerisano*** ukoliko su sve bazne promenljive različite od nule ($x_1,x_2 \neq 0$), kao što je slučaj u ovom primeru. Ukoliko postoji bar jedna bazna promenljiva koja je jednaka nuli bazno moguće rešenje je ***degenerisano*** ($x_1=0 \vee x_2=0 \vee x_1,x_2=0$).

b) Grafička metoda

Na slici 2.7.1. prikazana su sva bazna rešenja (BR) koja se nalaze u tačkama A=(0,0); B=(0,3); C=(2,0); D=(2,1); E=(3,0); Bazna moguća rešenja (BMR) su u tačkama A, B, C i D, dok tačka E nije bazno moguće rešenje. Optimalno rešenje se nalazi u tačci D=(2,1), gde funkcija cilja dostiže maksimalnu vrednost, tj. max $F(x)$= 5. Dobijeno bazno moguće rešenje je nedegenerisano jer su sve bazne promenljive različite od nule ($x_1,x_2 \neq 0$).



Slika 2.7.1. *Rešenje problema zadatka 2.7.1.*

2.8. SIMPLEKS METOD

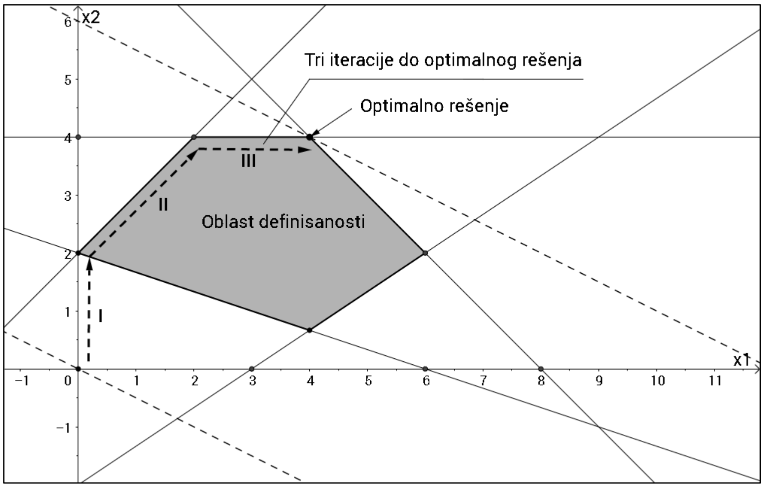
Simpleks metod je iterativni postupak iznalaženja optimalnog rešenja modela linearnog programiranja. Prvi put je razvijen od strane poznatog američkog matematičara George B. Dantzig-a 1948. godine. Zato se u literaturi često sreće i kao Dantzig-ova metoda. Za rešavanje problema LP postavljeno je i razrađeno više algoritama simpleks metode. To su računske procedure pomoću kojih se dolazi do optimalnog rešenja, koje se međusobno neznatno razlikuju jer koriste isti pristup u postupku rešenja problema. Simpleks metoda se zasniva na tri bitna principa:

1. Postoji mogućnost određivanja bar jednog dopustivog rešenja (plana), koji se cesto naziva bazičnim planom ili dopustivim bazičnim planom.
2. Postoji mogućnost da se proveri da li je bazični dopustivi plan optimalan ili ne.

3. Postoji mogućnost, da se u slučaju da dopustivi plan nije optimalan, izabere novi, koji je bliži optimalnom.

Simpleks metod se zasniva na sukcesivnom poboljšanju početnog dopustivog plana, sve dok se ne dobije optimalan plan. Algoritam simpleks metoda takođe omogućava da se ustanovi da li je zadatak rešiv ili ne, odnosno, da li postoje protivurečnosti u ograničenjima.

Način rešavanja problema simpleks metodom može se grafički prikazati kao na slici 2.8.1.



Slika 2.8.1. *Grafički prikaz simpleks metode*

Funkcija cilja polazi od koordinatnog početka i u tom trenutku ima vrednost jednaku nuli, tj. $F(x)=0$. Iterativnim postupkom, korak po korak, najkraćim putem, stiže se do optimalnog rešenja.

2.8.1. Procedura rešavanja simpleks problema

Za objašnjavanje procedure rešavanja simpleks metodom uzet je u razmatranje opšti problem linearnog programiranja, koji glasi: Potrebno je pronaći nenegativne vrednosti promenljivih x_1, x_2,..., x_n koje će zadovoljiti sistem ograničenja koji je dat linearnim nejednačinama i jednačinama

$$\sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ = \\ \geq \end{Bmatrix} b_i \quad i = 1, 2, \ldots, m$$

i obezbediti da funkcija cilja

$$\max \ F(x) = \sum_{j=1}^{n} c_j x_j$$

dostigne svoju maksimalnu vrednost.

Ovako formulisani problemi linearnog programiranja rešavaju se simpleks metodom prema sledećoj proceduri:

1. Sve nejednačine sistema prevodimo u jednačine. Ukoliko je nejednačina tipa ≤ onda se dodaje izravnavajuća promenljiva; ako je, nejednačina tipa ≥, onda se odgovarajuća izravnavajuća promenljiva oduzima od leve strane nejednačine. Uz izravnavajuće promcnljive u funkciji cilja uvek stoje nule. Veštačke promenljive se dodaju svim jednačinama i nejednačinama tipa ≥. Uz ove promenljive u funkciji cilja stavljaju se koeficijenti M.
2. Od koeficijenata prilagođenog modela sastavlja se početna simpleks tabela. Bazu vektorskog prostora za početno rešenje čine jedinični vektori. To su vektori koji odgovaraju „dodatim" promenljivim (dodate izravnavajuće i sve veštačke).
3. Za dodatni red simpleks tabele računa se koeficijent $F_j – c_j$, $\forall j$. Simpleks tabela sadrži početno rešenje.
4. Proveravaju se koeficijenti reda $(F_j – c_j)$. Ukoliko je:

   $F_j – c_j \geq 0$, $\forall j$, onda se tvrdi da je pronađeno optimalno rešenje.
   $F_j – c_j < 0$, $\forall j$, onda biramo:
   $$F_s - c_s = \max_j \left|(F_j - c_j) < 0\right| \qquad (2.8.1.)$$

   Koeficijent $(F_s–c_s)$ odgovara vektoru A_s, pa se određuje da vektor A_s uđe u novu bazu vektora.
5. Koordinate vektora b (kolona B simpleks tabele) deli se sa odgovarajućim pozitivnim koordinatama vektora A_s i odreduje se:
   $$\theta = \frac{b_r}{a_{rs}} = \min_i \frac{b_i}{a_{is}};\ a_{is} > 0,\ i = 1,2,\ldots,m \qquad (2.8.2.)$$

   Najmanja vrednost od ovih količnika odgovara vektoru A_r, pa se određuje da vektor A_r izlazi iz početne baze.
6. Koeficijenti za novu simpleks tabelu (ST_1) se izračunavaju na osnovu koeficijenata iz početne simpleks tabele (ST_0) prema sledećim pravilima:

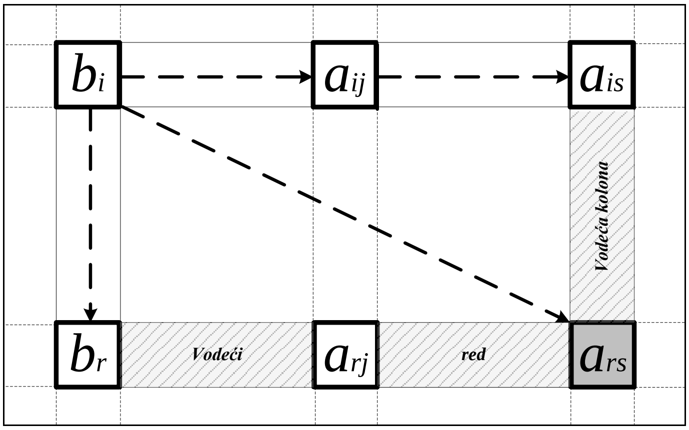
$$b_r' = \frac{b_r}{a_{rs}}, \quad i = r \qquad (2.8.3.)$$

$$b_i' = b_i - \frac{b_r}{a_{rs}} \cdot a_{is}, \quad i \neq r \qquad (2.8.4.)$$

$$a_{rj}' = \frac{a_{rj}}{a_{rs}}, \quad i = r \qquad (2.8.5.)$$

$$a_{ij}' = a_{ij} - \frac{a_{rj}}{a_{rs}} \cdot a_{is}, \quad i \neq r \qquad (2.8.6.)$$

Ovaj proračun se može obaviti i na osnovu šeme, prikazane na slici 2.8.2. Transformisani koeficijenti se unose u novu simpleks tabelu, koja sadrži i novo moguće rešenje.

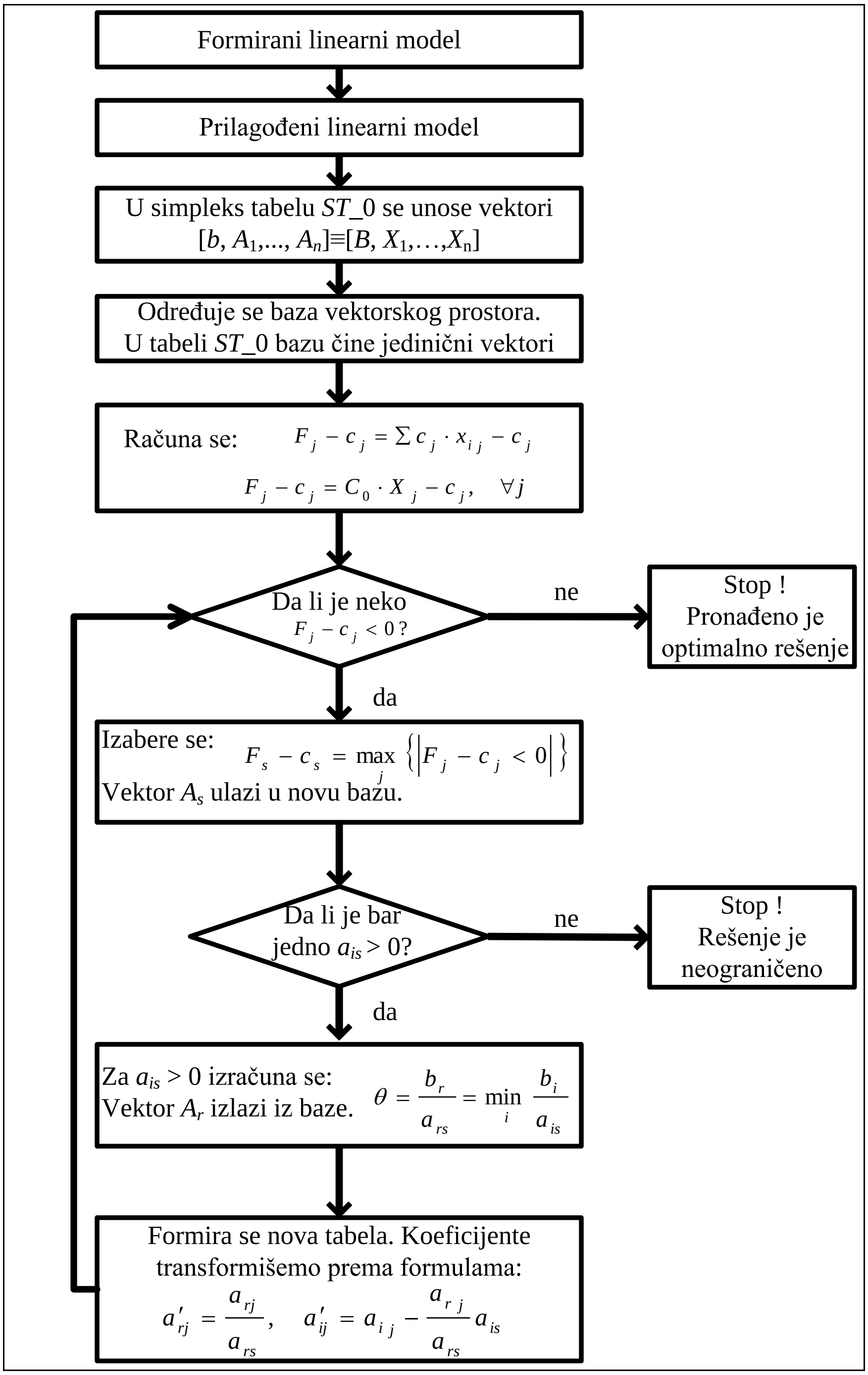


Slika 2.8.2. *Preračunavanje koeficijenata za narednu simpleks tabelu* (Dantzig, 1963)

7. Postupak se vraća na na tačku 4. iz ove procedure. Svaka naredna iteracija počinje tačkom 4. i ponavlja postupak do tačke 7., sve dok se u tačci 4. ne utvrdi da su sve razlike $F_j - c_j \geq 0$. Tada se postupak završava, pronađeno je optimalno rešenje linearnog problema.

Ceo postupak rešavanja linearnih problema, opisan ovom procedurom, može se predstaviti algoritmom, tj. fazama procesa izvođenja optimizacije, kao što je to predstavljeno na slici 2.8.3.

U prikazivanju simpleks algoritama, za rešavanje problema linearnog programiranja, u literaturi postoji velika šarolikost. Neki autori prikazuju algoritam u vidu rešavanja sistema jednačina primenom metoda zamene uz dodatni kriterijum optimalnosti, postoji tzv. matrični način izvođenja, i preko simpleks tabele. Kod primene simpleks tabele neki autori rešavaju kanonični oblik, a neki opšti oblik zapisa problema linearnog programiranja.



Slika 2.8.3. *Algoritam simpleks metoda*

2.8.2. Algebarska interpretacija simpleks metode

Logika simpleks metoda je u izboru prvog baznog rešenja sastavljenog od vektora dopunskih promenljivih i postepenom poboljšavanju dok se ne postigne optimum. Poboljšavanja se izvode zamenom baznih vektora i pri tome se vrše preračunavanja koeficijenata novih linearnih kombinacija. Velika prednost simpleks metode je u tome što se ne ispituju sva moguća bazna rešenja kojih ima c_m^n. Algebarska interpretacija simpleks metode biće razrađena na primerima pronalaženja minimuma i maksimuma funkcije cilja.

Osnovni zadatak linearnog programiranja se zapisuje u obliku u kome su prvih *k* promenljivih slobodne a ostalih *m* su bazne, kao:

$$\begin{aligned} x_{k+1} &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + b_1 \\ x_{k+2} &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k + b_2 \\ &\ldots \\ x_{k+m} &= a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mk}x_k + b_m \end{aligned} \tag{2.8.7.}$$

Funkcija cilja se izražava preko slobodnih promenljivih:

$$F(x) = c_0 + c_1x_1 + c_2x_2 + \ldots + c_kx_k \tag{2.8.8.}$$

Algebarska interpretacija simpleks metode biće razrađena na primeru pronalaženja minimuma i maksimuma funkcije kriterijuma.

Primer 2.8.1. Pronaći optimalno rešenje datog problema linearnog programiranja koristeći algebarsku interpretaciju simpleks metode.

$$\begin{aligned} & \min F(x) = 3x_1 - 6x_2 - 2x_3 + 7x_4 \\ \text{p.o.}\quad & 1x_1 + 7x_2 + 3x_3 + 7x_4 \le 46 \\ & 3x_1 - 1x_2 + 1x_3 + 2x_4 \le 8 \\ & 2x_1 + 3x_2 - 1x_3 + 1x_4 \le 10 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

Rešenje:

- Prevođenje u standardni oblik problema minimuma:

$$\begin{aligned} -1x_1 - 7x_2 - 3x_3 - 7x_4 &\ge -46 \\ -3x_1 + 1x_2 - 1x_3 - 2x_4 &\ge -8 \\ -2x_1 - 3x_2 + 1x_3 - 1x_4 &\ge -10 \end{aligned}$$

- Svođenje na osnovni zadatak LP uvođenjem dopunskih promenljivih x_5, x_6, x_7, koje su ujedno i bazne promenljive:

$$\begin{aligned} -1x_1 - 7x_2 - 3x_3 - 7x_4 &= -46 + x_5 \\ -3x_1 + 1x_2 - 1x_3 - 2x_4 &= -8 \qquad + x_6 \\ -2x_1 - 3x_2 + 1x_3 - 1x_4 &= -10 \qquad\qquad + x_7 \end{aligned}$$

$$\begin{aligned} x_5 &= -1x_1 - 7x_2 - 3x_3 - 7x_4 + 46 \\ x_6 &= -3x_1 + 1x_2 - 1x_3 - 2x_4 + 8 \\ x_7 &= -2x_1 - 3x_2 + 1x_3 - 1x_4 + 10 \end{aligned}$$

- Početno bazno rešenje je: $x_1=x_2=x_3=x_4=0$, $x_5=46$, $x_6=8$, $x_7=10$, $F(x)=0$.

– Izbor slobodne promenljive za ulazak u bazu vrši se prema kriterijumu:

$$\left\{\max_j |c_j|,\ j : c_j < 0\right\} \qquad (2.8.9.)$$

c_j su koeficijenti koji stoje uzpromenljive u funkciji cilja. Razmatra se samo apsolutna vrednost negativnih koeficijenata. Koeficijent koji ima veću apsolutnu vrednost ulazi u bazu. Kako je $|c_2|>|c_3|$ za ulazak u bazu bira se promenljiva x_2.

– Izbor promenljive za izlazak iz baze vrši se na osnovu θ ktiterijuma:

$$\theta = \min_i \left(-\frac{b_i}{a_{ij}}\right) \qquad (2.8.10.)$$

gde su:

– b_i – slobodni koeficijenti u ograničenjima,
– a_{ij} – koeficijenti koji stoje uz promenljivu koja ulazi u bazu u svim ograničenjima.

U ovom primeru vektor $b_i=(b_1, b_2, b_3)=(46, 8, 10)$, a koeficijenti koji stoje uz promenljivu x_2 su $a_{2j}=(a_{21}, a_{22}, a_{23})=(–7, 1, –3)$. Najmanji pozitivni θ koeficijent izlazi iz baze, tj.:

$$\theta = \min_{i=1,2,3}\left(-\frac{b_i}{a_{ij}}\right) = \min\left(-\frac{b_1}{a_{21}};-\frac{b_2}{a_{22}};-\frac{b_3}{a_{23}}\right) = \min\left(-\frac{46}{-7};-\frac{8}{1};-\frac{10}{-3}\right) = \min\left(\frac{46}{7};\frac{10}{3}\right) = \frac{10}{3}$$

Bazu napušta promenljiva x_7, zato što je u trećem ograničenju najmanji θ kriterijum.

– Rešenje posle I iteracije-nakon što se bazne promenljive x_2, x_5 i x_6 izraze preko slobodnih promenljivih:

$$x_2 = –0{,}667x_1 + 0{,}333x_3 – 0{,}333x_4 – 0{,}333x_7 + 3{,}333$$
$$x_5 = +3{,}667x_1 – 5{,}333x_3 – 4{,}667x_4 + 2{,}333x_7 + 22{,}667$$
$$x_6 = –3{,}667x_1 + 0{,}667x_3 – 2{,}333x_4 – 0{,}333x_7 + 11{,}333$$
$$F(x) = 7x_1 – 4x_3 + 6x_4 + 2x_7 – 20$$

za $x_1=x_3=x_4=x_7=0$ dobija se $F(x)= –20$.

– II iteracija - u bazu ulazi promenljiva x_3 jer jedino ona ima negativan predznak u funkciji cilja. Iz baze izlazi promenljiva x_5 jer samo u jednačini za baznu promenljivu x_5 promenljiva x_3 ima negativan predznak. Nove bazne promenjljive x_2, x_3, x_6 se izraze pomoću slobodnih x_1, x_4, x_5 i x_7:

$$x_2 = –0{,}438x_1 – 0{,}625x_4 – 0{,}063x_5 – 0{,}188x_7 + 4{,}75$$
$$x_3 = +3{,}688x_1 – 0{,}875x_4 – 0{,}188x_5 + 0{,}438x_7 + 4{,}25$$
$$x_6 = –4{,}125x_1 – 1{,}75x_4 + 0{,}125x_5 – 0{,}333x_7 + 8{,}5$$
$$F(x) = 4{,}25x_1 + 9{,}5x_4 + 0{,}75x_5 + 0{,}25x_7 – 37$$

Za novo bazno rešenje funkcija cilja ima vrednost –37, što je i optimalno rešenje, tj. $\min F(x)=–37$, jer u funkciji cilja ne postoji promenljiva sa negativnim predznakom. Vrednosti promenljivih za optimalno rešenje su: $X^*=(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*)=(0, 4.75, 4.25, 0, 0, 8.5, 0)$.

Primer 2.8.2. Pronaći optimalno rešenje datog problema linearnog programiranja koristeći algebarsku interpretaciju simpleks metode.

$$\max F(x) = 2x_1 + 1x_2 – 3x_3 + 5x_4$$

p.o.
$$1x_1 + 7x_2 + 3x_3 + 7x_4 \le 46$$
$$3x_1 – 1x_2 + 1x_3 + 2x_4 \le 8$$
$$2x_1 + 3x_2 – 1x_3 + 1x_4 \le 10$$
$$x_1, x_2, x_3, x_4 \ge 0$$

Rešenje:

- Svođenje problema na osnovni zadatak LP uvođenjem dopunskih promenljivih x_5,x_6,x_7, koje su ujedno i bazne promenljive, matematički model dobija oblik:

$$
\begin{aligned}
x_5 &= -1x_1 - 7x_2 - 3x_3 - 7x_4 + 46\\
x_6 &= -3x_1 + 1x_2 - 1x_3 - 2x_4 + \ \ 8\\
x_7 &= -2x_1 - 3x_2 + 1x_3 - 1x_4 + 10
\end{aligned}
$$

- Početno bazno rešenje je: x_1=x_2=x_3=x_4=0, x_5=46, x_6=8, x_7=10, $F(x)$=0.
- Izbor slobodne promenljive za ulazak u bazu vrši se prema kriterijumu:

$$\left\{\max_j |c_j|,\ j : c_j > 0\right\} \qquad (2.8.11.)$$

Kako je $|c_4|>|c_1|>|c_2|$ za ulazak u bazu bira se promenljiva x_4.

- Izbor promenljive za izlazak iz baze vrši se na osnovu θ ktiterijuma (8.10):

$$\theta = \min_i\left(-\frac{b_i}{a_{ij}}\right)$$

U ovom primeru vektor b_i=(b_1,b_2,b_3)=(46,8,10), a koeficijenti koji stoje uz promenljivu x_4 su a_{2j}=(a_{21},a_{22},a_{23})=(–7,–2,–3). Najmanji pozitivni θ koeficijent izlazi iz baze, tj.:

$$\theta = \min_i\left(\frac{-b_i}{a_{ij}}\right) = \min_{j=1,2,3}\left(\frac{-b_1}{a_{21}};\frac{-b_2}{a_{22}};\frac{-b_3}{a_{23}}\right) = \min\left(\frac{-46}{-7};\frac{-8}{-2};\frac{-10}{-1}\right) = \min\left(\frac{46}{7};\frac{8}{2};\frac{10}{1}\right) = 4$$

Bazu napušta promenljiva x_6, zato što je u drugom ograničenju najmanji θ kriterijum.

- Rešenje posle I iteracije - nakon što se bazne promenljive x_4, x_5 i x_7 izraze preko slobodnih promenljivih:

$$
\begin{aligned}
x_4 &= -1{,}5x_1 + \ 0{,}5x_2 - 0{,}5x_3 - 0{,}5x_6 + 4\\
x_5 &= 99{,}5x_1 - 10{,}5x_2 + 0{,}5x_3 + 3{,}5x_6 + 18\\
x_7 &= -0{,}5x_1 - \ 3{,}5x_2 + 1{,}5x_3 + 0{,}5x_6 + \ 6\\
F(x) &= -5{,}5x_1 + 3{,}5x_2 - 5{,}5x_3 - 2{,}5x_6 + 20
\end{aligned}
$$

za x_1=x_2=x_3=x_6=0 dobija se $F(x)$= 20.

- II iteracija - u bazu ulazi promenljiva x_2 jer jedino ona ima pozitivan predznak u funkciji cilja. Za izlazak iz baze konkurišu promenljive x_5 i x_7 jer je:

$$\theta = \min_{i=1,2,3}\left(\frac{-b_i}{a_{ij}}\right) = \min\left(\frac{-b_1}{a_{21}};\frac{-b_2}{a_{22}};\frac{-b_3}{a_{23}}\right) = \min\left(\frac{-4}{0{,}5};\frac{-18}{-10{,}5};\frac{-6}{-3{,}5}\right) = \min\left(;\frac{6}{3{,}5};\frac{6}{3{,}5}\right) = \frac{6}{3{,}5}$$

U ovom slučaju nije bitno koja će promenljiva da napusti bazu. Proizvoljno se bira da bazu napusti promenljiva x_5. Nove bazne promenjljive x_2, x_4, x_7 se izraze pomoću slobodnih x_1, x_3, x_5 i x_6:

$$
\begin{aligned}
x_2 &= \ \ 0{,}905x_1 + 0{,}048x_3 - 0{,}095x_5 + 0{,}333x_6 + 1{,}714\\
x_4 &= -1{,}048x_1 - 0{,}476x_3 - 0{,}048x_5 - 0{,}333x_6 + 4{,}875\\
x_7 &= -3{,}667x_1 + 1{,}333x_3 + 0{,}333x_5 - 0{,}667x_6 + 0\\
F(x) &= -2{,}333x_1 - 5{,}333x_3 - 0{,}333x_5 - 1{,}333x_6 + 26
\end{aligned}
$$

Za novo bazno rešenje funkcija cilja ima vrednost +26, što je i optimalno rešenje, tj. max$F(x)$=26, jer u funkciji cilja ne postoji promenljiva sa pozitivnim predznakom. Vrednosti promenljivih za optimalno rešenje su: $X^*=(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*)$=(0, 1.714, 0, 4.875, 0, 0, 0).

2.8.3. Matrična interpretacija simpleks metode

Zadatak linearnog programiranja može se prikazati i u vektorskom obliku. Traži se nenegativni vektor X koji će zadovoljiti sistem ograničenja i datu finkciju cilja, tj:

$$F(x) = C^T \cdot X$$
$$A \cdot X = b$$
$$X \geq 0$$

pri čemu je:

- X – vektor kolona promenljivih x_j
- C^T– vektor reda koeficijenata c_j
- b – vektor kolona nezavisnih članova
- A – matrica koeficijenata a_{ij}

Matrična interpretacija simpleks metode biće prikazana na primeru problema LP gde se zahteva proračun maksimalne vrednosti funkcije cilja.

Primer 2.8.3. Treba naći nenegativno rešenje sistema ograničenja koje će dati maksimalnu vrednost funkcije cilja, tj:

$$\begin{array}{ll} & \max F(x) = x_1 + x_2 + x_3 \\ \text{p.o.} & 6x_1 + 5x_2 + 4x_3 \leq 18 \\ & 2x_1 + 8x_2 + 3x_3 \leq 15 \\ & 4x_1 + 2x_2 + 5x_3 \leq 14 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Rešenje. Uvođenjem dopunskih promenljivih x_4, x_5 i x_6 dati problem se svodi na oblik simpleksnih jednačina:

$$\begin{array}{ll} & \max F(x) = x_1 + x_2 + x_3 + 0\cdot(x_4 + x_5 + x_6) \\ \text{p.o.} & 6x_1 + 5x_2 + 4x_3 + x_4 = 18 \\ & 2x_1 + 8x_2 + 3x_3 + x_5 = 15 \\ & 4x_1 + 2x_2 + 5x_3 + x_6 = 14 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{array}$$

Vektorski prikaz datog problema je sledeći:

$$\begin{bmatrix} 6 & 5 & 4 & 1 & 0 & 0 \\ 2 & 8 & 3 & 0 & 1 & 0 \\ 4 & 2 & 5 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \cdots \\ x_6 \end{bmatrix} = \begin{bmatrix} 18 \\ 15 \\ 14 \end{bmatrix}, \quad \max F(x) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \cdots \\ x_6 \end{bmatrix}$$

ili u skraćenom obliku: $$\sum_{j=1}^{3} a_j x_j + \sum_{i=4}^{6} A_i x_i = b$$

- *Formiranje početnog baznog rešenja* - Kako vektor rešenja može da ima najviše 3 pozitivne komponente to su x_1, x_2 i x_3 jednake nuli, pa sledi da je X=[0,0,0,x_4,x_5,x_6]. Sistem ograničenja se može prikazati kao:

$$\begin{bmatrix} 6 & 5 & 4 \\ 2 & 8 & 3 \\ 4 & 2 & 5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 18 \\ 15 \\ 14 \end{bmatrix}$$

Obzirom na to da je matrica A_i jedinična matrica, sledi da je x_4=18, x_5=15 i x_6=14. Vektor početnog rešenja je X=[0 0 0 18 15 14]. Vrednost početnog baznog rešenja je:

$$F(x) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 18 \\ 15 \\ 14 \end{bmatrix} = 0$$

- I *iteracija* - Početno rešenje je praktično bez značaja i potrebno je izvršiti njegovo poboljšanje uvođenjem u bazu vektora A_1, A_2 i A_3. Izbor vektora za ulazak u bazu vrši se prema kriterijumu:

$$\left\{ max \left| F_j - c_j \right|, (F_j - cj) < 0 \right\} \tag{2.8.12.}$$

Veličine F_j računaju se prema izrazu:

$$F_j = \sum_i c_i x_{ij}, \forall j \tag{2.8.13.}$$

Veličine x_{ij} koje predstavljaju koordinate vektora A_j u odnosu na datu bazu, računaju se pomoću izraza Y=$B^{-1}\cdot D$, tj.:

$$\begin{bmatrix} x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \\ x_{61} & x_{62} & x_{63} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 6 & 5 & 4 \\ 2 & 8 & 3 \\ 4 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 5 & 4 \\ 2 & 8 & 3 \\ 4 & 2 & 5 \end{bmatrix}$$

Vrednost F_j izračunava se na sledeći način.

$$\begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix} = \begin{bmatrix} c_4 & c_5 & c_6 \end{bmatrix} \cdot \begin{bmatrix} x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \\ x_{61} & x_{62} & x_{63} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 6 & 5 & 4 \\ 2 & 8 & 3 \\ 4 & 2 & 5 \end{bmatrix}$$

F_1=0, F_2=0, F_3=0. Izbor vektora za ulazak u bazu:

za j=1; F_1–c_1=0–1= –1
za j=2; F_2–c_2=0–1= –1
za j=3; F_3–c_3=0–1= –1

Za ulazak u bazu proizvoljno se bira vektor A_1.

Izbor vektora za izlazak iz baze:

$$\theta = \min_i \frac{x_i}{x_{ij}} > 0, \quad x_{ij} > 0 \tag{2.8.14.}$$

pri čemu je:

- i – oznaka vektora koji čine bazu
- j – oznaka vektora koji ulazi u bazu

U ovom slučaju j=1, i=4,5,6. θ={18/6, 15/2, 14/4}=3. Na osnovu toga sledi da bazu napušta vektor A_4. Novu bazu A' čine vektori A_1, A_5 i A_6.

Izražavanje vektora b u terminima nove baze [A_i']:

$$x_i' = \begin{Bmatrix} x_i - \theta \cdot x_{is} & za & i \neq r \\ \theta & za & i = r \end{Bmatrix} \tag{2.8.15.}$$

pri čemu je r - oznaka vektora koji napušta bazu. Na ovaj način se dobijaju komponente novog baznog rešenja X'. Tako je komponenta u odnosu na vektor koji je ušao u bazu (A_1) jednaka:

$$x_1' = \theta = x_4/x_{41} = 18/6 = 3.$$

Ostale koordinate vektora b, (odnosno komponente vektora rešenja X') u odnosu na odgovarajuće vektore nove baze takođe se računaju prema izrazu (2.8.15.):

$$x_5' = x_5 - \theta x_{51} = 15 - 3 \cdot 2 = 9$$
$$x_6' = x_6 - \theta x_{61} = 14 - 3 \cdot 4 = 2$$

Vektor rešenja X' nakon promene baze ima komponente (X je vektor-kolona) $X'=\{3\ 0\ 0\ 0\ 9\ 2\}$. Vrednost funkcije cilja sa novim vektorom rešenja X' je:

$$F'(x) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 9 \\ 2 \end{bmatrix} = 3$$

Dobijeno rešenje je bolje od početnog, ali se ne zna da li je i optimalno. Da bi to bilo utvrđeno, ponavlja se postupak izračunavanja prikazati kao linearna kombinacija vektora u novoj bazi A_1, A_5, A_6. Zatim se izračunavaju vrednosti F_j za vektore A_2, A_3 i A_4 i primenjuje se kriterijum (2.8.12.) za izbor vektora koji treba da uđu u bazu.

– II *iteracija* - Izračunavanje koeficijenta linearne kombinacije vektora A_j u odnosu na novu bazu $[A_i']$. Prema izrazu $Y=B^{-1}\cdot D$ sledi:

$$\begin{bmatrix} x_{14} & x_{12} & x_{13} \\ x_{54} & x_{52} & x_{53} \\ x_{64} & x_{62} & x_{63} \end{bmatrix} = \begin{bmatrix} 6 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 5 & 4 \\ 0 & 8 & 3 \\ 0 & 2 & 5 \end{bmatrix} = B^{-1} \cdot D \qquad (2.8.16.)$$

pri čemu je $B=[A_i']$.

Međutim u ovom koraku baza B ne daje jediničnu matricu, pa nije moguće izjednačavanje x_{ij} sa odgovarajućim elementima a_{ij} matrice $[A_i']=D$, kao što se to moglo uraditi u početnom rešenju. Prvo se izračunavaju vrednosti inverzne matrice B^{-1}, a zatim se na osnovu proizvoda $B^{-1}\cdot D$, na desnoj strani izraza (2.8.16.), dobijaju vrednosti odgovarajućih x_{ij}. Pošto je utvrđeno da je matrica B regularna kvadratna matrica može se izračunati njena inverzna matrica:

$$B^{-1} = \frac{1}{\det B} \cdot adjB$$

Nakon izračunavanja dobija se:

$$B^{-1} = \frac{1}{6} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 6 & 0 \\ -4 & 0 & 6 \end{bmatrix} = \begin{bmatrix} 1/6 & 0 & 0 \\ -1/3 & 1 & 0 \\ -2/3 & 0 & 1 \end{bmatrix}$$

Uvođenjem poznatih veličina u izraz (2.8.16.) i obavljanjem potrebnih radnji dobija se:

$$\begin{bmatrix} x_{14} & x_{12} & x_{13} \\ x_{54} & x_{52} & x_{53} \\ x_{64} & x_{62} & x_{63} \end{bmatrix} = \begin{bmatrix} 1/6 & 0 & 0 \\ -1/3 & 1 & 0 \\ -2/3 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 5 & 4 \\ 0 & 8 & 3 \\ 0 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 1/6 & 5/6 & 2/3 \\ -1/3 & 19/3 & 5/3 \\ -2/3 & -4/3 & 7/3 \end{bmatrix}$$

Zatim se računaju veličine F_j:

$$\begin{bmatrix}F_4 & F_2 & F_3\end{bmatrix}=\begin{bmatrix}c_1 & c_5 & c_6\end{bmatrix}\cdot\begin{bmatrix}x_{14} & x_{12} & x_{13}\\ x_{54} & x_{52} & x_{53}\\ x_{64} & x_{62} & x_{63}\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}1/6 & 5/6 & 2/3\\ -1/3 & 19/3 & 5/3\\ -2/3 & -4/3 & 7/3\end{bmatrix}=\begin{bmatrix}\frac{1}{6} & \frac{5}{6} & \frac{2}{3}\end{bmatrix}$$

odnosno F_4=1/6, F_2=5/6, F_3=2/3.

Izbor vektora za ulazak u bazu prema kriterijumu (2.8.12.):

F_4–c_4=1/6–0= 1/6
F_2–c_2=5/6–1= –1/6
F_3–c_3=2/3–1= –1/3

U bazu ulazi vektor A_3. Vektor koji izlazi iz baze određuje se na osnovu kriterijuma (2.8.14.), tj: θ =min{x_1/x_{13}, x_5/x_{53}, x_6/x_{63}}, odnosno θ =min{9/2, 27/5, 6/7}. Iz baze izlazi vektor A_6. Novu bazu [A_i''] čine vektori A_1, A_5, A_3. Izražavanje vektora b u terminima nove baze[A_i'']:

$x_3''=\theta=x_6/x_{63}=6/7$
$x_1''=x_1-\theta x_{13}=17/7$
$x_5''=x_5-\theta x_{53}=53/7$

Novo bazno rešenje je: X''={17/7 0 6/7 0 53/7 0}. Vrednost funkcije cilja za novo bazno rešenje je:

$$F''(x)=C\cdot X''=\begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}17/7\\ 0\\ 6/7\\ 0\\ 53/7\\ 0\end{bmatrix}=23/7$$

– Sprovođenjem još jednog kompletnog postupka poboljšanja dobija se optimalno rešenje: X'''={7/6 53/51 74/51 0 0 0}, a vrednost funkcije cilja za optimalno rešenje je max$F(x)$=373/102.

2.8.4. Tabelarni postupak – simpleks tabela

Simpleks tabela predstavlja jedan od algoritama simpleks metode. Nalaženje optimalnog rešenja, zadatka linearnog programiranja, znatno je pojednostavljeno nakon što su Čarns, Kuper i Henderson prikazali formalizovan tabelarni postupak. Logika simpleks metode je ostala nepromenjena. Postupak se sastoji u tome da se od koeficijcnata funkcije cilja i sistema ograničenja sastavi tabela (otuda naziv simpleks tabela), koja se zatim, prema određenim pravilima, menja dok se ne dođe do optimalnog rešenja. Optimalno rešenje se dobija postupno pošto se sačini dovoljan broj simpleks tabela. U proračunima se primenjuju isti kriterijumi ali je postupak znatno brži i jednostavniji korišćenjem simpleks tabele. Početna simpleks tabela ima izgled kao što je prikazano u tabeli 2.8.1.

Tabela 2.8.1. *Početna simpleks tabela ST_0 (Dantzig, 1963)*

Bazne promenljive			**Slobodne promenljive**								$\boldsymbol{\theta = b_i/a_{ij}}$
C			c_1	c_2	**...**	c_n	c_{n+1}	c_{n+2}	**...**	c_{n+m}	
C_b	X_b	B	x_1	x_2	**...**	x_n	x_{n+1}	x_{n+2}	**...**	x_{n+m}	
c_{n+1}	x_{n+1}	b_1	a_{11}	a_{12}	...	a_{1n}	1	0	...	0	θ_1
c_{n+2}	x_{n+2}	b_2	a_{21}	a_{22}	...	a_{2n}	0	1	...	0	θ_2
...	**...**	**...**	...	...	...	...	0	0	...	0	...
c_{n+m}	x_{n+m}	b_m	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	θ_m
F_j-c_j		**0**	F_1-c_1	F_2-c_2	**...**	F_n-c_n	**0**	**0**	**0**	**0**	$\lvert F_j-c_j\rvert\cdot\theta$

Oznake u simpleks tabeli su:

- C – vektor koeficijenata c_i uz promenljive x_i funkcije cilja, i=1,2,...,n.
- C_b – vektor koeficijenata c_i uz promenljive x_i funkcije cilja koje sačinjavaju bazno dopustivo rešenje, i=n+1, n+2,...,m.
- X_b – vektor promenljivih bazno dopustivog rešenja,
- B – vektor vrednosti promenljivih bazno dopustivog rešenja za posmatranu iteraciju,
- X_j – množitelji vektora baze,
- θ – kriterijum za određivanje koji vektor X_b izlazi iz baznog rešenja,
- F_j-C_j – kriterijum optimalnosti,
- $|F_j-c_j|\cdot\theta$ – priraštaj funkcije cilja za posmatranu iteraciju.

2.8.4.1. Rešavanje problema maksimuma

U narednim primerima biće ilustrovano formiranje početne simpleks tabele (ST_0), iterativni postupak dolaženja do optimalnog rešenja, kao i situacije koje bi mogle da se dogode prilikom rešavanja, za slučaj maksimiziranja funkcije cilja.

Primer 2.8.4. Pronaći maksimalnu vrednost funkcije cilja za dati matematički model.

$$
\begin{aligned}
& F(x) = 6x_1 + 7x_2 + 8x_3 \\
\text{p.o.}\quad & x_1 + 2x_2 + x_3 \leq 900 \\
& x_1 + 3x_2 + 2x_3 \leq 1600 \\
& 2x_1 \leq x_2 + x_3 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}
$$

Rešenje. Primena bilo kog algoritma simpleks metode zahteva odgovarajuće prilagođavanje modela kako bi se on rešio. Prilagođavanje modela vrši se prevođenjem svih nejednačina u jednačine, uvođenjem *dopunskih* (izravnavajućih) promenljivih, i proširenje funkcije cilja. Kod prethodnog sistema ograničenja, leve strane nejednačina su manje od desnih pa se dodavanjem izravnavajućih promenljivih levoj strani nejednačinama dobijaju jednačine. izravnavajuće promenljive imaju svoj ekonomski smisao i one označavaju neiskorišćene kapacitete, koji su opisani konkretnim ograničenjima. Logično je da, u proširenoj funkciji cilja, uz dopunske promenljive budu koeficijenti jednaki nuli. Prilagođen model, za primenu simpleks tabele, dobija oblik:

$$\max F(x) = 6x_1 + 7x_2 + 8x_3 + 0x_4 + 0x_5 + 0x_6$$

$$\begin{aligned} \text{p.o.}\quad & x_1 + 2x_2 + x_3 + x_4 = 900 \\ & x_1 + 3x_2 + 2x_3 + x_5 = 1600 \\ & 2x_1 - x_2 - x_3 + x_6 = 0 \\ & x_1,x_2,x_3,x_4,x_5,x_6 \geq 0 \end{aligned}$$

Potrebno je istaći dve bitne karakteristke postupka prilagođavanja modela:

1. Linearni model je prilagođen za primenu simpleks tabele (i ostalih algoritama) kada su sve nejednačine prevedene u jednačine, pri čemu se svakoj nejednačini dodaje po jedna izravnavajuća promenljiva; i
2. Kada se u tako prilagođenom modelu može formirati jedinična matrica (od koeficijenata koji stoje uz izravnavajuce promenljive).

Koeficijente iz prilagođenog modela unosimo u početnu simpleks tabelu. Dimenzije simpleks tabele zavise od dimenzije prilagođenog modela. Tako simpleks tabela ima: tri stalne kolone (C_b, X_b i B) i još onoliko kolona koliko promenljivih ima u modelu. Posmatrano po redovima, simpleks tabela ima: zaglavlje (u koje se unose koeficijenti iz funkcije cilja); onoliko redova koliko sistem ograničenja ima jednačina; i dodatni red optimalnosti (F_j–c_j). Nakon ovih objašnjenja, početna (nulta) simpleks tabela ima oblik kao što je prikazano u tabeli 2.8.2.

Tabela 2.8.2. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	6	7	8	0	0	0	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	
0	x_4	**900**	1	2	1	1	0	0	900
0	x_5	**1600**	1	3	**2**	0	1	0	800
0	x_6	**0**	2	-1	-1	0	0	1	/
F_j–c_j		**0**	**-6**	**-7**	**-8**	**0**	**0**	**0**	**6.400**

- **Popunjavanje početne simpleks tabele *ST*_0**

Prvi red zaglavlja sadrži koeficijente koji stoje uz promenljive u funkciji cilja. Ostali redovi odgovaraju jednačinama sistema ograničenja i sadrže koeficijente iz tog sistema, stim što su slobodni članovi sistema ograničenja upisani u kolonu *B*. Kolona *B* sadrži vrednosti koje čine rešenje u posmatranoj iteraciji. U nultoj simpleks tabeli (*ST*_0) kolona X_b je popunjena na osnovu promenljivih koje odgovaraju jediničnim kolanama (vektorima). U koloni C_b upisuju se kaeficijenti koji stoje u funkciji cilja uz promenljive iz kolone X_b.

Red (F_j–c_j) ima pasebnu ulagu u postupku rešavanja problema i on se izračunava tako što se koeficijenti iz kolone C_b pomnože odgovarajućim koeficijentima određene kolone X_j, pa se proizvodi saberu i od zbira aduzme kaeficijent iz zaglavalja c_j, kao što je prikazano relacijom (2.8.17.).

$$F_j–c_j = C_b \cdot X_j - c_j, \quad \forall\, j \tag{2.8.17.}$$

Tako je na primer, za kalonu x_1 izračunato:

$$F_1–c_1 = (0 \cdot 1 + 0 \cdot 1 + 0 \cdot 2) - 6 = -6$$

Istim proračunom se popunjavaju i preostala polja u redu za optimalnost. Sastavljanjem početne simpleks tabele počinje proces rešavanja problema. Ova tabela sadrži jedno maguće rešenje problema. Kolona X_b u simpleks tabeli sadrži promenljive koje čine rešenje problema za tu iteraciju, a kolona *B* sadrži vrednosti tih promenljivih. Prema tome, rešenje problema u nultoj iteraciji čine promenljive:

$$x_4 = 900, \;\; x_5 = 1600, \;\; x_6 = 0$$

Vrednost funkcije cilja za ovo rešenje nalazi se u preseku reda (F_j–c_j) i kolone *B*. Dobija se proračunom tako što se koeficijenti iz kolone C_b pomnože odgovarajućim koeficijentima kolone *B*, pa se proizvodi saberu. U nultoj iteraciji funkcija cilja iznosi:

$$F_0 = 0 \cdot 900 + 0 \cdot 1.600 + 0 \cdot 0 = 0$$

Otuda i potiče naziv nulta simpleks tabela i simbolički se predstavlja kao *ST*_0.

- **Ocena optimalnosti pronađenog rešenja za ST_0**

Potrebno je proveviti da li je rešenje iz početne (nulte) simpleks tabele optimalno ili ne. To se formuliše preko reda $F_j – c_j$. Između vrednosti funkcija cilja za dve uzastopne iteracije postoji veza koja se može izraziti sledećom relacijom:

$$F_1 = F_0 - \theta \cdot (F_j - c_j) \qquad (2.8.18.)$$

U ovoj relaciji je sa F_1 označena vrednost funkcije cilja u narednoj iteraciji, sa F_0 vrednost funkcije cilja u posmatranoj iteraciji, a sa θ vrednost promenljive koja ulazi u naredno rešenje. Kako je uvek $\theta > 0$, to će vrednost funkcije kriterijuma u narednoj iteraciji zavisiti od vrednosti i znaka $(F_j – c_j)$. Kada tražimo maksimalnu vrednost funkcije kriterijuma, promena nekog rešenja ima smisla samo ako se time povećava vrednost funkcije cilja, odnosno ako je $F_1 > F_0$. Prema relaciji (2.8.18.) lako se može zaključiti da će se ovo desiti uvek kada je $(F_j – c_j) < 0$. Na osnovu ovoga se može formulisati kriterijum za ocenu optimalnosti pronađenog rešenja:

Kada se traži maksimalna vrednost funkcije cilja, sve dok u redu ($F_j – c_j$) postoje negativni koeficifenti nije pronađeno optimalno rešenje problema.

U *ST*_0 u kolonama x_1, x_2 i x_3 postoje negativni koeficijenti u redu $(F_j – c_j)$, pa zaključujemo da rešenje iz ove tabele nije optimalno.

- **Izbor promenljive koja ulazi u naredno rešenje**

Pošto u početnoj tabeli nije pronađeno optimalno rešenje, treba odrediti novo rešenje koje će obezbediti veću vrednost funkcije kriterijuma u odnosu na vrednost koju je dalo rešenje iz posmatrane tabele. Promena rešenja se vrši postupno, korak po korak, tako što se jedna od promenljivih izvan rešenja odredi da uđe u naredno rešenje, a jedna promenljiva izlazi iz rešenja. U početnoj simpleks tabeli izvan rešenja se nalaze promenljive x_1, x_2 i x_3. Prema tome, u narednoj iteraciji u rešenje može da uđe jedna od ovih promenljivih, a da izađe jedna od promenljivih x_4, x_5 ili x_6. Cilj je da se što pre dođe do optimalnog rešenja, pa u naredno rešenje treba da uđe ona promenljiva koja obezbeđuje najveći porast (priraštaj) funcije cilja. Na osnovu relacije (2.8.18.) formuliše se kriterijum za izbor promenljive koja ulazi u naredno bazno moguće rešenje.

Ukoliko se u problemu traži maksimalna vrednost funkcije cilja, u naredno rešenje treba da uđe ona promenljiva kojoj u simpleks tabeli odgovara:

$$\max_j \left\{ \left| (F_j - c_j) < 0 \right| \right\} = F_s - c_s \qquad (2.8.19.)$$

Postavlja se pitanje: kako izvršiti izbor koja će promenljiva da uđe u bazu kada se dogodi situacija da dve ili više promenljivih izpunjava uslov (2.8.19.). Na osnovu relacije (2.8.17.), može se zaključiti da je, u tom slučaju, potrebno uzeti u obzir i vrednost parametra θ, odnosno vrednost promenljive koja ulazi u naredno rešenje. Dakle, ukoliko je za dve, ili više promenljivih ispunjen uslov iz relacije (2.8.19.), onda u naredno rešenje treba da uđe ona promenljiva kojoj odgovara:

$$\max_j \left\{ \theta_j \cdot \left| F_j - c_j \right| \right\} = \theta_s \cdot (F_s - c_s) \qquad (2.8.20.)$$

U početnoj (nultoj) *ST*_0 najveću apsolutnu vrednost od svih negativnih koeficijenata $(F_j – c_j)$ ima koeficijent koji odgovara promenljivoj x_3. Prema relaciji (2.8.19.) dobija se:

$$\max_j \left\{ F_j - c_j \right| < 0 \} = \max_j \left| -6, -7, -8 \right| = F_3 - c_3 = -8$$

na osnovu čega se određuje da u naredno rešenje ulazi promenljiva x_3.

- **Određivanje promenljive koja će da izađe iz rešenja**

Pošto je određeno da promenljiva x_3 uđe u naredno rešenje, potrebno je odrediti koja će promenljiva od x_4,x_5 i x_6 da izađe iz rešenja. Ovaj kriterijum određen je na osnovu relacije:

$$\theta = \min_i \frac{b_i}{a_{ij}}, \quad a_{ij} > 0 \tag{2.8.21.}$$

Relacija (2.8.21.) može se pojednostavljeno formulisati i objasniti na sledeći način:

- označi se kolona simpleks tabele koja odgovara promenljivoj koja ulazi u naredno rešenje. U problemu koji rešavamo to je kolona x_3,
- podele se odgovarajući koeficijenti iz kolone B pozitivnim koeficijentima iz označene kolone i utvrde se svi količnici. Za ST_0 ti količnici su:

$$\frac{900}{1} = 900, \quad \frac{1600}{2} = 800$$

u trećem redu treba deliti sa (-1), pa se taj količnik ne uzima u obzir,

- iz rešenja izlazi ona promenljiva kojoj odgovara najmanji količnik odnosno:

$$\theta = \min\,(900, 800) = 800$$

Na osnovu ovog kriterijuma, iz rešenja izlazi promenljiva x_5, jer njoj odgovara najmanji količnik.

- **Izračunavanje elemenata naredne simpleks tabele**

Promenljiva x_3 ulazi u naredno rešenje pa se zato ta kolona, u ST_0, šrafira (osenča). To je tzv. vodeća ili *karakteristična kolona*. Iz rešenja izlazi promenljiva x_5. Promenljiva x_5 nalazi se u drugom redu nulte simpleks tabele, pa je u tabeli šrafiran i drugi red. To je tzv. vodeći ili *karakteristični red*. U preseku karakterističnog reda i karakteristične kolone nalazi se koeficijenat a_{rs}. Njegov uobičajen naziv je *karakteristični koeficijent*. U simpleks tabeli ST_0 karakteristični koeficijent je a_{23}=2, i posebno je naglašen (tamnije osenčan).

Na ovaj način je početna tabela prilagođena da se na osnovu nje izračunaju elementi za narednu simpleks tabelu, tj. ST_1. Prema određenim pravilima, na osnovu pripremljene početne simpleks tabele, izračunavaju se vrednosti svih promenljivih za novo rešenje, vrednost funkcije kriterijuma i vrednost ostalih koeficijenata za narednu ST_1 tabelu. Za proračun elemenata naredne simpleks tabele posebno je važan koeficijenat a_{rs}. Postupak izračunavanja koeficijenata određen je relacijama:

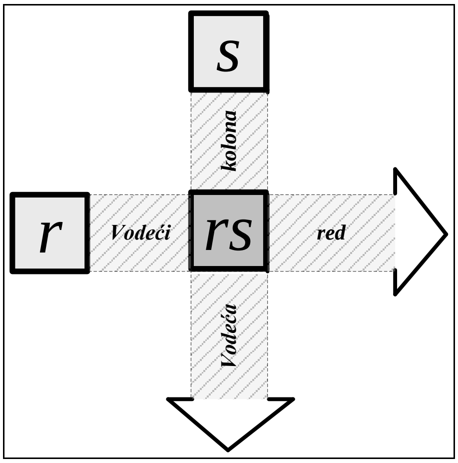
$$b_r' = \frac{b_r}{a_{rs}}, \quad i = r \tag{2.8.22.}$$

$$b_i' = b_i - \frac{b_r}{a_{rs}} \cdot a_{is}, \quad i \neq r \tag{2.8.23.}$$

$$a_{rj}' = \frac{a_{rj}}{a_{rs}}, \quad i = r \tag{2.8.24.}$$

$$a_{ij}' = a_{ij} - \frac{a_{rj}}{a_{rs}} \cdot a_{is}, \quad i \neq r \tag{2.8.25.}$$

Indeks (s) odgovara promenljivoj koja ulazi u naredno rešenje, tj. to je vodeća (karakteristična) kolona, a indeks (r) promenljivoj koja treba da izađe iz rešenja, tj. to je vodeći (karakteristični) red, kao što je prikazano na slici 2.8.4.



Slika 2.8.4. *Značenje indeksa (s) i (r) (Dantzig, 1963)*

Relacije (2.8.22.) – (2.8.25.) mogu se pojednostavljeno objasniti na sledeći način:

- svi koeficijenti iz šrafiranog reda podele se karakterističnim koeficijentom [relacije (2.8.22.) i (2.8.24.)],
- koeficijent za *i*-ti red i *j*-tu kolonu, naredne simpleks tabele, izračunava se tako što se od odgovarajućeg koeficijenta iz prethodne simpleks tabele oduzme proizvod koeficijenta iz *i*-tog reda šrafirane kolone i koeficijenta iz *j*-te kolone šrafiranog reda (koji je već podeljen karakterističnim koeficijentom), [relacije (2.8.23.) i (2.8.25.)].

Proračun elemenata za narednu simpleks tabelu prikazan je šematski na slici 2.8.2. Na osnovu pripremljene početne simpleks tabele *ST*_0 i napred definisanih pravila, naredna simpleks tabela *ST*_1 se dobija na sledeći način:

- u kolonu X_b unose se promenljive koje čine rešenje u prvoj iteraciji: x_4,x_3, i x_6;
- u kolonu C_b upisuju se koeficijenti koji u funkciji cilja (ili u zaglavlju tabele) stoje uz promenljive iz kolone X_b, a to su: 0, 8 i 0;
- koeficijenti iz šrafiranog reda, pripremljene tabele, dele se sa karakterističnim koeficijentom, tj. sa 2, a rezultati se upisuju u narednu tabelu *ST*_1 u red u kome se nalazi nova promenljiva x_3. To su koeficijenti: 800, 1/2, 3/2, 1, 0, 1/2, 0;
- svi ostali koeficijenti se izračunavaju pomoću relacija (2.8.23.) i (2.8.25.), i to:

Kolona *B*:

prvi red: 900 – (1.600/2)·1 = 100
treći red: 0 – (1.600/2)·(–1) = 800
red (F_j–c_j): 0 – (1.600/2)·(–8) = 6.400

Kolona x_1:

prvi red: 1 – (1/2)·1 = 1/2
treći red: 2 – (1/2)·(–1) = 5/2
red (F_j–c_j): (–6) – (1/2)·(–8) = –2

Kolona x_2:

prvi red: 2 – (3/2)·1 = 1/2
treći red: (–1) – (3/2)·(–1) = 1/2
red (F_j–c_j): (–7) – (3/2)·(–8) = 5

Kolona x_3:

– odgovara promenljivoj koja je ušla u rešenje. U drugom redu ove kolone već je upisana jedinica, a svi ostali koeficijenti iz ove kolone jednaki su nuli. Tako će biti u svim simpleks tabelama: promenljive iz kolone x_b imaće u preseku reda (u kome se nalaze) i odgovarajuće kolone jedinice, a sve ostale koeficijente nule.

Kolone x_4 i x_6:

– u šrafiranom redu prethodne tabele imaju nule. Zbog toga u ovim kolonama neće doći do transformacije keoficijenata, pa ih prepisujemo. Slično bi se dogodilo kada u šrafiranoj koloni tabele postoji nula, tada se neće menjati koeficijenti iz reda u kome se nalazi ta nula, pa se taj red prepisuje iz prethodne tabele.

Kolona x_5:

prvi red: $0 - (1/2)\cdot 1 = -1/2$
treći red: $0 - (1/2)\cdot(-1) = 1/2$
red (F_j–C_j): $0 - (1/2)\cdot(-8) = 4$

Na osnovu dobijenih rezultata sastavlja se simpleks tabela nakon prve iteracije ST_1, kao što je prikazano u tabeli 2.8.3.

Tabela 2.8.3. *Simpleks tabela nakon prve iteracije ST_1*

C_b	X_b	B	6	7	8	0	0	0	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	
0	x_4	**100**	**1/2**	1/2	0	1	–1/2	0	200
8	x_3	**800**	1/2	3/2	1	0	1/2	0	1.600
0	x_6	**800**	5/2	1/2	0	0	1/2	1	320
$F_j - c_j$		**6400**	**–2**	**5**	**0**	**0**	**4**	**0**	**400**

- **Ocena optimalnosti rešenja iz *ST*_1**

Očigledno da rešenje posle prve iteracije nije optimalno, jer se u redu (F_j–c_j) u koloni x_1 nalazi negativni koeficijent, tj. $F_1 - c_1 = -2$.

- **Izbor promenljive koja ulazi u naredno rešenje**

Pošto u ST_1 postoji samo jedan negativni koeficijenat, određujemo da promenljiva x_1 (kojoj odgovara taj koeficijenat) uđe u naredno rešenje. Zbog toga je u ST_1 osenčana kolona x_1.

- **Određivanje promenljive koja izlazi iz rešenja**

Prema relaciji (2.8.20.), podele se odgovarajući koeficijenti iz kolone B sa pozitivnim koeficijentima osenčane kolone x_1 i dobija:

$$\frac{100}{0{,}5} = 200, \quad \frac{800}{0{,}5} = 1.600, \quad \frac{800}{2{,}5} = 320$$

odakle se određuje da je: $\theta = \min(200, 1.600, 320) = 200$.

Najmanji količnik (200) odgovara promenljivoj x_4, pa iz rešenja izlazi promenljiva x_4. U ST_1 osenčan je red u kome je promenljiva x_4. U preseku karakteristične kolone i karakterističnog reda nalazi se karakteristični koeficijent a_{11}, koji iznosi a_{11}=1/2. Na ovaj način tabela ST_1 je pripremljena da se na osnovu nje izračunaju elementi za narednu simpleks tabelu ST_2. Ona se dobija na sledeći način:

– u kolonu x_b unose se promenljive koje čine rešenje u drugoj iteraciji: x_1, x_3 i x_6;
– u kolonu C_b unose se koeficijenti iz zaglavlja koji stoje uz promenljive iz kolone x_b, a to su: 6, 8, 0;

- koeficijenti iz šrafiranog reda *ST*_1 dele se sa karakterističnim koeficijentom, tj. a_{11}=1/2 i upisuju u prvi red *ST*_2, a to su: 200, 1, 1, 0, 2, -1, 0.
- ostali koeficijenti se izračunavaju na sledeći način:

Kolona *B*:

drugi red: 800 – (100/0,5)·0,5 = 700
treći red: 800 – (100/0,5)·2,5 = 300
red (F_j–c_j): 6.400 – (100/0,5)·(–2) = 6 800

Kolona x_1: ima jedinicu u prvom redu, a ostali koeficijenti su nule.

Kolona x_2:

drugi red: 1,5 – (0,5/0,5)·0,5 = 1
treći red: 0,5 – (0,5/0,5)·2,5 = –2
red (F_j–c_j): 5 – (0,5/0,5)·(–2) = 7

Kolone x_3 i x_6: imaju nule u osenčanom redu, pa njihove koeficijnte prepisujemo.

Kolona x_4:

drugi red: 0 – (1/0,5)·0,5 = –1
treći red: 0 – (1/0,5)·2,5 = –5
red (F_j–c_j): 0 – (1/0,5)·(–2) = 4

Kolona x_5:

drugi red: 0,5 – (–0,5/0,5)·0,5 = 1
treći red: 0,5 – (–0,5/0,5)·2,5 = 3
red (F_j–c_j): 4 – (–0,5/0,5)·(–2) = 2

Na osnovu ovih rezultata popunjava se tabela nakon druge iteracije (*ST*_2), tj. tabela 2.8.4.

Tabela 2.8.4. *Simpleks tabela nakon druge iteracije ST_2*

C_b	X_b	B	6	7	8	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6
6	x_1	**200**	**1**	1	0	2	–1	0
8	x_3	**700**	0	1	1	–1	1	0
0	x_6	**300**	0	–2	0	–5	3	1
F_j–c_j		**6 800**	**0**	**7**	**0**	**4**	**2**	**0**

U redu za optimalnost (F_j–c_j) u *ST*_2 nema negativnih koeficijenata, pa se zaključuje da je u ovoj tabeli pronađeno optimalno rešenje problema. Njega čine promenljive:

$$x_1 = 200, \quad x_3 = 700, \quad x_6 = 300$$

Optimalno rešenje obezbeđuje maksimalnu vrednost funkcije cilja:

$$\max F(x) = 6\,800$$

Napomena 2.8.1.: Razlike F_j–c_j su utvrđene na isti način kao i drugi koeficijenti u tabeli. Koeficijenti F_j–c_j mogu biti utvrđeni i kao zbir proizvoda između koeficijenata kolone C_b i koeficijenata odgovarajuće kolone umanjen za koeficijent iz zaglavlja (taj postupak je rađen za proračun F_j–c_j koeficijenata za *ST*_0). Ovo ukazuje na mogućnost kontrole tačnosti rešavanja modela.

Napomena 2.8.2.: Rešenje u svakoj iteraciji predstavlja bazno moguće rešenje. To znači da rešenje mora zadovoljavati ograničavajuće faktore. Prilikom rešavanja modela poželjno je proveriti da li rešenje u svakoj iteraciji zadovoljava ograničavajuće faktore, tj. da li predstavlja moguće rešenje, jer u suprotnom radi se o grešci koja bi se iz iteracije u iteraciju nagomilavala.Ukoliko se dogodi greška dobilo bi se rešenje koje nije optimalno, ili nemoguće ukoliko ne zadovoljava ograničavajuće faktore.

Primer 2.8.5. Pronaći maksimalnu vrednost funkcije cilja za dati matematički model.

$$
\begin{array}{ll}
 & F(x) = 80x_1 + 50x_2 + 40x_3 + 20x_4 \\
\text{p.o.} & 8x_1 + 6x_2 + 4x_3 + 2x_4 \leq 400 \\
 & 1x_1 + 1x_2 + 1x_3 + 1x_4 = 100 \\
 & 8x_1 + 4x_2 + 4x_3 \qquad \geq 300 \\
 & x_j \geq 0,\ j=1,2,3,4
\end{array}
$$

Rešenje. Uvođenjem izravnavajućih (dopunskih) promenljivih matematički model postaje:

$$
\begin{array}{ll}
 & \max F(x) = 80x_1 + 50x_2 + 40x_3 + 20x_4 + 0x_5 + 0x_6 \\
\text{p.o.} & 8x_1 + 6x_2 + 4x_3 + 2x_4 + x_5 \qquad = 400 \\
 & 1x_1 + 1x_2 + 1x_3 + 1x_4 \qquad = 100 \\
 & 8x_1 + 4x_2 + 4x_3 \qquad - x_6 = 300 \\
 & x_j \geq 0,\ j=1,2,...,6
\end{array}
$$

Uočava se da u modelima koji sadrže sva tri tipa ograničenja ($\leq$, $=$, $\geq$) uvođenje samo izravnavajućih promenljivih ne obezbeđuje formiranje jedinične matrice u prilagođenom modelu. U predhodnom primeru je naglašeno da je formiranje jedinične matrice druga bitna karakteristika postupka prilagođavanja modela i da se na osnovu nje dobija početno bazično rešenje modela. Zato se, u nejednačinama tipa $\geq$ i jednačinama uvode *veštačke* promenljive. Na taj način se obezbeđuje formiranje jedinične matrice i na osnovu nje početno rešenje.

Veštačke promenljive nemaju ekonomsko značenje već kao kalkulativno sredstvo treba da olakšaju postupak pronalaženja optimalnog rešenja. One se, zbog toga, ne mogu pojaviti u optimalnom rešenju. Da bi se matematičkom procedurom obezbedilo ispadanje veštačkih promenljivih iz bazičnog rešenja, one se uvode u funkciju cilja sa koeficijentom $-M$, gde je M proizvoljno veliki pozitivni broj. tj. $M >> 0$. Pošto se traži maksimalna vrednost funkcije cilja, tj. max $F(x)$, ne može se nikada postići njena maksimalna vrednost dok su veštačke promenljive pozitivne, tj. dok se nalaze u rešenju.

Nakon uvođenja i veštačkih promenljivih matematički model LP, potpuno prilagođen za rešavanje, ima sledeći oblik:

$$
\begin{array}{ll}
 & \max F(x) = 80x_1 + 50x_2 + 40x_3 + 20x_4 + 0x_5 + 0x_6 - M \cdot x_7 - M \cdot x_8 \\
\text{p.o.} & 8x_1 + 6x_2 + 4x_3 + 2x_4 + x_5 \qquad = 400 \\
 & 1x_1 + 1x_2 + 1x_3 + 1x_4 \qquad + x_7 \qquad = 100 \\
 & 8x_1 + 4x_2 + 4x_3 \qquad - x_6 \qquad + x_8 = 300 \\
 & x_j \geq 0,\ j=1,2,...,8
\end{array}
$$

Prilagođavanje modela uvođenjem izravnavajućih i veštačkih promenljivih prikazano je u tabeli 2.4.1. u poglavlju 2.4.2.3.

Napomena 2.8.3.: Redosled uvođenja promenljivih u model ne utiče na dalji postupak i konačan rezultat. Međutim, kako se na mestu na kome se nalazi jedinična matrica u prvoj iteraciji u ostalim iteracijama nalazi inverzna matrica bazičnih vektora, kao značajan element za sprovođenje analize osetljivosti optimalnog rešenja na promene pojedinih elemenata modela, poželjno je da se model prilagodi tako da se dobije sređena jedinična matrica. Zbog toga je uobičajeno da se prilagođavanje vrši na sledeći način: najpre se u nejednačinama tipa $\geq$ oduzmu izravnavajuće promenljive, a zatim redom se dodavanjem izravnavajućih ili veštačkih promenljivih prilagođavaju sva ograničenja čime se obezbeđuje dobijanje sređene jedinične matrice. U ovom primeru prilagođavanje modela nije izvršeno na opisan način.

Nakon unosa svih koeficijenata iz prilagođenog modela, početna (nulta) simpleks tabela, tj. ST_0, dobija oblik kao što je prikazano u tabeli 2.8.5.

Tabela 2.8.5. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	**80**	**50**	**40**	**20**	**0**	**0**	$-M$	$-M$	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
0	x_5	**400**	8	6	4	2	1	0	0	0	50
$-M$	x_7	**100**	1	1	1	1	0	0	1	0	100
$-M$	x_8	**300**	**8**	4	4	0	0	-1	0	1	37,5
$F_j–c_j$		**0**	-80	-50	-40	-20	0	0	0	0	**3000**
		-400	-9	-5	-5	-1	0	1	0	0	**337,5**

Koeficijenti u redu ($F_j–c_j$) izračunati su na uobičajeni način, pa su podeljeni u dva dela i napisani u dva reda. U prvi red unose se koeficijenti uz koje se ne javlja M, a u drugi koeficijenti uz koje se javlja M. Tako je, koeficijent $F_1–c_1 = –80–9M$ dobijen množenjem koeficijenata iz kolone C_b i odgovarajućih koeficijenata iz kolone x_1, proizvodi su sabrani i od zbira oduzet koeficijent iz zaglavlja c_1, tj.:

$$[0 \cdot 8 + (-M) \cdot 1 + (-M) \cdot 8] – 80 = – 9M – 80$$

Podela koeficijenata iz reda ($F_j–c_j$) na dva dela izvršena je samo iz praktičnih razloga. Kako je $M>>0$, to će, sve dok su veštačke promenljive u rešenju, drugi red određivati koja promenljiva ulazi u naredno rešenje. Rešenje u nultoj simpleks tabeli nije optimalno jer postoje negativne razlike $F_j–c_j$. Posmatraju se najpre razlike $F_j–c_j$ u drugom redu, pa kada među njima nema više negativnih prelazi se na prvi red razlika $F_j–c_j$.

U naredno rešenje ulazi promenljiva x_1 zato što je razlika ($F_1–c_1$) negativna i najveća po apsolutnoj vrednosti. Osenčana kolonu x_1 postaje karakteristična kolona. Nakon deljenja koeficijenata iz kolone B sa koeficijentima iz osenčane kolone, zaključuje se da iz rešenja treba da izađe promenljiva x_8, jer njoj odgovara najmanji količnik 300/8 = 37,5. Osenčani red x_8 postaje karekteristični red. Na uobičajeni način izračunavaju se koeficijente za simpleks tabelu ST_1, koja je prikazana u tabeli 2.8.6.

Tabela 2.8.6. *Simpleks tabela nakon prve iteracije ST_1*

C_b	X_b	B	**80**	**50**	**40**	**20**	**0**	**0**	$-M$	$-M$	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
0	x_5	**100**	0	2	0	**2**	1	1	0	-1	50
$-M$	x_7	**125/2**	0	1/2	1/2	1	0	1/8	1	-1/8	62,5
80	x_1	**75/2**	1	1/2	1/2	0	0	-1/8	0	1/8	/
$F_j–c_j$		**3000**	0	-10	0	-20	0	-10	0	10	**1000**
		-62,5	0	-1/2	-1/2	-1	0	-1/8	0	9/8	**50**

Nakon preračunavanja koeficijenata, utvrđeno je da ni u ST_1 nije pronađeno optimalno rešenje. U naredno rešenje ulazi promenljiva x_4 (osenčana kolona), a iz rešenja izlazi promenljiva x_5 (osenčani red). Simpleks tabela ST_2, prikazana je u tabeli 2.8.7.

Tabela 2.8.7. *Simpleks tabela nakon druge iteracije ST_2*

C_b	X_b	B	**80**	**50**	**40**	**20**	**0**	**0**	$-M$	$-M$	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
20	x_4	**50**	0	1	0	1	1/2	1/2	0	-1/2	/
$-M$	x_7	**25/2**	0	-1/2	**1/2**	0	-1/2	-3/8	1	3/8	25
80	x_1	**75/2**	1	1/2	1/2	0	0	-1/8	0	1/8	75
$F_j–c_j$		**4000**	0	10	0	0	10	0	0	0	**0**
		-12,5	0	1/2	-1/2	0	1/2	3/8	0	5/8	**12,5**

Rešenje nakon druge iteracije, takođe, nije optimalno. Veštačka promenljiva x_7 još uvek je u rešenju i ima pozitivnu vrednost, a i koeficijent (F_3–c_3) je negativan. U naredno rešenje ulazi promenljiva x_3, a iz rešenja izlazi promenljiva x_7. Sastavljena je simpleks tabela *ST*_3, koja je prikazana u tabeli 2.8.8.

Tabela 2.8.8. *Simpleks tabela nakon treće iteracije ST_3*

C_b	X_b	B	80	50	40	20	0	0	$-M$	$-M$	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
20	x_4	**50**	0	1	0	1	1/2	1/2	0	-1/2	100
40	x_3	**25**	0	-1	1	0	-1	-3/4	2	3/4	/
80	x_1	**25**	1	1	0	0	1/2	1/4	-1	0	100
F_j–c_j		**4000**	0	10	0	0	10	0	0	0	**0**
		0	0	0	0	0	0	0	1	1	**0**

U simpleks tabeli *ST*_3 dobijeno je optimalno rešenje. Njega čine promenljive:

$$x_1 = 25,\ \ x_3 = 25,\ \ x_4 = 50$$

a maksimalna vrednost funkcije cilja iznosi: max $F(x) = 4000$.

Bez obzira što je, nakon treće iteracije, pronađeno optimalno rešenje, uočava se da je za nebazičnu promenljivu x_6 razlika F_6–C_6=0. To znači da se može odrediti još jedno rešenje sa istom vrednošću funkcije cilja, odnosno još jedno optimalno rešenje u kome će biti promenljiva x_6. Kod određivanja koja promenljiva treba da napusti bazu može se uočiti da su oba pozitivna količnika ista: 50:1/2=100 i 25:1/4=100. To znači da obe promenljive x_4 i x_1 podjednako konkurišu da napuste bazu. Svejedno je koja će od ovih promenljivih izaći, jer će i jedna i druga imati vrednost nula u novom optimalnom rešenju. Obe promenljiva će, u narednoj iteraciji, doprineti povećanju funkcije cilja za nula. Naravno jedna od njih će biti bazična a druga nebazična promenljiva. Proizvoljno je određeno da promenljiva x_4 izađe iz rešenja. Novo optimalno rešenje pronađeno je u *ST*_4 i prikazano u tabeli 2.8.9.

Tabela 2.8.9. *Drugo optimalno rešenje – nakon četvrte iteracije ST_4*

C_b	X_b	B	80	50	40	20	0	0	$-M$	$-M$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	x_6	**100**	0	2	0	2	1	1	0	-1
40	x_3	**100**	0	1/2	1	3/2	-1/4	0	2	0
80	x_1	**0**	1	1/2	0	-1/2	1/4	0	-1	0
F_j–c_j		**4000**	0	10	0	0	10	0	0	0
		0	0	0	0	0	0	0	1	1

Novo, drugo, optimalno rešenje sačinjavaju promenljive:

$$x_1 = 0,\ \ x_3 = 100,\ \ x_6 = 100,$$

a maksimalna vrednost funkcije kriterijuma iznosi: max $F(x) = 4000$, i ista je kao i za prethodno, prvo, optimalno rešenje iz tabele *ST*_3.

Napomena 2.8.4.: Konveksnom kombinacijom ova dva optimalna rešenja mogu se dobiti i druga, takođe, optimalna rešenja. Treba naglasiti da su optimalna rešenja iz tabele *ST*_3 i tabele *ST*_4 bazna rešenja, a sva rešenja koja se mogu dobiti kao konveksna kombinacija baznih rešenja nisu bazna.

Ukoliko označimo optimalno rešenje iz simpleks tabele *ST*_3 sa X_o^1, a rešenje iz *ST*_4 sa X_o^2, dobija se izraz za konveksnu kombinaciju:

$$X_o^3 = q \cdot X_o^1 + (1 - q) \cdot X_o^2, \quad 0<q<1,$$

gde X_o^3 predstavlja novo optimalno rešenje. Kako je:

$$X_o^1 = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \\ x_6 \end{bmatrix} = \begin{bmatrix} 25 \\ 25 \\ 50 \\ 0 \end{bmatrix}, \quad X_o^1 = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 100 \\ 0 \\ 100 \end{bmatrix}$$

za, npr. q=2/5 dobija se konveksnu kombinacija:

$$X_o^3 = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \\ x_6 \end{bmatrix} = \frac{2}{5} \cdot X_o^1 + \frac{3}{5} \cdot X_o^2 = \frac{2}{5} \cdot \begin{bmatrix} 25 \\ 25 \\ 50 \\ 0 \end{bmatrix} + \frac{3}{5} \cdot \begin{bmatrix} 0 \\ 100 \\ 0 \\ 100 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 20 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 60 \\ 0 \\ 60 \end{bmatrix} = \begin{bmatrix} 10 \\ 70 \\ 20 \\ 60 \end{bmatrix}$$

Treće (nebazično) optimalno rešenje čine promenljive:

$$x_1 = 10, \quad x_3 = 70, \quad x_4 = 20, \quad x_6 = 60$$

sa maksimalnom vrednoscu funkcije cilja je ostala nepromenjena, tj.

$$\max F(x) = 80 \cdot 10 + 40 \cdot 70 + 20 \cdot 20 + 0 \cdot 60 = 4000.$$

2.8.4.2. *Rešavanje problema minimuma*

Naredni primer ilustruje rešavanje problema linearnog programiranja za slučaj minimiziranja ciljne funkcije.

Primer 2.8.6. Pronaći minimalnu vrednost funkcije cilja za dati matematički model.

$$\begin{aligned} & F(x) = 10x_1 + 3x_2 + 12x_3 + 8x_4 \\ \text{p.o.} \quad & 1x_1 + 2x_3 + 1x_4 \geq 16 \\ & 1x_1 + 1x_2 - 1x_4 = 4 \\ & 3x_1 + 2x_2 + 4x_3 + 2x_4 \leq 38 \\ & x_j \geq 0, \ j = 1,2,3,4 \end{aligned}$$

Rešenje. Nakon prilagođavanja modela za primenu simpleks metode dobija se model LP:

$$\begin{aligned} & \min F(x) = 10x_1 + 3x_2 + 12x_3 + 8x_4 + 0x_5 + M \cdot x_6 + M \cdot x_7 + 0x_8 \\ \text{p.o.} \quad & 1x_1 + 2x_3 + 1x_4 - x_5 + x_6 = 16 \\ & 1x_1 + 1x_2 - 1x_4 + x_7 = 4 \\ & 3x_1 + 2x_2 + 4x_3 + 2x_4 + x_8 = 38 \\ & x_j \geq 0, \ j = 1,2,\ldots,8 \end{aligned}$$

Napomena 2.8.5.: Razlike kod prilagođavanja matematičkog modela, sa zahtevom za minimizacijom ciljne funkcije u odnosu na model u kame se traži njena maksimalna vrednost, su sledeće:

- uz veštačke promenljive u funkciji cilja, kod problema minimuma, pojavljuju se koeficijenti $+M$;
- izmenjen je i kriterijum za izbor promenljive koja treba da uđe u naredno rešenje, koji glasi: *sve dok u redu za optimalnost* (F_j–c_j) *postoje pozitivni koeficijenti nije pronađeno optimalno rešenje problema*;
- u naredno rešenje ulazi ona promenljiva kojoj odgovara najveća pozitivna vrednost koeficijenta (F_j–c_j).

Simpleks tabela se popunjava na uobičajen način, a važe i sva pravila proračuna koja su primenjivana u problemima maksimizacije ciljne funkcije. Optimalno rešenje ovog problema pronađeno je od *ST*_0 - *ST*_3, koje su prikazane u tabelama od 2.8.10 – 2.8.13.

Tabela 2.8.10. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	**10**	**3**	**12**	**8**	**0**	***M***	***M***	**0**	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
M	x_6	**16**	1	0	**2**	1	-1	1	0	0	8
M	x_7	**4**	1	1	0	-1	0	0	1	0	/
0	x_8	**38**	3	2	4	2	0	0	0	1	9,5
$F_j – c_j$		**0**	-10	-3	-12	-8	0	0	0	0	**96**
		20	2	1	2	0	-1	0	0	0	**16**

Tabela 2.8.11. *Simpleks tabela ST_1*

C_b	X_b	B	**10**	**3**	**12**	**8**	**0**	***M***	***M***	**0**	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
12	x_3	**8**	1/2	0	1	1/2	-1/2	1/2	0	0	16
M	x_7	**4**	**1**	1	0	-1	0	0	1	0	4
0	x_8	**6**	1	2	0	0	2	-2	0	1	6
$F_j – c_j$		**96**	-4	-3	0	-2	-6	6	0	0	**16**
		4	1	1	0	-1	0	-1	0	0	**4**

Tabela 2.8.12. *Simpleks tabela ST_2*

C_b	X_b	B	**10**	**3**	**12**	**8**	**0**	***M***	***M***	**0**	$\theta = b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
12	x_3	**6**	0	-1/2	1	1	-1/2	1/2	-1/2	0	/
10	x_1	**4**	1	1	0	-1	0	0	1	0	4
0	x_8	**2**	0	**1**	0	1	2	-2	-1	1	2
$F_j – c_j$		**112**	0	1	0	-6	-6	6	4	0	**2**
		0	0	0	0	0	0	-1	-1	0	**0**

Tabela 2.8.13. *Optimalno rešenje – simpleks tabela ST_3*

C_b	X_b	B	**10**	**3**	**12**	**8**	**0**	***M***	***M***	**0**
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
12	x_3	**7**	0	0	1	3/2	1/2	1/2	-1	1/2
10	x_1	**2**	1	0	0	-2	-2	2	2	-1
3	x_2	**2**	0	1	0	1	2	-2	-1	1
$F_j – c_j$		**110**	0	0	0	-7	-8	8	5	-1
		0	0	0	0	0	0	-1	-1	0

Rešenje u tabeli ST_3 je optimalno jer nema više pozitivnih koeficijenata ($F_j – c_j$), i ono glasi: $x_1=2$, $x_2=2$, $x_3=7$, a minimalna vrednost funkcije kriterijuma iznosi: min $F(x)=110$.

2.8.5. Problem linearnog programiranja sa neograničenim rešenjem i bez rešenja

Problemi, koji su rešavani u predhodnim poglavljima, imali su konačna optimalna rešenja. Logično se postavlja pitanje da li svaki problem linearnog programiranja ima konačno optimalno rešenje. Kod realnih ekonomskih problema, ukoliko se korektno formuliše sistem ograničenja na osnovu stvarnih ograničavajućih uslova, problem će uvek imati konačno optimalno rešenje. Ukoliko se, kod ekonomskih problema, u postupku traženja optimalnog rešenja ne može odrediti konačno optimalno rešenje, to je pre posledica neadekvatnog postavljanja matematičkog modela nego što je stvarni problem takav.

Kod grafičkog rešavanja problema lako se može uočiti da li problem ima rešenje ili ne, a može se odrediti i priroda tog rešenje. Sistem ograničenja formira zajedničko područje (oblast definisansti), koje predstavlja skup mogućih rešenja problema. Kada je zajedničko područje zatvoreno problem ima konačno optimalno rešenje, u suprotnom ukoliko je zajedničko područje otvoreno problem može imati konačno optimalno rešenje, ali može imati i neograničeno rešenje. Ovde se ne razmatra slučaj kada sistem ograničenja uopšte ne formira zajedničko područje, jer tada problem nema rešenja.

Kada se zadatak linearnog programiranja rešava simpleks tabelom, takođe, je moguće odrediti da li problem ima neograničeno rešenje, i da li uopšte ima rešenja.

2.8.5.1. Problem sa neograničenim rešenjem

U primeru 2.8.7. ilustrovan je zadatak LP kada se može tvrditi da problem ima neograničeno rešenje i da se ekstremna vrednost funkcije cilja postiže negde u beskonačnosti.

Primer 2.8.7. Pronaći maksimalnu vrednost funkcije cilja za dati matematički model.

$$\begin{aligned} & \max F(x) = 40x_1 + 60x_2 \\ \text{p.o.}\quad & -1\,x_1 + 4x_2 \geq 400 \\ & -2\,x_1 + 2x_2 \leq 500 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Rešenje. Nakon proširenja i prilagođavanja za rešavanje, dobija se sistem:

$$\begin{aligned} & \max F(x) = 40x_1 + 60x_2 + 0x_3 - M\cdot x_4 + 0x_5 \\ \text{p.o.}\quad & -1\,x_1 + 4x_2 - x_3 + x_4 = 400 \\ & -2\,x_1 + 2x_2 + x_5 = 500 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

a) Problem rešavan simpleks tabelom.

Tabela 2.8.14. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	40	60	0	$-M$	0	$\theta=b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	
$-M$	x_4	**400**	-1	**4**	-1	1	0	100
0	x_5	**500**	-2	2	0	0	1	250
F_j-c_j		**0**	-40	-60	0	0	0	**6000**
		-400	1	-4	1	0	0	**400**

Početno rešenje, prikazano u tabeli 2.8.14, nije optimalno. Pošto se traži maksimalna vrednost fimkcije cilja, određuje se da u bazu ulazi promenljiva x_2, a iz baze izlazi promenljiva x_4. Simpleks tabela *ST*_1 prikazana je u tabeli 2.8.15.

Tabela 2.8.15. *Simpleks tabela ST_1*

C_b	X_b	B	40	60	0	$-M$	0	$\theta=b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	
60	x_2	**100**	-1/4	**1**	-1/4	1/4	0	
0	x_5	**300**	-3/2	0	1/2	-1/2	1	
F_j-c_j		**6000**	-55	0	-15	15	0	
		0	0	0	0	1	0	

Rešenje iz simpleks tabele *ST*_1 nije optimalno. U naredno rešenje ulazi promenljiva x_1. Potrebno je odrediti promcnljivu koja će da izađe iz rešenja. To se određuje na standardni način, na osnovu količnika između kolone B i kolone x_1. U koloni x_1 nema pozitivnih koeficijenata pomoću kojih se može odrediti pozitivna vrednost θ kriterijuma. To znači da se ne može odrediti promenljiva koja će izaći iz rešenja. Ovakva situacija signalizira da je rešenje problema *neograničeno* i da se vrednost funkcije cilja može povećavati preko svih granica. Simpleks tabela *ST*_2 se ne može sastaviti, ali se vrednost promenljivih mogu odrediti na sledeći način.

Pošto se ne može odrediti koja promenljiva treba da izađe iz baze, proizvoljno se izabere jedna od njih, iz kolone X_b. Bira se između promenljive x_2 i x_5. Proizvoljno je izabrana promenljiva x_5. Takođe, proizvoljno se odredi što veća pozitivna vrednost promenljive koja je ušla u naredno rešenje, x_1. Vrednost druge promenljive izračunava se prema pravilu:

$$x_2^3 = x_2^2 - (-\frac{1}{4})x_1^3$$

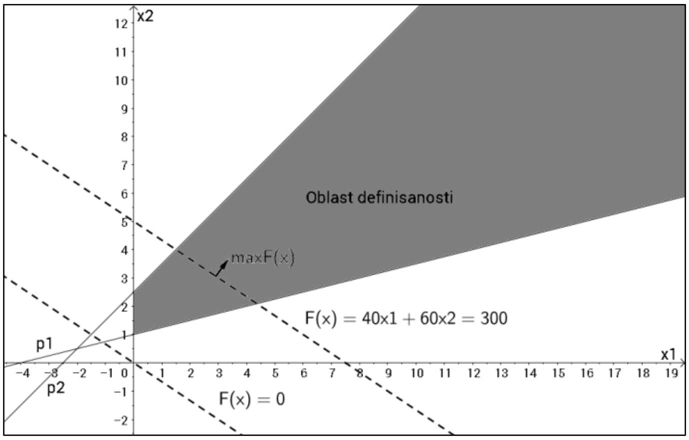
gde super indeks označava iteraciju za koju se računa vrednost promenljive. Proizvoljno je izabrana vrednost za promenljivu x_1= 4000, pa se za promenljivu x_2 dobija:

$$x_2^3 = 100 - (-\frac{1}{4}) \cdot 4000 = 1.100$$

Promenljivoj x_1 se može dodeliti i veća vrednost, promenljiva x_2 ostaje pozitivna, i dobija sve veću vrednost. Zato se kaže da je vrednost promenljivih u ovom slučaju neograničena, a maksimalna vrednost funkcije cilja postiže negde u beskonačnosti.

Napomena 2.8.6.: Ovakvi problemi nemaju neki praktični značaj, jer neograničeno rešenje nije optimalno rešenje.

b) Problem rešavan grafičkim metodom, prikazan je na slici 2.8.5.



Slika 2.8.5. *Grafičko rešenje problema 2.8.7.*

2.8.5.2. Problem bez rešenja

Ukoliko je problem prilagođen za rešavanje uvođenjem veštačkih promenljivih, onda se tokom rešavanja veštačke promenljive moraju eliminisati iz rešenja, tako da u optimalnom rešenju ne može biti prisutna ni jedna od njih. Kod korektno formulisanih problema taj uslov je uvek ispunjen, ali kod rešavanja problema mogu da se jave i neki izuzeci.

- Može se desiti da je posle određenog broja iteracija kriterijum optimalnosti, prema koeficijentima iz reda (F_j–c_j), ispunjen, ali da se u rešenju pored realnih i dopunskih promenijivih nalaze i veštačke promenljive sa pozitivnom vrednošću. U tom slučaju problem nema optimalno rešenje. Sistem nejednačina je kontradiktoran, odnosno, uslovi su nekonzistentni.
- Može se desiti situacija da je kriterijum optimalnosti ispunjen, u rešenju se nalaze i veštačke promenljive, ali je njihova vrednost jednaka nuli. U ovom slučaju rešenje problema je optimalno (veštačke promenljive se zanemare), a sistem ograničenja ima više uslova nego što je potrebno. Suvišna ograničenja nisu protivrečna ostalim ograničenjima.

2.8.6. Problem degeneracije

Kolona *B*, u simpleks tabeli, sadrži vrednosti promenljivih koje čine rešenje problema u toj iteraciji (vrednost promenljivih iz kolone X_b). U predhodnim primerima te vrednosti su uvek bile pozitivne, a takva rešenja nazivaju se nedegenerisaina. Rešenje koje u koloni *B* ima jednu, ili više nula, predstavlja *degenerisano* rešenje. U primeru 2.8.8. prikazaće se problem linearnog programiranja kada se javlja degeneracija rešenja, objasniće se u čemu je njegov značaj, i kako se on rešava.

Ukoliko se u određivanju promenljive, koja treba da izađe iz rešenja, pojave bar dva najmanja i jednaka količnika, onda će novo rešenje imati i promenljive koje će biti jednake nuli. Drugim rečima, novo rešenje postaje degenerisano. Pored toga, u ovom slučaju se ne može jednoznačno odrediti promenljiva koja treba da izađe iz rešenja. Prisustvo nule u koloni *B*, simpleks tabele, otvara mogućnost da promenljiva koja ulazi u naredno rešenje dobije vrednost nula. Kad se to desi, vrednost funkcije cilja narednog rešenja neće se razlikovati od njene vrednosti za prethodno rešenje. Pošto vrednost funkcije cilja ostaje ista, dobija se novo rešenje koje ne predstavlja poboljšanje u odnosu na prethodna. U takvim slučajevima, nekoliko uzastopnih tabela mogu dati rešenja sa istorn vrednošću funkcije cilja. Tada postoji mogućnost da se rešenja ponavljaju.

Degeneracija rešenja se ne javlja tako često, pa kod praktičnog rešavanja nije posebno važan problem. Osim toga, razvijen je i postupak kojim se rešava problem određivanja promenljive (za slučaj degeneracije) koja izlazi iz rešenja, tako da dobijena rešenja uvek konvergiraju prema optimalnom rešenju. U primeru 2.8.8. biće razmotrena pojava degeneracije, kao i postupak izbora promenljive koja izlazi iz rešenja u tom slučaju.

Primer 2.8.8. Pronaći maksimalnu vrednost funkcije cilja za dati matematički model.

$$\max F(x) = 30x_1 + 60x_2 + 50x_3$$
$$\text{p.o.} \quad 3x_1 + 4x_2 + 2x_3 \le 60$$
$$1x_1 + 2x_2 + 2x_3 \le 30$$
$$2x_1 + 1x_2 + 2x_3 \le 40$$
$$x_1, x_2, x_3 \ge 0$$

Rešenje. Uvođenjem dopunskih promenljivih x_4, x_5 i x_6 dobija se prilagođeni model:

$$\max F(x) = 30x_1 + 60x_2 + 50x_3 + 0\cdot(x_4 + x_5 + x_6)$$
$$\text{p.o.} \quad 3x_1 + 4x_2 + 2x_3 + x_4 = 60$$
$$1x_1 + 2x_2 + 2x_3 + x_5 = 30$$
$$2x_1 + 1x_2 + 2x_3 + x_6 = 40$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \ge 0$$

Početna simpleks tabela *ST*_0, prikazana je u tabeli 2.8.16.

Tabela 2.8.16. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	**30**	**60**	**50**	**0**	**0**	**0**	$\theta=b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	
0	x_4	**60**	3	4	2	1	0	0	15
0	x_5	**30**	1	**2**	2	0	1	0	15
0	x_6	**40**	2	1	2	0	0	1	40
$F_j\!-\!c_j$		**0**	-30	-60	-50	0	0	0	**900**

U naredno rešenje ulazi promenljiva x_2. Potrebno je da se odredi promenljiva koja izlazi iz rešenja. Zbog toga se, uobičajenim postupkom, određuju količnici između kolone *B* i kolone X_2:

$$60/4=15, \quad 30/2=15, \quad 40/1=40.$$

Iz dobijenog rešenja se uošava da su prva dva količnika najmanja i jednaka. To znači da za izlazak iz rešenja podjednako konkurišu promenljive x_4 i x_5. Zbog toga se ne može jednoznačno odrediti koja promenljiva treba da izađe iz rešenja. Potrebno je, za redove u kojima su jednaki količnici (redovi x_4 i x_5), odrediti količnike i izineđu kolone X_1 i X_2:

$$3:4=3/4, \quad 1:2=1/2$$

Kako je od ova dva količnika drugi manji, značu da iz rešenja izlazi promenljiva iz odgovarajućeg reda, x_5.

Napomena 2.8.7.: Ukoliko su količnici prve kolone i označene kolone ponovo isti, nastavljamo sa određivanjem količnika za naredne kolone, sve dok se ne dobije jedan manji količnik.
Simpleks tabela *ST*_1 prikazana je u tabeli 2.8.17.

Tabela 2.8.17. *Simpleks tabela ST_1*

C_b	X_b	B	**30**	**60**	**50**	**0**	**0**	**0**
			x_1	x_2	x_3	x_4	x_5	x_6
0	x_4	**0**	1	0	-2	1	-2	0
60	x_2	**15**	1/2	1	1	0	1/2	0
0	x_6	**25**	3/2	0	1	0	-1/2	1
$F_j\!-\!c_j$		**900**	0	0	10	0	30	0

U simpleks tabeli *ST*_1 pronađeno je optimalno rešenje:

$$x_2 = 5, \quad x_4 = 0, \quad x_6 = 25$$

Maksimalna vrednost funkcije cilja je: max$F(x)$=900. Bazična promenljiva x_4 jednaka je nuli, pa je ovo optimalno rešenje degenerisano.

2.8.7. Algoritmi simpleks metoda

Originalna simpleks procedura se sastoji u tome da se svaki element simpleks tabele transformiše, prema određenim pravilima, u odgovarajuće elemente nove tabele. Taj postupak transformisanja se ponavlja tako što se naredna simpleks tabela izračunava na osnovu prethodne tabele. U postupku računanja koeficijenata ne sme biti grešaka. Greška učinjena u jednoj tabeli prenosi se i širi i u naredne tabele. Čak i zaokruživanja, koja vrše elektronski računari kod proračuna, mogu uticati na tačnost pronađenog rešenja ukoliko je broj iteracija veliki.

Problem kontrole tih grešaka rešavan je na dva načina:

- Formulisani su metodi za kontrolu operacija u simpleks proceduri. Zadatak tih metoda je da otkriju eventualne greške u računanju i spreče njihovo dalje prenošenje.
- Drugi pristup je značajniji. Radi se o daljem razvoju i usavršavanju simpleks procedure kako bi se ublažili ovi problemi.

Tako je, raznim modifikacijama simpleks procedure, razvijeno više algoritama simpleks metoda. Za jedan broj algoritama karakteristično je da u postupku rešavanja problema koriste inverznu matricu vektora baze i originalne-polazne podatke ostalih elemenata matematičkog modela. Takav postupak obezbeđuje da se eventualne greške, načinjene u jednoj iteraciji, ne prenose na naredne iteracije. Koordinate linearno zavisnih vektora utvrđuju se za svaku iteraciju pomoću inverzne matrice i originalnih podataka. Od algoritama sa ovim karakteristikama biće razmotreni:

- Dantzig – ov algoritam (metod inverzne matrice),
- revidirani simpleks metod, i
- dualni simpleks metod.

Dualni simpleks metod koji, za razliku od ostalih algoritama simpleks metoda, daje postupak određivanja optimalnog rešenja kada u bazičnom rešenju ima i negativnih koordinata, odnosno kada je neko $x_i<0$. Ovaj algoritam ima važnu primenu i značajne prednosti u određenim slučajevima.

2.8.7.1. Dantzigov algoritam

Postupak ovog algoritma prikazan je na opštem modelu linearnog programiranja, koji je uvođenjem izravnavajućih i veštačkih promenljivih prilagođen za rešavanje simpleks procedurom. Razmatra se problem maksimizacije vrednosti funkcije cilja. Matematički model ovako definisanog problema izgleda:

$$\max F(x) = C^T \cdot X$$
$$A \cdot X = b$$
$$X \geq 0$$

gde su:

- C^T – vektor red sa n koordinata,
- X – vektor kolona sa n koordinata,
- b – vektor kolona sa m koordinata
- A – matrica reda $m \times n$.

Pored toga, matrica A formira n vektor kolona sa po m koordinata, tj. $A=[A_1\ A_2,..., A_n]$. Među vektorima matrice A ima m linearno nezavisnih. Oni čine bazu vektorskog prostora i pomoću njih se mogu jednoznačno izraziti svi vektori modela. Od bazičnih vektora možemo formirati kvadratnu matricu Ab, tj. $Ab=[A_1\ A_2\ ...\ A_m]$, a od m pozitivnih koordinata vektora X, koje stoje uz bazne vektore, formira se vektor $Xb=[x_1,x_2,...x_m]$. Sistem ograničenja $A \cdot X=b$, odnosno vektor b, se može izraziti pomoću matrice baze Ab relacijom 2.8.26.:

$$Ab \cdot Xb = b \qquad (2.8.26.)$$

Iz ove relacije se, pomoću inverzne matrice, mogu odrediti koordinate vektora Xb odnosno rešenje problema za posmatranu iteraciju, tj.:

$$Xb = Ab^{-1} \cdot b \qquad (2.8.27.)$$

Odgovarajuća vrednost funkcije cilja za pronađeno rešenje je

$$F(X) = Cb \cdot Xb \qquad (2.8.28.)$$

gde vektor Cb obuhvata koordinate vektora C koje stoje uz bazične promenljive (vektor Xb) u funkciji cilja.

Linearno zavisni vektori matrice A se mogu izraziti pomoću matrice baze Ab, na sledeći način:

$$Ab \cdot X_j = A_j \tag{2.8.29.}$$

gde je A_j, $j=(m+1,m+2,...,n)$, nebazni vektor. Nakon sređivanja dobija se izraz:

$$X_j = Ab^{-1} \cdot A_j \tag{2.8.30.}$$

Relacijom (2.8.30.) izvršena je transformacija koeficijenata matrice A, pa je sada moguće izračunati i koeficijente $(F_j – c_j)$, kao što je prikazano relacijom (2.8.31.).

$$F_j – c_j = Cb \cdot X_j – c_j,\ \ j = (m+1,...,n) \tag{2.8.31.}$$

Ovi koeficijenti omogućavaju ocenu optimalnosti pronađenog rešenja a nakon toga i izbor vektora koji će da uđe u narednu bazu vektorskog prostora. Vektori koji će da uđu u novo rešenje određuju se na osnovu relacije (2.8.32.).

$$\left[\max_j \theta_j \cdot (F_j - c_j)\right] < 0 \tag{2.8.32.}$$

Relacije (2.8.33.) predstavlja kriterijum koji određuje vektor koji napusta bazu vektorskog prostora.

$$\theta = \min_i \frac{b_i}{a_{ij}}, \quad i = 1,2,\dots,m \tag{2.8.33.}$$

Na ovaj način je postupak rešavanja problema Dantzigovim algoritmom zaokružen. Opisani postupak se zasniva na korišćenju inverzne matrice, a preciznije se formuliše na sledeći način:

***Korak* 1.** Uvođenjem dopunskih i veštačkih promenljivih, matematički model problema se prilagodi za rešavanje. Nakon toga se izvrši popis svih vektora iz modela.

***Korak* 2.** Od linearno nezavisnih vektora, koji čine bazu vektorskog prostora, formira se matrica Ab.

***Korak* 3.** Pronalazi se inverzna matrica matrice Ab, tj. matrica Ab^{-1}.

***Korak* 4.** Izračunava se bazno moguće rešenje problema: $Xb = Ab^{-1} \cdot b$.

***Korak* 5.** Za pronađeno rešenje izračunava se vrednost funkcije cilja: $F(X)=Cb \cdot Xb$.

***Korak* 6.** Određuju se koordinate svih linearno zavisnih vektora $X_j=Ab^{-1} \cdot A_j$, $j=(m+1, ..., n)$.

***Korak* 7.** Izračunavaju se koeficijenti $(F_j–c_j)$ za sve linearno zavisne vektore $F_j–c_j = Cb \cdot X_j – c_j,\ \ j = (m+1,...,n)$.

***Korak* 8.** Na osnovu vrednosti koeficijenata ($F_j – c_j$) proverava se da li je pronađeno optimalno rešenje.

***Korak* 9.** Ukoliko nije pronađeno optimalno rešenje, onda se, takođe na osnovu koeficijenata ($F_j – c_j$), odredi vektor koji ulazi u narednu bazu vektorskog prostora.

***Korak* 10.** Odredi se vektor koji treba da izađe iz baze vektorskog prostora.

***Korak* 11.** Formira se nova matrica Ab, odnosno nova baza vektorskog prostora, čime se postupak vraća na korak 2. Čitav postupak se ponavlja sve dok se u koraku 8. ne utvrdi da je pronađeno optimalno rešenje.

U primeru 2.8.9. ilustrovan je ceo postupak korišćenja Dantzigovog algoritma.

***Primer* 2.8.9.** Pronaći maksimalnu vrednost funkcije cilja za dati matematički model.

$$\max F(x) = 6x_1 + 4x_2$$

po

$$\begin{aligned} 2x_1 + 1x_2 &\leq 100 \\ 1x_1 + 3x_2 &\leq 120 \\ 2x_1 + 2x_2 &\leq 120 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Rešenje. Prošireni (prilagođeni) oblik matematičkog modela je:

$$\begin{aligned} &\max F(x) = 6x_1 + 4x_2 + 0\cdot(x_3 + x_4 + x_5) \\ &2x_1 + 1x_2 + x_3 \qquad\qquad\quad = 100 \\ &1x_1 + 3x_2 \qquad + x_4 \qquad = 120 \\ &2x_1 + 2x_2 \qquad\qquad + x_5 = 120 \\ &x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

1. U prilagođenom modelu postoje sledeće matrice i vektori sistema:

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 3 & 0 & 1 & 0 \\ 2 & 2 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 100 \\ 120 \\ 120 \end{bmatrix}, \quad C = \begin{bmatrix} 6 & 4 & 0 & 0 & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad A_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2. Bazu vektorskog prostora čine linearno nezavisni vektori. U prvoj iteraciji to su jedinični vektori, pa je matrica Ab za početnu iteraciju:

$$Ab_1 = \begin{bmatrix} A_3 & A_4 & A_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Kako je matrica Ab jedinična matrica, onda je inverzna matrica Ab^{-1}:

$$Ab_1^{-1} = Ab_1$$

4. Rešenje za prvu iteraciju određuje se preko relacije (2.8.27.), pa je:

$$Xb = Ab_1^{-1} \cdot b = I \cdot b = b$$

odnosno:
$$Xb = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 100 \\ 120 \\ 120 \end{bmatrix}$$

5. Vrednost funkcije cilja se izračunava preko relacije (2.8.27.), tj. za ovo rešenje iznosi:

$$F(X) = Cb \cdot Xb = \begin{bmatrix} c_3 & c_4 & c_5 \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 120 \\ 120 \end{bmatrix} = 0$$

6. Koordinate linearno zavisnih, nebazičnih, vektora određuju se preko relacije (2.8.30.), tj.:

$$X_j = Ab_1^{-1} \cdot A_j = \mathrm{I} \cdot A_j = A_j, \qquad j=1,2.$$

7. Koeficijenti ($F_j – c_j$) određuju se preko relacije (2.8.31.), tj.:

za $j=1$: $$F_1 - c_1 = Cb \cdot X_1 - c_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 6 = -6$$

za $j=2$: $$F_2 - c_2 = Cb \cdot X_2 - c_2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} - 4 = -4$$

8. Nije pronađeno optimalno rešenje jer postoje koeficijenti ($F_j – c_j$) < 0.
9. Prema relaciji (2.8.32.) u narednu bazu vektorskog prostora ulazi vektor A_1.
10. Prema relaciji (2.8.33.) iz baze izlazi vektor A_3, tj.:

$$\theta = \min_{i=1,2,3} \left(\frac{b_1}{a_{11}}, \frac{b_2}{a_{21}}, \frac{b_3}{a_{31}} \right) = \min_i \left(\frac{100}{2}, \frac{120}{1}, \frac{120}{2} \right) = \min_i (50,120,60) = 50 \Rightarrow A_3$$

11. Formira se nova matrica Ab_2, i vraća na tačku 2. Postupak se, od tačke 2., ponavlja i za drugu iteraciju. Bazu vektorskog prostora sada čine linearno nezavisni vektori:

$$Ab_2 = \begin{bmatrix} A_1 & A_4 & A_5 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

3. Inverzna matrica matrice Ab_2, je:

$$Ab_2^{-1} = \begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

4. Rešenje problema u drugoj iteraciji je:

$$Xb = \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = Ab_2^{-1} \cdot b = \begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 120 \\ 120 \end{bmatrix} = \begin{bmatrix} 50 \\ 70 \\ 20 \end{bmatrix}$$

5. Vrednost funkcije cilja nakon druge iteracije iznosi:

$$F(X) = Cb \cdot Xb = \begin{bmatrix} c_1 & c_4 & c_5 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 50 \\ 70 \\ 20 \end{bmatrix} = 300$$

6. Koordinate linearno zavisnih, nebazičnih, vektora, u drugoj iteraciji, računaju se prema izrazu: $X_j = Ab_2^{-1} \cdot A_j$, j=2,3, tj.:

$$X_2 = Ab_2^{-1} \cdot A_2 = \begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 5/2 \\ 1 \end{bmatrix}$$

$$X_3 = Ab_2^{-1} \cdot A_3 = \begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ -1/2 \\ -1 \end{bmatrix}$$

7. Koeficijenti (F_j–c_j) izračunavaju se prema relaciji: $F_j - c_j = Cb \cdot X_j - c_j$, j=2,3,tj.:

za j=2: $$F_2 - c_2 = Cb \cdot X_2 - c_2 = \begin{bmatrix} 6 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/2 \\ 5/2 \\ 1 \end{bmatrix} - 4 = -1$$

za j=3: $$F_3 - c_3 = Cb \cdot X_3 - c_3 = \begin{bmatrix} 6 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/2 \\ -1/2 \\ -1 \end{bmatrix} - 0 = 3$$

8. Još uvek nije pronađeno optimalno rešenje, jer postoji koeficijent (F_2–c_2)= –1 < 0.
9. U narednu bazu vektorskog prostora ulazi vektor A_2.
10. Prema relaciji (8.33), količnici između odgovarajućih koordinata vektora B i vektora X_2 iznose:

$$\theta = \min_{i=1,2,3} \left(\frac{b_1}{a_{12}}, \frac{b_2}{a_{22}}, \frac{b_3}{a_{32}} \right) = \min_i \left(\frac{50}{1/2}, \frac{70}{5/2}, \frac{20}{1} \right) = \min_i (100, 28, 20) = 20 \Rightarrow A_5$$

Iz baze izlazi vektor A_5, jer njemu odgovara najmanji količnik.

11. Proračun se vraća na korak 2. Formira se matrica Ab_3 za treću iteraciju:

$$Ab_3 = \begin{bmatrix} A_1 & A_4 & A_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 3 \\ 2 & 0 & 2 \end{bmatrix}$$

3. Inverzna matrica matrice Ab_3, je:

$$Ab_3^{-1} = \begin{bmatrix} 1 & 0 & -1/2 \\ 2 & 1 & -5/2 \\ -1 & 0 & 1 \end{bmatrix}$$

4. Rešenje problema u trećoj iteraciji je:

$$Xb = \begin{bmatrix} x_1 \\ x_4 \\ x_2 \end{bmatrix} = Ab_3^{-1} \cdot b = \begin{bmatrix} 1 & 0 & -1/2 \\ 2 & 1 & -5/2 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 120 \\ 120 \end{bmatrix} = \begin{bmatrix} 40 \\ 20 \\ 20 \end{bmatrix}$$

5. Funkcije cilja nakon treće iteracije iznosi:

$$F(X) = Cb \cdot Xb = \begin{bmatrix} c_1 & c_4 & c_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_4 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} 40 \\ 20 \\ 20 \end{bmatrix} = 320$$

6. Koordinate linearno zavisnih, nebazičnih, vektora, u trećoj iteraciji, za j=3,5, su:

$$X_3 = Ab_3^{-1} \cdot A_3 = \begin{bmatrix} 1 & 0 & -1/2 \\ 2 & 1 & -5/2 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$$

$$X_5 = Ab_3^{-1} \cdot A_5 = \begin{bmatrix} 1 & 0 & -1/2 \\ 2 & 1 & -5/2 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -5/2 \\ 1 \end{bmatrix}$$

7. Koeficijenti (F_j–c_j), u trećoj iteraciji, za j=3,5, iznose:

za j=3: $$F_3 - c_3 = Cb \cdot X_3 - c_3 = \begin{bmatrix} 6 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} - 0 = 2$$

za j=5: $$F_5 - c_5 = Cb \cdot X_5 - c_5 = \begin{bmatrix} 6 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} -1/2 \\ -5/2 \\ 1 \end{bmatrix} - 0 = 1$$

8. Nakon treće iteracije pronađeno je optimalno rešenje, jer nema više negativnih koeficijenata (F_j–c_j). Vrednosti promenljivih za optimalno rešenje iznose: x_1=40, x_2=20, x_3=20. Maksimalna vrednost funkcije cilja je: max $F(x)$=320.

Opisani postupak rešavanja problema pomoću Dantzigovog algoritma ne zahteva obavezno rešavanje problema sa početkom od prve iteracije, odnosno od iteracije u kojoj bazu vektorskog prostora čine jedinični vektori. Postoji mogućnost da se proračun otpočne i od neke druge baze vektorskog prostora. Jedini problem u tom slučaju je što se može dogoditi da tako dobijeno početno bazično rešenje ne mora biti i moguće rešenje, odnosno, može se desiti da u tom slučaju neka od koordinata vektora *Xb* bude negativna.

2.8.7.2. Revidiran simpleks metod

Kod uobičajenog simpleks postupka, kriterijumi za ocenu optimalnosti rešenja i za izbor novog baznog vektora zahtevaju određivanja koeficijenata $(F_j–c_j)$ za svako j. Zbog toga je bilo potrebno u koraku 6. Dantzig-ovog algoritma odrediti koordinate svih nebazičnih vektora. Kada je n znatno veće od m, tada postoji veliki broj nebazičnih vektora čije koordinate treba odrediti. Međutim, revidirani simpleks metod ne zahteva određivanje koordinata svih nebazičnih vektora. Kod ovog algoritma mogu se prethodno odrediti koeficijenti $(F_j–c_j)$ i na osnovu njih oceniti optimalnost rešenja. Ukoliko nije pronađeno optimalno rešenje, odredi se novi bazični vektor, pa se odrede koordinate samo tog vektora. Na taj način se obim računanja znatno smanjuje jer ne treba izračunavati koordinate ostalih nebazičnih vektora.

Računski postupak revidirane simpleks metode detaljnije se prikazuje na primeru 2.8.9. Matematički model prilagođenog problema ima sledeće elemente:

$$
\begin{array}{l}
\max F(x) = 6x_1 + 4x_2 + 0\cdot(x_3 + x_4 + x_5) \\
2x_1 + 1x_2 + x_3 \qquad\qquad = 100 \\
1x_1 + 3x_2 \qquad + x_4 \qquad = 120 \\
2x_1 + 2x_2 \qquad\qquad + x_5 = 120 \\
x_1, x_2, x_3, x_4, x_5 \geq 0
\end{array}
$$

Revidirani simpleks metod zahteva da se funkcija cilja unese u sistem ograničenja, kao jedno novo ograničenje. To novo ograničenje ima oblik: $6x_1 + 4x_2 = x_6$, gde je promenljiva x_6 uvedena umesto oznake $F(x)$. Na ovaj način je dobijeno novo ograničenje čiji je oblik:

$$–6x_1 – 4x_2 + x_6 = 0$$

kojim se proširava prethodni sistem ograničenja. Matematički model dobija oblik:

$$
\begin{array}{l}
2x_1 + 1x_2 + x_3 \qquad\qquad\qquad = 100 \\
1x_1 + 3x_2 \qquad + x_4 \qquad\qquad = 120 \\
2x_1 + 2x_2 \qquad\qquad + x_5 \qquad = 120 \\
–6x_1 – 4x_2 \qquad\qquad\qquad + x_6 = 0 \\
x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{array}
$$

Na ovaj način, početni problem je dobio potreban oblik. Tek sada se može početi sa primenom revidiranog simpleks metoda. Ovaj metod se razlikuje od originalnog simpleks metoda po tome što se kod njega ne transformišu svi elementi matrice A, već samo elementi inverzne matrice Ab^{-1}. Transformacija se vrši po istim formulama i prema istim kriterijumima, kao i kod originalne simpleks tabele, ali u tabeli koja je prilagođena revidiranom simpleks metodu, tj. u revidiranoj simpleks tabeli.

Revidirana simpleks tabela je tako sastavljena da se u njoj koraci 2, 3, 4 i 5, Dantzig-ovog algoritma, lako izvršavaju. Ova tabela ima četiri kolone: prva kolona određuje vektore Xb koji čine matricu Ab, druga kolona sadrži bazično moguće rešenje B i vrednost funkcije cilja $F(x)$ za datu iteraciju (u poslednjem redu kolone), treća kolona obuhvata inverznu matricu Ab^{-1}, a četvrta se koristi za vektor X_j, koji ulazi u narednu bazu. Revidirana simpleks tabela ima zaglavlje i onoliko redova koliko prošireni model ima ograničenja.

Prva, početna, revidirana simpleks tabela prikazana je tabelom 2.8.18.

Tabela 2.8.18. *Početna revidirana simpleks tabela ST_0*

Xb	**B**	$\boldsymbol{Ab^{-1}}$				$\boldsymbol{X_j}$
X_3	100	1	0	0	0	
X_4	120	0	1	0	0	
X_5	120	0	0	1	0	
X_6	0	0	0	0	1	

Tabela 2.8.18. sadrži početno bazno moguće rešenje i odgovarajuću vrednost funkcije cilja, tj.:

$$x_3 = 100, \quad x_4 = 120, \quad x_5 = 120 \quad \text{i} \quad F_0(x) = 0$$

Proverava se da li je dobijeno optimalno rešenje. Ukoliko nije izračunavaju se koeficijenti $(F_j – c_j)$, prema relaciji (2.8.31.), tj.:

$$F_j – c_j = Cb \cdot X_j – c_j$$

Sistem ograničenja je proširen koeficijentima funkcije cilja, zato se u relaciji (2.8.31.) i na levoj i na desnoj strani mogu izostaviti koeficijenti c_j. Tako se ova relacija svodi na izraz oblika:

$$F_j = Cb \cdot X_j$$

Koeficijenti F_j imaju istu vrednost kao i koeficijenti $(F_j – c_j)$ kod drugih algoritama. Zbog toga će oni imati isti značaj i istu funkciju, tj. služiće za ocenu optimalnosti rešenja i za izbor vektora koji treba da uđe u narednu bazu.
Koordinate vektora X_j određuju se relacijom (2.8.30.), tj.:

$$X_j = Ab^{-1} \cdot A_j$$

nakon zamene prethodni izraz dobija oblik kao što je prikazano relacijom (2.8.34.):

$$F_j = Cb \cdot Ab^{-1} \cdot A_j \qquad (2.8.34.)$$

Značajno pojednostavljenje postupka revidiranog simpleks metoda je i u tome što ne treba posebno računati proizvod $Cb \cdot Ab^{-1}$ iz relacije (2.8.34.). Poslednji red matrice Ab^{-1} u revidiranoj simpleks tabeli sadrži vektor red, koji je rezultat proizvoda:

$$Cb \cdot Ab^{-1}$$

Korišćenjem predhodnih relacija, nastavlja se postupak rešavanja primera 2.8.9. Vektori A_1 i A_2 su nebazni, pa se za njih izračunavaju koeficijenti F_j. Prema relaciji (2.8.34.), $F_j = Cb \cdot Ab^{-1} \cdot A_j$, j=1,2, oni su:

$$F_1 = Cb \cdot Ab^{-1} \cdot A_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 2 \\ -6 \end{bmatrix} = -6$$

$$F_2 = Cb \cdot Ab^{-1} \cdot A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \\ -4 \end{bmatrix} = -4$$

Koeficijenti su negativni, pa se zaključuje da nije pronađeno optimalno rešenje i da u narednu bazu vektorskog prostora treba da uđe vektor A_1. Potrebno je odrediti samo koordinate vektora X_1, na osnovu relacije (2.8.30.), tj.:

$$X_1 = Ab^{-1} \cdot A_1 = I \cdot A_1 = A_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \\ -6 \end{bmatrix}$$

Dobijene koordinate vektora X_1 se upisuju u kolonu X_j početne simpleks tabele *T*_8.22, tako da je njen konačni oblik prikazan u tabeli 2.8.19.:

Tabela 2.8.19. *Konačni oblik početne revidirane simpleks tabele*

Xb	***B***	Ab^{-1}				X_1
X_3	100	1	0	0	0	2
X_4	120	0	1	0	0	1
X_5	120	0	0	1	0	2
X_6	0	0	0	0	1	-6

Početna tabela je pripremljena za pronalaženje narednog rešenja. Dalji postupak je identičan kao i kod obične simpleks tabele. Kolona X_1 je označena kolona, i ona predstavlja vodeću kolonu. Prema količniku između kolone *B* i označene kolone X_1 određuje se da iz baze treba da izađe vektor A_3, pa se označi i prvi red tabele. Na osnovu početne tabele dobija se tabela 2.8.20. za narednu iteraciju.

Tabela 2.8.20. *Revidirana simpleks tabela nakon prve iteracije*

Xb	***B***	Ab^{-1}				X_2
X_1	50	1/2	0	0	0	1/2
X_4	70	-1/2	1	0	0	5/2
X_5	20	-1	0	1	0	1
X_6	300	3	0	0	1	-1

Tabela 2.8.20. sadrži inverznu matricu za drugu iteraciju, bazno moguće rešenje i vrednost funkcije cilja. Potrebno je da se odrede koeficijenti F_j za j=2,3 i proveri da li je pronađeno optimalno rešenje. Prema relaciji (2.8.34.) izračunava se:

$$F_2 = Cb \cdot Ab^{-1} \cdot A_2 = \begin{bmatrix} 3 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \\ -4 \end{bmatrix} = -1$$

$$F_3 = Cb \cdot Ab^{-1} \cdot A_3 = \begin{bmatrix} 3 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 3$$

Koeficijent F_2 je negativan, pa u narednu bazu vektorskog prostora ulazi vektor A_2. Koordinate vektora X_2 se proračunavaju na osnovu relacije (2.8.30.), tj.:

$$X_2 = Ab^{-1} \cdot A_2 = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \\ -4 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 5/2 \\ 1 \\ -1 \end{bmatrix}$$

Koeficijenti se unose u poslednju kolonu tabele 2.8.20. To je označena kolona, i ona predstavlja vodeću kolonu. Na osnovu količnika između kolone B i kolone X_2, označen je treći red, koji predstavlja vodeći red. Iz baze izlazi vektor A_5. Na osnovu popunjene tabele 2.8.20. dobija se tabela 2.8.21. nakon druge iteracije.

Tabela 2.8.21. *Revidirana simpleks tabela nakon druge iteracije*

Xb	**B**	$\boldsymbol{Ab^{-1}}$				**X**
X_1	40	1	0	-1/2	0	
X_4	20	2	1	-5/2	0	
X_2	20	-1	0	1	0	
X_6	320	2	0	1	1	

Izračunavaje se koeficijenti F_j za j=3,5, tj.

$$F_3 = Cb \cdot Ab^{-1} \cdot A_3 = \begin{bmatrix} 2 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2$$

$$F_5 = Cb \cdot Ab^{-1} \cdot A_5 = \begin{bmatrix} 2 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = 1$$

Svi koeficijenti F_j su pozitivni. To znači da je rešenje, pronađeno u tabeli 2.8.21., optimalno rešenje, koga sačinjavaju promenljive:

$$x_1 = 40, \quad x_2 = 20, \quad x_4 = 20$$

a maksimalna vrednost funkcije cilja iznosi:

$$\max F(x) = 320$$

2.8.7.3. Dualni algoritam

U odnosu na uobičajenu simpleks proceduru, dualni algoritam predstavlja specijalni postupak za rešavanje problema linearnog programiranja. Njegova osnovna karakteristika je u tome da se optimalno rešenje problema pronalazi preko niza baznih rešenja koja nisu nenegativna, ali kod kojih koeficijenti (F_j–c_j) ispunjavaju uslov optimalnog rešenja. Iako se radi o opštem metodu koji se može primeniti na svaki problem linearnog programiranja, nije u svim slučajevima lako odrediti pogodan start za primenu dualnog algoritma. Ipak, u nizu slučajeva dualni algoritam ima vrlo važnu primenu i donosi niz prednosti u odnosu na ostale algoritme simpleks metoda. Najočiglednije prednosti u primeni ovog algoritma odnose se na probleme postoptimalne analize, kao i na rešavanje jedne vrste problema minimuma. Dualni algoritam je razmotren na primeru 2.8.10.

***Primer* 2.8.10**. Potrebno je pronaći minimalnu vrednost funkcije cilja za matematički model koji je dat kao:

$$\min F(x) = C^T \cdot X$$
$$A \cdot X \geq b$$
$$X \geq 0$$

gde su C i b nenegativni vektori. Početni problem se može na dva načina prilagoditi za primenu dualnog algoritma. Prvi način je da se transformiše u odgovarajući problem maksimuma, kao:

$$\max F(x) = -\,C^T \cdot X$$
$$-A \cdot X \leq -\,b$$
$$X \geq 0$$

Na ovaj problem može se primeniti dualni algoritam. Nije potrebno da se uvode veštačke promenljive; vektor b ima sve negativne koordinate, pa će i početno rešenje-vektor Xb, takođe, imati negativne koordinate; vektor C ima negativne koordinate, pa će kpeficijenti (F_j–c_j) za početno rešenje biti pozitivni, odnosno, ispunjavaće uslov optimalnosti, što znači da su ispunjeni svi uslovi za primenu ovog algoritma.

Drugi načinu transformisanja ne zahteva promenu funkcije cilja. U ovom slučaju potrebno je da se rešava transformisani problem minimuma:

$$\min F(x) = C^T \cdot X$$
$$-A \cdot X \leq -\,b$$
$$X \geq 0$$

I ovim načinom se obezbeđuje pogodan start za primenu dualnog algoritma. Potrebno je da se odrede kriterijumi po kojima se menja baza vektorskog prostora. Kod dualnog algoritma postupak izmene vektora u bazi je obrnut u odnosu na uobičajcni postupak. Ovo se ogleda u tome što se prvo odredi vektor koji izlazi iz baze vektorskog prostora, pa tek onda vektor koji ulazi u bazu.

Iz baze vektorskog prostora izlazi onaj vektor A_i kome odgovara najveća negativna koordinata baznog rešenja B, tj.

$$b_r = \max_i |b_i|, \quad b_i < 0 \tag{2.8.35.}$$

Vektor koji ulazi u novu bazu određuje se tako što se vodi računa da vrednost nove promenljive bude pozitivna i da koeficijenti (F_j–c_j) i za novo rešenje budu negativni. Potrebno je da se odredi:

$$\theta = \frac{b_r}{a_{rj}} > 0$$

Kako je b_r<0, potrebno je da se razmatraju samo vrednosti a_{rj}<0. Određuje se vektor koji ulazi u novu bazu pomoću izraza (8.36).

$$\frac{F_j - c_j}{a_{rj}}, \quad a_{rj} < 0 \tag{2.8.36.}$$

Vodi se računa o vrednostima koeficijenata (F_j–c_j). U novu bazu vektorskog prostora ulazi onaj vektor A_j kome odgovara najmanja vrednost izraza (8.36). Izrazom (8.37) formulisan je i drugi kriterijum.

$$\frac{F_u - c_u}{a_{ru}} = \min_j \frac{F_j - c_j}{a_{rj}}, \quad a_{rj} < 0, \quad j = 1,2,\ldots,k \tag{2.8.37.}$$

Primenom ova dva kriterijuma (2.8.35.) i (2.8.37.), dualni algoritam se može koristiti i u bilo kom drugom, poznatom, algoritmu. Ilistracija postupka dualnog algoritma prikazana je u primeru 2.8.11. Problem se rešava primenom potpune simpleks tabele.

***Primer* 2.8.11**. Korišćenjem dualnog algoritma rešiti matematički model:

$$\begin{aligned} & \min F(x) = 160x_1 + 100x_2 + 120x_3 \\ \text{p.o.} \quad & 2x_1 + 1x_2 + 2x_3 \geq 350 \\ & 1x_1 + 1x_2 + 1x_3 \geq 300 \\ & 4x_1 + 1x_2 + 2x_3 \geq 400 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Rešenje. Množenjem sistema ograničenja sa –1 (izbegavaju se veštačke promenljive), i uvođenjem dopunskih promenljivih x_4, x_5 i x_6, dobija se prilagođeni model za primenu dualnog algoritma:

$$\begin{aligned} & \min F(x) = 160x_1 + 100x_2 + 120x_3 + 0\cdot(x_4 + x_5 + x_6) \\ \text{p.o.} \quad & -2x_1 - 1x_2 - 2x_3 + x_4 = -350 \\ & -1x_1 - 1x_2 - 1x_3 + x_5 = -300 \\ & -4x_1 - 1x_2 - 2x_3 + x_6 = -400 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned}$$

Formira se početna simpleks tabela ST_0.

Tabela 2.8.22. *Početna simpleks tabela ST_0*

C_b	X_b	B	160	100	120	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6
0	x_4	**-350**	-2	-1	-2	1	0	0
0	x_5	**-300**	-1	-1	-1	0	1	0
0	x_6	**-400**	**-4**	-1	-2	0	0	1
$F_j – c_j$		**0**	-160	-100	-120	0	0	0

U početnoj simpleks tabeli x_6=–400 je najnegativnija koordinata početnog baznog rešenja *B*. Prema relaciji (2.8.35.) bazu napusta vektor A_6. Za *r*=6 (treći red simpleks tabele), uzimajući u obzir samo negativne koencijente a_{6j}, određuje se vrednost količnika prema relaciji (2.8.37.), i dobija se:

$$\min_j \frac{F_j - c_j}{a_{6j}} = \min\left(\frac{-160}{-4} = 40, \frac{-100}{-1} = 100, \frac{-120}{-2} = 60\right) = 40$$

Najmanji količnik je za *j*=1, što znači da u narednu bazu ulazi vektor A_1. U tabeli 2.8.22. označeni su kolona x_1 i treći red (x_6). Transformacija koeficijenata za novu tabelu vrši se prema pravilima uobičajene simpleks procedure. Izračunava se simpleks tabela nakon prve iteracije, koja je predstavljena u tabeli 2.8.23.

Tabela 2.8.23. *Simpleks tabela ST_1*

C_b	X_b	B	160	100	120	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6
0	x_4	**-150**	0	-1/2	-1	1	0	-1/2
0	x_5	**-200**	0	-3/4	**-1/2**	0	1	-1/4
160	x_1	**100**	1	1/4	1/2	0	0	-1/4
$F_j – c_j$		**16000**	0	-60	-40	0	0	-40

U tabeli 2.8.23. postoje još dve negativne koordinate baznog rešenja. Promenljiva x_5=–200 je najnegativnija vrednost, tako da iz baze izlazi vektor A_5. Iz relacije (2.8.37.) određuje se najmanji količnik, a to je za *j*=3, tj.:

$$\min_j \frac{F_j - c_j}{a_{5j}} = \min\left(\frac{-60}{-3/4} = 120, \frac{-40}{-1/2} = 80, \frac{-40}{-1/4} = 160\right) = 80$$

tako da u naredno bazno rešenje ulazi vektor A_3. Simpleks tabela nakon druge iteracije prikazana je u tabeli 2.8.24.

Tabela 2.8.24. *Simpleks tabela ST_2*

C_b	X_b	B	160	100	120	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6
0	x_4	**250**	0	1	0	1	-2	0
120	x_3	**400**	0	3/2	1	0	-2	1/2
160	x_1	**-100**	1	**-1/2**	0	0	1	-1/2
$F_j – c_j$		**32000**	0	0	0	0	-80	-20

U tabeli 2.8.24. se nalazi još jedna negativna koordinata u koloni *B* i odgovara promenljivoj x_1. Zato, vektor A_1 izlazi iz baze vektorskog prostora. Najmanji količnik iz relacije (2.8.37.) jednak je nuli, a ostvaruje se za *j*=2, tj.:

$$\min_j \frac{F_j - c_j}{a_{1j}} = \min\left(\frac{0}{-1/2} = 0, \frac{-20}{-1/2} = 40\right) = 0$$

tako da vektor A_2 ulazi u narednu bazu. Formira se simpleks tabela nakon treće iteracije, koja je prikazana u tabeli 2.8.25.

Tabela 2.8.25. *Simpleks tabela ST_3*

C_b	X_b	B	160	100	120	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6
0	x_4	**50**	2	0	0	1	0	-1
120	x_3	**100**	3	0	1	0	1	-1
100	x_2	**200**	-2	1	0	0	-2	1
F_j–c_j		**32000**	0	0	0	0	-80	-20

Rešenje iz tabele 2.8.25. je optimalno rešenje, i njega sačinjavaju promenljive:

x_2=200, x_3=100, x_4=50,

a minimalna vrednost funkcije cilja iznosi: min $F(x)$=32.000.

Napomena 2.8.8.: Simpleks tabela *ST*_3, pored rešenja problema i vrednosti funkcije cilja, sadrži i inverznu matricu, koja se može koristiti za razne postoptimalne analize. Međutim, pošto su sve nejednačine prvobitnog modela pomnožene sa (-1), potrebno je sada inverznu matricu koja odgovara optimalnom rešenju, takođe, pomnožiti sa (-1). Za analizu optimalnog rešenja može se koristiti inverzna matrica:

$$Ab^{-1} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

Napomena 2.8.9.: Potrebno je da se pomnože i dualne promenljive sa (-1). Pojam dualnih promenljivih i gde se one nalaze u simpleks tabeli detaljnije se razmatra u poglavlju 2.9.

2.9. DUALNI PROBLEMI

Svakom problemu linearnog programiranja odgovara jedan dualni problem i to tako da problemu minimuma korespondira određeni problem maksimuma, odnosno problemu maksimuma korespondira određeni problem minimuma. Iz ove osnovne osobine linearnog programiranja, može se zaključiti da je dual od dualnog problema primarni problem. Sa matematičkog aspekta, potpuno je svejedno koji se od ova dva problema smatra primarni, a koji dualni. Međutim, sa ekonomskog aspekta samo jedan problem (originalni-primarni) ima ekonomski smisao, dok drugi nema. Čak i kad se može dati ekonomska interpretacija dualnog problema, ona nije dovoljno očigledna.

U određenim slučajevima matematički model osnovnog problema linearnog programiranja ne može da posluži za nalaženje optimalnog plana primenom opisanog simpleks metoda. U ovakvim slučajevima se primarni problem preformuliše u dualni, pomoću kog je moguće naći traženo rešenje. Dualni problem se formira pomoću istih podataka na osnovu kojih je formiran i primarni Kako se radi o dva uzajamno dualna problema između njih postoje sledeće korespodencije:

1. Dualni model ima onoliko promenljivih koliko primarni zadatak ograničenja, i ograničenja koliko primarni zadatak promenljivih;
2. Slobodni članovi u ograničenjima primalnog zadatka postaju koeficijenti uz promenljive funkcije cilja dualnog modela, a koeficijenti uz promenljive funkcije cilja primarnog modela postaju slobodni članovi u ograničenjima dualnog modela;
3. Smer nejednačina dualnog modela je suprotan smeru nejednačina primarnog modela;
4. Ako je u primarnom modelu tražen maksimum funkcije cilja, u dualnom se traži minimum, i obrnuto;
5. Dopunskoj promenljivoj x_{n+1} primara koja je u optimalnom baznom rešenju odgovara promenljiva y_i u dualnom modelu, pri čemu je $x_{n+1} \cdot y_i = 0$ $i=1, 2, ..., m$;
6. Realnoj promenljivoj x_j primara iz baznog dopustivog rešenja odgovara dopunska promenljiva y_{m+j} dualnog modela sa nultom vrednošću $x_j \cdot y_{m+j} = 0$; $j=1,2,...,n$;
7. Matrica ograničenja u primarnom modelu jednaka je transponovanoj matrici ograničenja u dualnom modelu.

2.9.1. Simetrični dualni problem

Karakteristike simetričnog modela linearnog programiranja su:

- sva ograničenja primarnog problema su data u obliku nejednačina,
- sve nejednačine su istog smera,
- uslov nenegativnosti važi za sve dualne promenljive.

Relacijom (2.9.1.) je dat primarni, a relacijom (2.9.2.) dualni model problema linearnog programiranja, u opštem obliku.

Primarni problem:

$$
\begin{aligned}
&\min F(x) = c_1x_1 + c_2x_2 + ...+ c_nx_n \\
&\text{p.o.} \quad a_{11}x_1 + a_{12}x_2 + ...+ c_{1n}x_n \geq b_1 \\
&a_{21}x_1 + a_{22}x_2 + ...+ c_{2n}x_n \geq b_2 \\
&... \\
&a_{m1}x_1 + a_{m2}x_2 + ...+ c_{mn}x_n \geq b_m \\
&x_j \geq 0,\ j=1,2,...,n
\end{aligned}
\tag{2.9.1.}
$$

Dualni problem:

$$
\begin{aligned}
&\max G(y) = b_1y_1 + b_2y_2 + ...+ b_my_m \\
&\text{p.o.} \quad a_{11}y_1 + a_{21}y_2 + ...+ c_{m1}y_m \leq c_1 \\
&a_{12}y_1 + a_{22}y_2 + ...+ c_{m2}y_m \leq c_2 \\
&...
\end{aligned}
\tag{2.9.2.}
$$

$a_{1n}y_1 + a_{2n}y_2 + ... + c_{mn}y_m \leq c_n$
$y_i \geq 0, j=1,2,...,m$

gde su $y_1,,y_2,...,\ y_m$ dualne promenljive, a $G(y)$ vrednost funkcije cilja dualnog problema.

U sažetom obliku primarni i dualni problem je prikazan relacijama (2.9.3.) i (2.9.4.).

Primarni problem:

$$\min F(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1,2,\ldots,m \tag{2.9.3.}$$

$$x_j \geq 0, \quad j = 1,2,\ldots,n$$

Dualni problem:

$$\max G(y) = \sum_{i=1}^{m} b_i y_i$$

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1,2,\ldots,n \tag{2.9.4.}$$

$$y_i \geq 0, \quad i = 1,2,\ldots,m$$

Problemi primara i njemu odgovarajućeg duala mogu se napisati i vektorskom notacijom, kao što je prikazano relacijama (2.9.5.) i (2.9.6.).

Primarni problem:

$$\min F(x) = C^T \cdot X$$
$$\text{p.o.} \quad A \cdot X \geq b \tag{2.9.5.}$$
$$X \geq 0$$

Dualni problem:

$$\max G(y) = b^T \cdot Y$$
$$\text{p.o.} \quad A^T \cdot Y \leq C \tag{2.9.6.}$$
$$Y \geq 0$$

Dual ima onoliko promenljivih koliko primar ima ograničenja i onoliko ograničenja koliko primar ima promenljivih. Matrica ograničenja duala jednaka je transponovanoj matrici ograničenja primara, dok koeficijenti u funkciji cilja duala predstavljaju slobodne članove primara i obrnuto.

Napomena 2.9.1.: Slobodni članovi b_i primara nemoraju obavezno da budu nenegativni.

Primer 2.9.1. Napisati model dualnog problema ukoliko ja dat primarni.

Primarni model je:

$$\min G(y) = 2y_1 + 3y_2 + 4y_3 + y_4$$
$$8y_1 + 5y_2 + 3y_3 + y_4 \geq 400$$
$$2y_1 + y_2 + 6y_3 + 5y_4 \geq 250$$
$$5y_1 + 4y_2 + 3y_3 + 2y_4 \geq 220$$
$$6y_1 + 3y_2 + 4y_3 + 5y_4 \geq 250$$

Rešenje. Dualni model glasi:

$$\max F(x) = 400x_1 + 250x_2 + 220x_3 + 250x_4$$
$$8x_1 + 2x_2 + 5x_3 + 6x_4 \leq 2$$
$$5x_1 + x_2 + 4x_3 + 3x_4 \leq 3$$
$$3x_1 + 6x_2 + 3x_3 + 4x_4 \leq 4$$
$$1x_1 + 5x_2 + 2x_3 + 5x_4 \leq 1$$

Primer 2.9.2. Napisati model dualnog problema ukoliko ja dat primarni.

Primarni model je:

$$\min G(y) = 3y_1 + 2y_2 + 4y_3$$
$$4y_1 + 5y_2 + 5y_3 \geq 200$$
$$2y_1 + 6y_2 + y_3 \geq 450$$
$$6y_1 + y_2 + 3y_3 \geq 220$$
$$6y_1 + 3y_2 + 4y_3 \geq 250$$

Rešenje. Dualni model glasi:

$$\max F(x) = 200x_1 + 450x_2 + 220x_3 + 250x_4$$
$$4x_1 + 2x_2 + 6x_3 + 6x_4 \leq 3$$
$$5x_1 + 6x_2 + x_3 + 3x_4 \leq 2$$
$$5x_1 + x_2 + 3x_3 + 4x_4 \leq 4$$

Nakon prilagođavanja modela za primenu simpleks metode dobija se model LP:

$$\max F(x)= 200x_1 + 450x_2 + 220x_3 + 250x_4 + 0x_5 + 0\cdot x_6 + 0\cdot x_7$$

p.o.

$$4x_1 + 2x_2 + 6x_3 + 6x_4 + x_5 = 3$$
$$5x_1 + 6x_2 + x_3 + 3x_4 + x_6 = 2$$
$$5x_1 + x_2 + 3x_3 + 4x_4 + x_7 = 4$$
$$x_j \geq 0,\ j=1,2,...,7$$

Optimalno rešenje ovog problema pronađeno je od *ST*_0 – *ST*_2, koje su prikazane u tabelama od 2.9.1. – 2.9.3.

Tabela 2.9.1. *Početna (nulta) simpleks tabela ST_0*

C_b	X_b	B	200	450	220	250	0	0	0	$\theta=b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	
0	x_5	**3**	4	2	**6**	6	1	0	0	3/2
0	x_6	**2**	5	6	1	3	0	1	0	1/3
0	x_7	**4**	5	1	3	4	0	0	1	4
$F_j–c_j$		**0**	-200	-450	-220	-250	0	0	0	**150**

Tabela 2.9.2. *Simpleks tabela ST_1*

C_b	X_b	B	200	450	220	250	0	0	0	$\theta=b_i/a_{ij}$
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	
0	x_5	**7/3**	7/3	0	**17/3**	5	1	-1/3	0	7/17
450	x_2	**1/3**	5/6	1	1/6	1/2	0	1/6	0	2
0	x_7	**11/3**	25/6	0	17/6	7/2	0	-1/6	1	22/17
$F_j–c_j$		**150**	133,3	0	-145	-25	0	75	0	**59,71**

Tabela 2.9.3. *Simpleks tabela ST_2*

C_b	X_b	B	200	450	220	250	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6	x_7
220	x_3	**7/17**	7/17	0	**1**	15/17	3/17	-1/17	0
450	x_2	**9/34**	13/17	1	0	6/17	1/34	3/17	0
0	x_7	**85/34**	3	0	0	1	1/2	0	1
$F_j – c_j$		**209,71**	**234,7**	**0**	**0**	**102,9**	**52,06**	**66,47**	**0**

Rešenje u tabeli ST_2 je optimalno jer nema više pozitivnih koeficijenata ($F_j – c_j$). S obzirom da se ovde traže vrednosti y_j one se čitaju drugačije u tabeli 2.9.3. u odnosu na klasičnu simpleks tabelu. Naime, čitaju se vrednosti u redu $F_j – c_j$ i to od dopunskih promenljivih udesno (od kolone x_5) i one glase: y_1=52,06, y_2=66,47, y_3=0, a minimalna vrednost funkcije kriterijuma iznosi: min $G(y)$=209,71. Takođe, ovde se mogu očitati i dopunske promenljive primara u istom redu ($F_j – c_j$) od kolone x_1 udesno, tj. y_4=234,71, y_5=0, y_6=0 i y_7=102,9.

Primer 2.9.3. Napisati model dualnog problema ukoliko ja dat primarni.

Primarni model je:

$$\begin{aligned} &\max G(y) = 5y_1 - 2y_2 + 3y_3 \\ &2y_1 + 3y_2 + 4y_3 \le 200 \\ &3y_1 - 5y_2 + y_3 \le 450 \\ &4y_1 + y_2 \le 220 \\ &- 4y_3 \le 250 \end{aligned}$$

Rešenje. Dualni model glasi:

$$\begin{aligned} &\min F(x) = 200x_1 + 450x_2 + 220x_3 + 250x_4 \\ &2x_1 + 3x_2 + 4x_3 \ge 5 \\ &3x_1 - 5x_2 + x_3 \ge -2 \\ &4x_1 + x_2 - 4x_4 \ge 3 \end{aligned}$$

2.9.2. Nesimetrični dualni problem

Karakteristike nesimetričnog dualnog problema linearnog programiranja su:

- ograničenja primara su data u obliku nejednačina i/ili jednačina,
- nejednačine u primarnom modelu nisu istog smera (tipa),
- u dualnom modelu sva su ograničenja istog tipa,
- kad se traži max $G(y)$ sva su ograničenja tipa $\le$,
- kad se traži min $G(y)$ sva su ograničenja tipa $\ge$,
- uslov nenegativnosti ne važi za sve dualne promenljive,
- dualne promenljive koje odgovaraju jednačinama i „nesimetričnim“ nejednačinama nisu ograničene u pogledu znaka (mogu biti i negativne).

Nesimetrični problem se uvek može svesti na simetrični, pa je zbog toga dovoljno da se posveti pažnja samo na simetrični dualni problem.

Relacijom (2.9.7.) dat je primarni problem, relacijom (2.9.8.) ekvivalentni simetrični oblik primara, a relacijom (2.9.9.) njegov dualni model problema linearnog programiranja.

$$\min F(x) = \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1,2,\ldots,m-2$$
$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = m-1 \tag{2.9.7.}$$
$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = m$$
$$x_j \geq 0, \quad j = 1,2,\ldots,n$$

Kako je dat problem minimuma, potrebno je da se sva ograničenja, u primarnom modelu, transformišu u oblik nejadnačina tipa ≥, tj. da se prilagode za opšti problem minimuma. Ograničenja tipa „=“ transformišu se u oblik nejednačina, kao:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = m-1$$
$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = m-1$$

nakon svođenja ograničenja tipa ≤ na tip ≥, množenjem obe strane sa (-1), dobija:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = m-1$$
$$-\sum_{j=1}^{n} a_{ij} x_j \geq -b_i, \quad i = m-1$$

Kada se i poslednje ograničenje (za $i = m$), koje ima oblik nejednačine tipa ≤ svede na tip nejednačine ≥, množenjem koeficijentom (-1), onda se dobija ekvivalentan simetričan oblik primala, prikazan relacijom (2.9.8.).

$$\min F(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = 1,2,\ldots,m-2$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = m-1$$

$$-\sum_{j=1}^{n} a_{ij} x_j \geq -b_i, \quad i = m-1 \qquad (2.9.8.)$$

$$-\sum_{j=1}^{n} a_{ij} x_j \geq -b_i, \quad i = m$$

$$x_j \geq 0, \quad j = 1,2,\ldots,n$$

Odgovarajući dualni problem ima model dat relacijom (2.9.9.).

$$\max G(y) = \sum_{i=1}^{m} b_i y_i$$

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1,2,\ldots,n \qquad (2.9.9.)$$

$$y_i \geq 0, \quad i = 1,2,\ldots,m-2$$

Primer 2.9.4. Napisati dualni model linearnog programiranja za dati primalni model.
Primalni model je:

$$\begin{aligned}
&\min G(y) = 2y_1 + 3y_2 + 4y_3 + y_4\\
&8y_1 + 5y_2 - 3y_3 + y_4 \geq 400\\
&2y_1 + y_2 + 6y_3 + 5y_4 \leq 250\\
&5y_1 - 4y_2 + 3y_3 + 2y_4 = 220\\
&6y_1 + 3y_2 + 4y_3 - 5y_4 \geq 250
\end{aligned}$$

Rešenje. Drugo ograničenje u modelu je ograničenje tipa $\leq$, koje se transformiše u oblik nejednačina tipa $\geq$, množenjem sa koeficijentom (-1), tj:

$$-2y_1 - y_2 - 6y_3 - 5y_4 \geq -250$$

Treće ograničenje je ograničenje tipa jednačine, koje se transformiše u oblik nejednačina, kao:

$$\begin{aligned}
&5y_1 - 4y_2 + 3y_3 + 2y_4 \geq 220\\
&5y_1 - 4y_2 + 3y_3 + 2y_4 \leq 220
\end{aligned}$$

odnosno:

$$\begin{aligned}
&5y_1 - 4y_2 + 3y_3 + 2y_4 \geq 220\\
&-5y_1 + 4y_2 - 3y_3 - 2y_4 \geq -220
\end{aligned}$$

Model transformisan u opšti oblik problema minimuma dobija oblik:

$$\begin{aligned}
&\min G(y) = 2y_1 + 3y_2 + 4y_3 + y_4\\
&8y_1 + 5y_2 - 3y_3 + y_4 \geq 400\\
&-2y_1 - y_2 - 6y_3 - 5y_4 \geq -250\\
&5y_1 - 4y_2 + 3y_3 + 2y_4 \geq 220\\
&-5y_1 + 4y_2 - 3y_3 - 2y_4 \geq -220\\
&6y_1 + 3y_2 + 4y_3 - 5y_4 \geq 250
\end{aligned}$$

Dualni model glasi:

$$
\begin{aligned}
&\max F(x) = 400x_1 - 250x_2 + 220x_3 - 220x_4 + 250x_5\\
&8x_1 - 2x_2 + 5x_3 - 5x_4 + 6x_5 \le 2\\
&5x_1 - x_2 - 4x_3 + 4x_4 + 3x_5 \le 3\\
&-3x_1 - 6x_2 + 3x_3 - 3x_4 + 4x_5 \le 4\\
&x_1 - 5x_2 + 2x_3 - 2x_4 - 5x_5 \le 1
\end{aligned}
$$

2.9.3. Svojstva dualnosti

Teorija dualnosti izučava matematička svojstva odnosa primarnog problema i njegovog duala. U ovom poglavlju, biće izložena najvažnija svojstva i uočene neke njihove moguće primene. Razmatra se problem maksimizacije funkcije $F(x)$ na bilo kom skupu linearnih ograničenja (primarni problem), i njemu pridruženi model minimizacije funkcije $G(y)$ (dualni problem).

Svojstvo 2.9.1. (Slaba dualnost)

Ako je x dopustivo rešenje primara, a y dopustivo rešenje duala, tada je $F(x) \le G(y)$.

Svojstvo slabe dualnosti obezbeđuje gornju granicu $G(y)$ maksimalne vrednosti funkcije cilja primara, odnosno donju granicu $F(x)$ minimalne vrednosti funkcije cilja duala.

Primer 2.9.5. Napisati dualni model linearnog programiranja za dati primarni model.

$$
\begin{aligned}
&\max F(x) = 5x_1 + 2x_2\\
\text{p.o.}\quad &1x_1 \le 6\\
&2x_2 \le 18\\
&3x_1 + 2x_2 \le 24\\
&x_1, x_2 \ge 0
\end{aligned}
$$

Rešenje. Dualni model je:

$$
\begin{aligned}
&\min G(y) = 6y_1 + 18y_2 + 24y_3\\
\text{p.o.}\quad &1y_1 + 3y_3 \ge 5\\
&2y_2 + 2y_3 \ge 2\\
&y_1, y_2, y_3 \ge 0
\end{aligned}
$$

U primeru 2.9.5., za primarni model, može se lako proveriti da je $x=(x_1,x_2)=(6,0)$ dopustivo rešenje problema. Može se ustanoviti da je $y=(y_1,y_2,y_3)=(6,1,1/3)$, takođe, dopustivo rešenje za dualni model. U ovim tačkama funkcija cilja primara je $F(x)=30$, a funkcija cilja duala $G(y)=62$, tj. $F(x)=30<62=G(y)$. To znači da maksimum funkcije cilja, za primarni model, nije veći od 62, a minimum funkcije cilja, za dualni model, nije manji od 30.

Svojstvo 2.9.2.

a) Ako je x dopustivo rešenje primara koje nije optimalno i $F(x)=G(y)$, tada tačka y nije dopustivo rešenje duala.
b) Ako je x dopustivo rešenje primara, y dopustivo rešenje duala i $F(x)=G(y)$, tačke x i y su optimalna rešenja problema.

Svojstvo 2.9.2. b) omogućuje da se bez rešavanja modela primara i duala, nekim od metoda, utvrdi da su rešenja x i y optimalna. Dovoljno je dokazati da je $F(x)=G(y)$. Za predhodni primer 2.9.5. optimalno rešenja primarnog modela je $x^*=(x_1^*,x_2^*)=(6,3)$, a optimalno rešenja duala $y^*=(y_1^*,y_2^*,y_3^*)=(2,0,1)$, pri čemu je u tim tačkama $F(x)=G(y)=36$.

Svojstvo 2.9.3. (Jaka dualnost)

Prema svojstvu jake dualnosti, za primar – dual problem važi jedan od sledećih iskaza:

- Primar ima optimalno rešenje ako i samo ako dual ima optimalno rešenje, pri čemu su optimalne vrednosti funkcija cilja ova dva problema jednake, tj $F(x)=G(y)$.
- Primar nema rešenja ako i samo ako dual nema rešenja.
- Ako jedan problem nema dopustivo rešenje, a drugi problem ima dopustivo rešenje, on nema optimalno rešenje jer funkcija cilja nije ograničena na dopustivom skupu.

2.10. PROGRAMSKI PAKETI ZA REŠAVANJE PROBLEMA LINEARNOG PROGRAMIRANJA

Programski paketi imaju široku primenu pri rešavanju raznih zadataka liearnog programiranja. Danas postoje mnogobrojni programski paketi koji omogućavaju brzo i efikasno rešavanje raznih problema iz ove oblasti. Pomoću njih se veoma brzo može dobiti optimalno rešenje problema i izvršiti analiza osetljivosti koja ima veliki praktični značaj. Najpoznatiji razvijeni programski paketi koji se koriste u tu svrhu su LINDO, LINGO, QM for Windows, GAMS, Solver, WinQSB, WinGULF, SAS/OR, i dr. Određeni programski paketi imaju mogućnost rešavanja problema i iz drugih oblasti operacionih istraživanja, kao što je nelinearno programiranje, dinamičko programiranje, i dr.

2.10.1. Programski paket Lindo

LINDO – *__L__inear __In__teractive and __D__iscrete __O__ptimizer* – je interaktivni softverski paket koji se može koristiti za rešavanje problema linearnog programiranja, u svim aplikacijama gde je potrebno rešavati problem optimizacije. Razvijen je 1980. godine i od tada je prilagođen Windows okruženju i grafički orijentisanim programima. Softverski paket LINDO se koristi za rešavanje problema zadatih direktno sa tastature.

Elementi LINDO modela su:

1. Cilj – uvek u prvoj liniji LINDO modela, počinje izrazom ***min*** ili ***max***. Označava funkciju cilja, čiji minimum ili maksimum treba odrediti.

2. Jedna ili više varijabli – nepoznate veličine koje treba odrediti da bi se ostvario cilj.
3. Jedno ili više ograničenja – postavljenih na varijablama. U LINDO modelima ograničenjima prethodi jedna od sledećih linija: ***SUBJECT TO; SUCH THAT; S. T.; ST***; Kraj ograničenja se označava sa ***END***, što je obavezno samo u slučaju kada se koriste dodatne komande.

LINDO sintaksa:

- Ime varijable je ograničeno na 8 karaktera.
- LINDO prepoznaje sledeće operatore: + plus, – minus, < manje, > veće, = jednako, <= manje ili jednako i >= veće ili jednako.
- Operacije se izvode s leva na desno.
- Komentari mogu biti bilo gde u modelu, a prethodi im uzvičnik.
- Ograničenja i funkcija cilja mogu biti u više linija.
- LINDO nije osetljiv na veličinu slova.
- Sa desne strane jednačine ograničenja mogu biti samo konstante.
- Sa leve strane ograničenja mogu biti samo promenljive i njihovi koeficijenti.

Komande menija:

File

Save	Snimanje ulaznih podataka (modela), izveštaja ili komandnog prozora. Format u kojem ih snimamo je *.LTX – LINDO tekstualni format.
Log Output	Ako je aktivna ova opcija, sve aktivnosti u aktivnom prozoru se snimaju u tekstualni fajl (dnevnik, log).
Take Commands	Za preuzimanje LINDO komandi iz drugih programa.
Basis Save	Snimanje rešenja aktivnog modela.
Basis Read	Čitanje rešenja modela, koje je bilo sačuvano korišćenjem Basis Save komande.
Title	Prikazuje ime aktivnog modela, ukoliko mu je bilo dodeljeno.
Date	Prikaz tekućeg datuma i vremena.
Elapsed Time	Vreme proteklo u tekućoj LINDO sesiji.

Edit

Options	Uvid i izmene raznih parametara korišćenih u LINDO sesiji.
Paste Symbol	Ispisivanje svih simbola koji se mogu koristiti u LINDO modelu.

Solve

Solve	Rešava aktivni model.
Debug	Omogućava nalaženje greške u modelu.

Reports

Solution	Omogućava određivanje izgleda rešenja.
Range	Daje analizu (intervale, za koje nađeno optimalno rešenje ostaje nepromenjeno) parametara sa desne strane ograničenja i koeficijenata uz promenljive u funkciji cilja.

Parametrics	Daje rezultate promene vrednosti parametara sa desne strane ograničenja.
Statistics	Prikazuje ključnu statstiku za model u aktivnom prozoru.
Peruse	Pregled rešenja u željenom formatu (grafičkom ili tekstualnom, sa odabranim karakteristikama).
Picture	Prikaz aktivnog model u matričnoj formi.
Basis Picture	Prikaz vrsta i kolona poslednje matričnoj transformaciji Solver-a.
Tableau	Prikaz simpleks tabele aktivnog modela.
Formulation	Prikaz svih ili selektovanih delova modela.
Show Column	Prikaz selektovane kolone bez ostatka modela.

Napomen.: da bi opcije menija **Reports** bile aktivne, potrebno je da prozor modela problema bude aktivan.

Window

Open Command Window	Otvara komandni prozor za unos LINDO komandi.
Open Status Window	Otvara prozor sa rešenjem, kao posle opcije Solve.

Help

Za upoznavanje sa karakteristikama programa (kroz primere), preporučuje se korišćenje Help-a, koji ih veoma detaljno i pregledno prikazuje.

Napomena. Neke opcije u navedenim menijima nisu pomenute, jer se poznavanje njihovog značenja pretpostavlja. Odnosno, te opcije su sastavni deo prozora ostalih Windows aplikacija, pa se ne navode posebno.

Dodatne komande LINDO modela:

Pored osnovih elemenata modela, mogu se navesti i sledeće komande (posle END), kojima se proširuju mogućnosti programa:

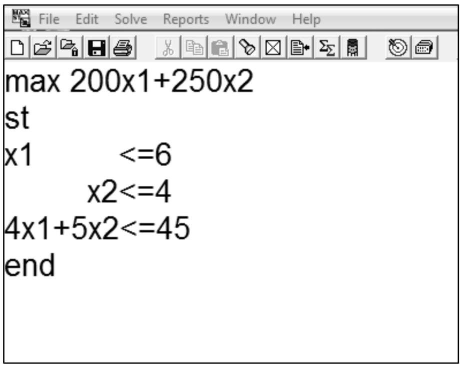
FREE *ime promenljive*	Omogućava da navedene promenljiva ima realne vrednosti, pozitivne i negativne
GIN *ime promenljive*	Ograničava vrednosti navedene promenljive na pozitivne celobrojne.
INT *ime promenljive*	Ograničava vrednosti navedene promenljive na binarne (0 ili 1).
SLB *ime promenljive*	Postavlja donju granicu promenljive (SLB X 10 – znači da će X biti veće ili jednako od 10)
SUB *ime promenljive*	Postavlja gornju granicu promenljive (SUB X 10 – znači da će X biti manje ili jednako sa 10)
QCP *ime promenljive*	naznačava prvo ograničenje u modelu kvadratnog programiranja
TITLE *naziv*	Omogućava da se modelu dodeli naziv. Naziv će biti prikazan u Reports window, korišćenjem komande Title iz File menija.

Primer 2.10.1. Kompanija proizvodi dva tipa zamrzivača – vertikalni i sandučar. Cena vertikalnog zamrzivača je 200 eura, a sandučara 250 eura.Odrediti maksimalni prihod kompanije ako je dnevni kapacitet kompanije 6 kom vertikalnih zamrzivača i 4 sandučara. Broj radnika na proizvodnji iznosi 45, a za proizvodnju vertikalnog zamrzivača je potrebno 4 radnika, a za sandučara 5 radnika.

Rešenje. Matematički model iznosi:

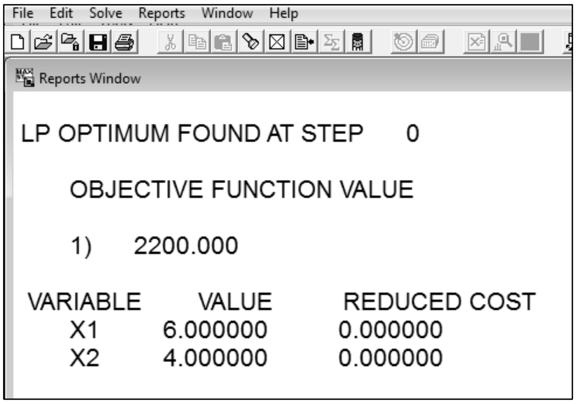
$$
\begin{array}{rl}
 & \max F(x) = 200x_1 + 250x_2 \\
\text{p.o.} & x_1 \leq 6 \\
 & x_2 \leq 4 \\
 & 4x_1 + 5x_2 \leq 45 \\
 & x_1, x_2 \geq 0
\end{array}
$$

Matematički model za LINDO programski paket ima sledeći oblik (slika 2.10.1.):



Slika 2.10.1. *Matematički model u LINDO programu*

Dobijeno rešenje pomoću LINDO programa je prikazano na slici 2.10.2.



Slika 2.10.2. *Rešenje problema u LINDO programu*

Maksimalna moguća dobit iznosi 2200eura, pri čemu proizvodnja vertikalnih zamrzivača iznosi 6 kom. (x_1=6) , a sandučara 4 kom. (x_2=4).

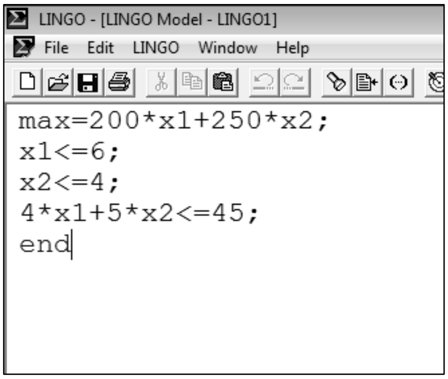
2.10.2. Programski paket Lingo

Ovaj moćni softverski alat je prvenstveno namenjen za rešavanje problema linearnog i nelinearnog programiranja. Upotreba elemenata, sintaksi i komandi u LINGO programu se razlikuju u odnosu na primenu LINDO programa, tako da uporedni prikaz rešavanja istog problema omogućava da se oni lakše uoče.

Glavne karakteristike LINGO programskog paketa su jednostavno postavljanje modela, zatim njegov matematički programski jezik koji omogućuje zapisivanje problema u prirodnom obliku koji je sličan standardnoj matematičkoj notaciji. Rezultat toga je smanjenje vremena kod izrade modela i preglednost modela linearnog, nelinearnog i celobrojnog problema napisanog u čitljivoj intuitivnoj formi. Rezultat toga je što su modeli jednostavni za postavljanje, razumevanje i održavanje. LINGO-v jezik za modeliranje uključuje i datoteku matematičkih, statističkih i finansijskih funkcija. Na osnovu toga, on ima mogućnost izrade modela koji koriste podatke direktno iz baza podataka ili proračunskih tablica (npr. Exel) čime se postiže skraćenje vremena i veća jednostavnost u manipuliranju s podacima. Takođe, moguće je izlazne podatke prebaciti u razne forme proračunskih tablica.

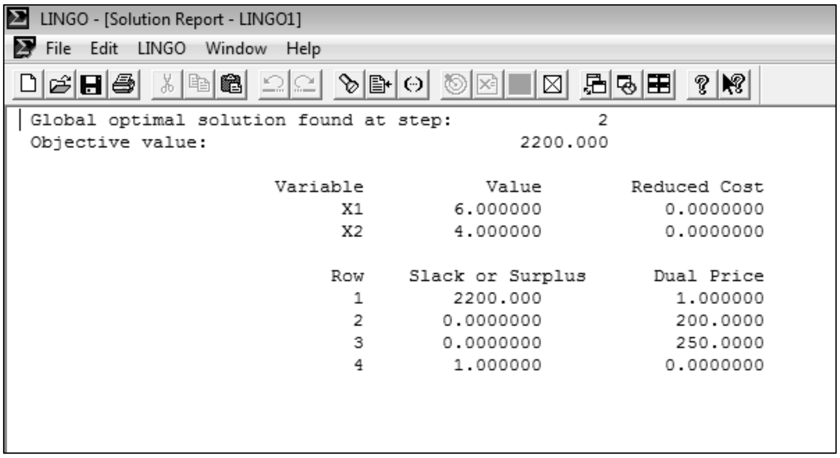
Primer 2.10.2. Problem linearnog programiranja 2.10.1. rešiti pomoću programskog paketa LINGO.

Rešenje. Na slici 2.10.3. je prikazan matematički model problema u LINGO programskom paketu.



Slika 2.10.3. *Matematički model u LINGO programu*

Dobijeno rešenje pomoću LINGO programa je prikazano na slici 2.10.4.



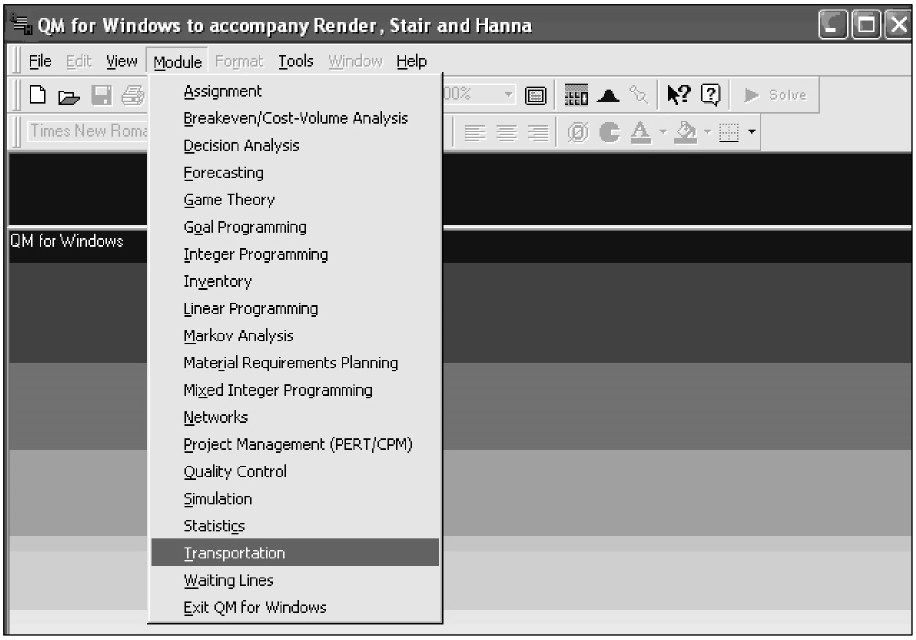
Slika 2.10.4. *Rešenje problema u LINGO programu*

Dobijen je identični rezultat kao kod primera 2.10.1.

2.10.3. Programski paket QM for Windows

Programski paket "*QM for Windows*" (**Q**uantitative **M**ethod **for Windows**) je razvio profesor Howard Weiss. On spada u omiljene softverske pakete za kvantitativne tehnike, koji koriste studenti. Njegova primena ne zahteva jake računarske konfiguracije i vrlo je lako njime rukovati.

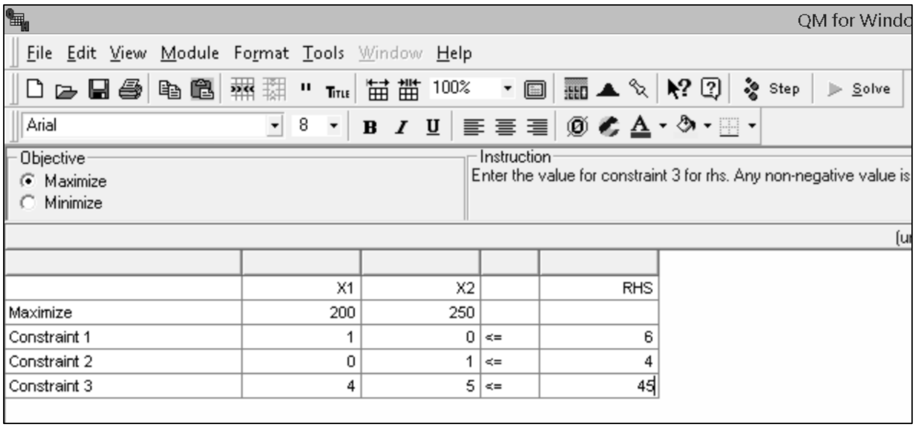
"QM for Windows" program radi pod Windows operativnim sistemom, tako da je korisnički orjentisan (User Frendly) i korišćenje je znatno olakšano sa osnovnim poznavanjem rada sa Windows okruženjem. Konkretnije, nakon instaliranja programa, postavlja se u liniji sa menijima meni "Module", gde je moguće izabrati jednu od ponuđenih opcija za kvantitativno proračunavanje raznih problema. U ponudi je veliki broj metoda, kao što se može uočiti na slici 2.10.5.



Slika 2.10.5. *Ponuda menija „Module" u QM for Windows programu*

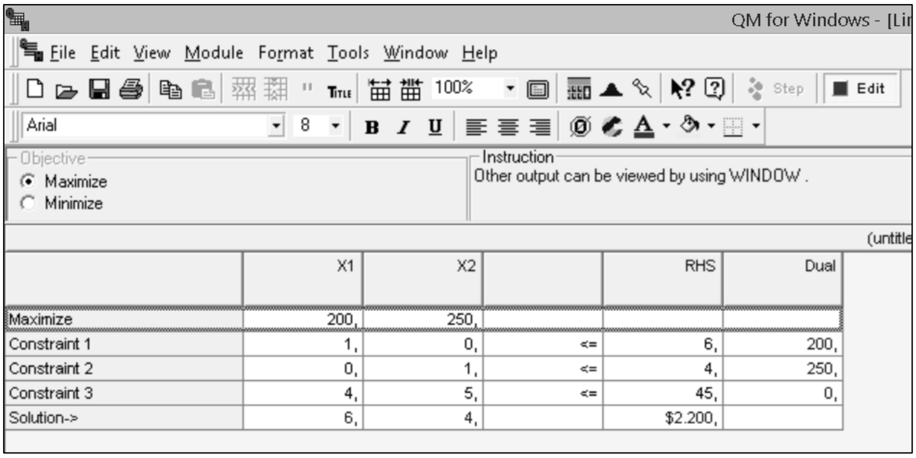
Primer 2.10.3. Problem linearnog programiranja 2.10.1. rešiti pomoću programskog paketa QM for Windows.

Rešenje. Na slici 2.10.6. je prikazan matematički model problema u QM for Windows programskom paketu.



Slika 2.10.6. *Matematički model u QM for Windows programu*

Dobijeno rešenje pomoću QM for Windows programa je prikazano na slici 2.10.7.



Slika 2.10.7. *Rešenje problema u QM for Windows programu*

Kao što se vidi, dobijen je identični rezultat kao kod prethodno primenjenih programskih paketa.

2.11. POSTOPTIMALNA ANALIZA (ANALIZA OSETLJIVOSTI)

Prilikom rešavanja zadataka linearnog programiranja polazi se od postojećih podataka (ulaznih parametara) vezanih za dati problem. Ukoliko se u modelu linearnog programiranja promeni neki od ulaznih parametara nakon određivanja optimalnog rešenja, postavlja se pitanje da li nastale promene dovode do promene strukture vektorske baze na osnovu koje je određeno optimalno rešenje. Iz tog razloga, veoma je važno odrediti koliko je model osetljiv na ove promene ulaznih parametara. Postoptimalnom analizom se može ispitati do koje granice se mogu menjati ulazni parametri problema linearnog programiranja, a da to ne utiče na strukturu optimalnog rešenja. Drugim rečima, korišćenjem postupka postoptimalne analize moguće je ispitati optimalnost prethodno određenog rešenja pod promenjenim uslovima.

U cilju identifikacije ulaznih parametara koji se mogu menjati, polazi se od matematičkog modela u vektorskom obliku, kako je ranije dat:

$$F(x) = C \cdot X = \begin{bmatrix} c_1 & c_2 & \cdots & c_n & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} \tag{2.11.1.}$$

$$A \cdot X \overset{\geq}{\underset{\leq}{=}} B$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ a_{31} & a_{32} & \cdots & a_{3n} & 0 & 0 & \cdots & 0 \\ \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} \overset{\geq}{\underset{\leq}{=}} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_m \end{bmatrix} \tag{2.11.2.}$$

Na osnovu izraza (2.11.1.) i (2.11.2.) može se zaključiti da su u matematičkom modelu moguće sledeće varijacije ulaznih parametara:

- promene u vektor – vrsti *C*, što najčešće ukazuje na promene cena, troškova, vremena, dužina transporta i dr.
- promene u vektoru *B*, što najčešće ukazuje na promene nekog od ograničenja, kao što je kapacitet resursa, količina resursa i dr.
- promene u matrici *A*, što najčešće ukazuje na promene normativa, vremena i drugih zahteva i dr.

Cilj postoptimalne analize je da se odredi u kojim granicama smeju da se menjaju ulazni parametri matematičkog modela, a da to ne dovede do promene optimalnog plana, odnosno rešenja problema.

Savremeni porgramski paketi imaju opciju vršenja postoptimalne analize, što omogućava dobijanje brzog rezultata.

Primer 2.11.1. Kompanija proizvodi tri tipa stolova – A, B i C. Proizvodni proces se odvija na tri mašine – M1, M2 i M3. Raspoloživost mašina na mesečnom nivou iznosi: mašina M1 – 160h, mašina M2 – 145h i mašina M3 – 155h. Za proizvodnju stola tipa A je potrebno sledeće vreme obrade: 3h na mašini M1, 2h na mašini M2 i 5h na mašini M3. Za proizvodnju stola tipa B je potrebno: 2h na mašini M1, 4h na mašini M2 i 4h na mašini M3. Za proizvodnju stola tipa C je potrebno: 3h na mašini M1, 1h na mašini M2 i 4h na mašini M3. Ispitivanje tržište je pokazalo da za stolove tipa A ne postoji ograničenje plasmana na tržištu, a stolove tipa B i C je moguće mesečno plasirati najviše do po 15 kom. Dobit po jedinici proizvoda iznosi: 12 n.j. za sto tipa A. 11 n.j. za sto tipa B i 14 n.j. za sto tipa C.

Odrediti:

a. matematički model koji definiše optimalni proizvodni program koji omogućava maksimalnu dobit,
b. optimalni program proizvodnje pomoću programskog paketa LINDO, i
c. postoptimalnu analizu optimalnog programa proizvodnje pomoću programskog paketa LINDO.

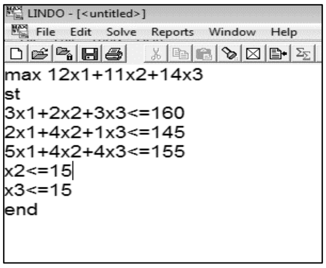
Rešenje.

a. Matematički model iznosi:

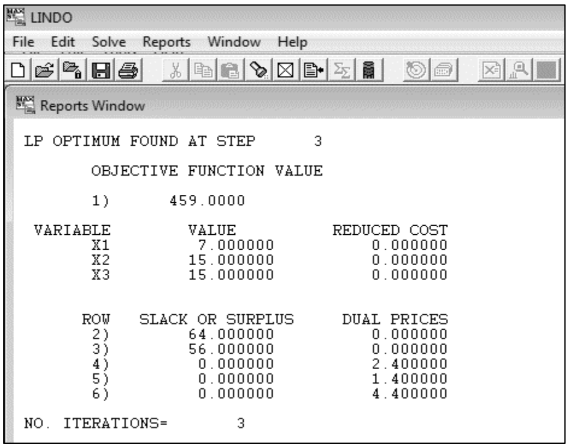
$$\max F(x) = 12x_1 + 11x_2 + 14x_3$$

$$\begin{aligned} \text{p.o.} \quad & 3x_1 + 2x_2 + 3x_3 \leq 160 \\ & 2x_1 + 4x_2 + 1x_3 \leq 145 \\ & 5x_1 + 4x_2 + 4x_3 \leq 155 \\ & x_2 \leq 15 \\ & x_3 \leq 15 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

b. Optimalni program proizvodnje pomoću programskog paketa LINDO (slike 2.11.1. i 2.11.2.):



Slika 2.11.1. *Izgled ekrana sa postavljenim matematičkim problemom*



Slika 2.11.2. *Izgled ekrana sa rešenjem problema*

Maksimalna dobit koju može da ostvari kompanija iznosi 459 n.j./mes. uz uslov da proizvedu 7 kom. stolova tipa A, 15 kom. stolova tipa B i 15 kom. stolova tipa C.

c. Postoptimalna analiza optimalnog programa proizvodnje pomoću programskog paketa LINDO (slika 2.11.3.)

```
RANGES IN WHICH THE BASIS IS UNCHANGED:

                          OBJ COEFFICIENT RANGES
VARIABLE        CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
      X1      12.000000         1.750000        12.000000
      X2      11.000000         INFINITY         1.400000
      X3      14.000000         INFINITY         4.400000

                          RIGHTHAND SIDE RANGES
     ROW        CURRENT        ALLOWABLE        ALLOWABLE
                   RHS          INCREASE         DECREASE
       2     160.000000         INFINITY        64.000000
       3     145.000000         INFINITY        56.000000
       4     155.000000       106.666664        35.000000
       5      15.000000         8.750000        15.000000
       6      15.000000         8.750000        15.000000
```

Slika 2.11.3. *Izgled ekrana posle izvršene postoptimalne analize*

Programski paket je našao dozvoljene promene vrednosti vektor – vrste C (object coefficient ranges) i dozvoljene promene vrednosti ulaznih parametara vektora ograničenja B (Righthand Side Ranges). Granice u okviru kojih je dozvoljeno povećanje i smanjenje ulaznih parametara, pod uslovom da nađeno rešenje (plan) ostane optimalno, iznose:

- dobit po proizvodu x_1, odnosno koeficijent c_1 sme se povećati za 1,75 n.j. i smanjiti za najviše 12 n.j.,
- dobit po proizvodu x_2, odnosno koeficijent c_2 sme se povećati beskonačno i smanjiti za najviše 1,4 n.j.,
- dobit po proizvodu x_3, odnosno koeficijent c_3 sme se povećati beskonačno i smanjiti za najviše 4,4 n.j.,
- mesečni kapacitet (raspoloživost) mašine M1 se sme povećati beskonačno i smanjiti najviše za 64 h,
- mesečni kapacitet mašine M2 se sme povećati beskonačno i smanjiti najviše za 56 h,
- mesečni kapacitet mašine M3 se sme povećati najviše za 106,67 h i smanjiti najviše za 35 h,
- plasman stolova tipa B se može povećati za najviše 8,75 kom. i smanjiti najviše za 15 kom., i
- plasman stolova tipa C se može povećati za najviše 8,75 kom. i smanjiti najviše za 15 kom.

Sve dok se ulazni parametri matematičkog modela menjaju u ovim granicama, nađeno rešenje ostaje optimalno.

2.12. TRANSPORTNI PROBLEM

Izučavanje problema transporta primenom analitičkih metoda datira iz perioda pedesetih godina prošlog veka. Naziv potiče još iz vremena njegovog postanka 1941. godine kada su transportni problemi poslužili da se konstruiše prvi od matematičkih problema u linearnom programiranju, koji se kasnije primenjivao u raznim oblastima ljudske delatnosti. Neki specijalni slučajevi transportnog zadatka su izučavani još pre pojave radova iz linearnog programiranja. Transportni zadatak je prvi put uočen u radovima ruskog matematičara L.V. Kantorovića "Matematičke metode u organizaciji i planiranju proizvodnje" iz 1939. godine. Prvu strogu formulaciju transportnog zadatka dao je američki matematičar F.F.

Hitchcook. Dantzig je 1951. zasnovao metod za rešavanje transportnog zadatka koje je baziran na simpleks metodu. Već u periodu 1953-1955. nastali su novi metodi za rešavanje transportnog zadatka koji su mnogo efikasniji od simpleks metoda.

Razvojem metodologije LP pokazano je da su transportni problemi specijalan slučaj zadataka LP, bez obzira što su neki od njih ranije postavljeni i rešeni. Specifičnost transportnih problema kao zadataka LP ne ogleda se na funkciji cilja $F(x)$ već u skupu ograničenja L, gde se pojavljuju izvesna uprošćenja koeficijenata matrice A skupa ograničenja, koji se za razliku od drugih slučajeva, izražavaju u vrednostima nula ili jedan.

Analitički metodi transporta u najvećem broju slučajeva vezuju se za izbor najpovoljnije varijante transporta pri kojoj su troškovi minimalni u odnosu na određenu saobraćajnu mrežu i transportna sredstva. Danas to više nisu jedini zadaci koji se rešavaju kao transportni problemi. Sve češće se tome podvrgavaju i zadaci optimalnog razmeštaja mašina, postrojenja, pomoćnih službi, skladišta, servisa ili energetskih objekata, zadaci raspodele prevoznih sredstava na korisnike, zadaci optimalne lokacije novih pogona, zadaci najpovoljnijeg izbora radnika za obavljanje određenih poslova, i drugog sa ciljem postizanja veće ekonomičnosti rada i vremena. Sve su to zadaci koji se svode na rešavanje različitih varijanti transportnog problema. Mnoga specifična pitanja ovog problema još uvek su prisutna u velikom broju naučnih rasprava i radova koji se bave praktičnom primenom analitičkih metoda kod rešavanja transportnih problema.

Može se istaći da je u svim ovim problemima zajedničko sledeće:

- uvek se radi o prevozu (ili raspodeli) jednog homogenog proizvoda,
- transport se vrši iz više izvora na veći broj lokacija,
- pronađeno rešenje je optimalno za sve učesnike, posmatrane zajedno.

Transportni problem (TP) linearnog programiranja je problem minimizacije ukupnih troškova transporta: resursa, putnika, energije, informacije itd.

U osnovnom modelu TP pretpostavka je da su poznati:

- količina resursa koju poseduju izvori (proizvođači, centri ponude, ishodišta, magacini, skladišta, otpremne stanice), a koji je po svojoj prirodi jednorodna (homogena),
- količina resursa koju potražuju ponori (potrošači, primaoci, prodavnice, odredišta, prijemne stanice i slično), koju je potrebno distribuirati, a koja je po svojoj prirodi takođe jednorodna (homgena),
- cene transporta po jedinici robe od određenog izvora do određenog odredišta.

2.12.1. Opšti model transportnog problema

Transportnim zadatkom se može odrediti optimalan plan prevoženja jedne vrste robe ako je dato:

- broj otpremnih stanica (OS) – izvora, proizvodnih stanica ili centara, odakle treba organizovati prevoz robe,
- broj prijemnih stanica (PS) – ponora, potrošačkih stanica ili centara, u koje treba dopremiti robu iz otpremnih stanica,
- ukupna količina robe koju treba prevesti iz otpremnih u prijemne stanice,
- cene prevoza po jedinici robe od svih otpremnih do svih prijemnih stanica.

U transportnom zadatku se mogu minimizirati ukupni troškovi prevoza, vreme prevoženja, itd. Pod *optimalnim planom* prevoženja podrazumevamo onaj plan prevoženja robe od otpremnih do prijemnih stanica kojim se *minimizira* ukupna cena prevoza.

2.12.1.1. Otvoreni i zatvoreni transportni problem

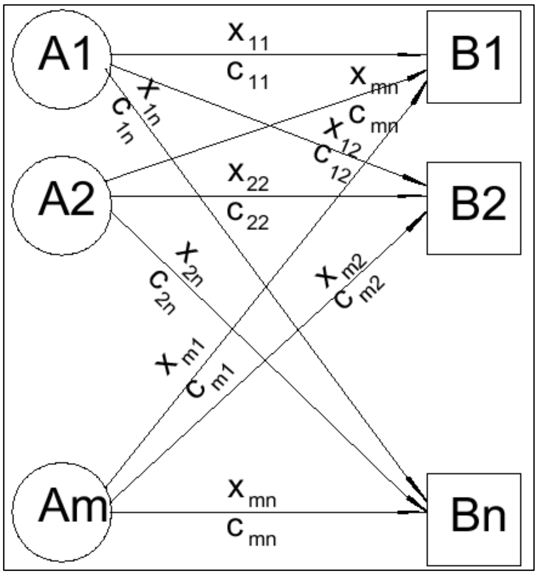
Neka je dato m otpremnih i n prijemnih stanica. Otpremne stanice ćemo obeležiti slovima A_1, A_2, ..., A_m, a prijemne sa B_1, B_2, ..., B_n. Količinu robe (iste vrste) u otpremnim stanicama obeležimo sa a_1, a_2,... , a_m, a potrebe prijemnih stanica sa b_1, b_2,... , b_n. Veličine a_1, a_2,... , a_m nazivaju se *ponude* otpremnih stanica, a veličine b_1, b_2,... , b_n *potražnje* prijemnih stanica. Ukoliko ove veličine, koje mogu biti izražene u nekim jedinicama (kilogramima, tonama, vagonima, kilometrima i dr.), zadovoljavaju jednakost:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \qquad (2.12.1.)$$

onda se transportni zadatak naziva *zatvorenim* (slika 2.12.1.). Ako je ukupna ponuda robe u otpremnim stanicama veća od ukupne potražnje te robe u prijemnim stanicama, ili obrnuto, transportni zadatak se naziva *otvorenim*.

Označimo sa c_{ij}, $i=1,..., m$, $j=1,..., n$ cenu prevoza jedinice robe između stanica A_i i B_j, a sa x_{ij} – broj jedinica robe koje treba prevesti iz A_i u B_j. Za ovako formulisan zatvoren model transportnog zadatka, podaci se mogu prikazati tabelarno, kao što je prikazano u tabeli 2.12.1. Tabela sadrži sve podatke koji definišu transportni problem, a na osnovu kojih se može formirati opšti matmatički model transportnog problema.

Otpremne stanice (OS) i njihove ponude naznačene su u pretkoloni, a prijemne stanice (PS) i njihove potražnje u zaglavlju tabele 2.12.1. U levim gornjim uglovima polja ove tabele unete su cene prevoza c_{ij}, a u desnim donjim uglovima - veličine prevoza x_{ij}.



Slika 2.12.1. *Šematski prikaz zatvorenog transportnog problema (Jovanović, 2005)*

Tabela 2.12.1. *Opšti model zatvorenog transportnog zadatka–prvi način prikazivanja*

Potrošači ***Proizvođači***	P_1 (b_1)	P_2 (b_2)	...	P_n (b_n)
I_1 (a_1)	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}
I_2 (a_2)	c_{21} x_{21}	c_{22} x_{22}	...	c_{2n} x_{2n}
...	...	...	...	...
I_m (a_m)	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}

U tabeli 2.12.2. prikazana je druga varijanta ispisivanja ovako formulisanog zatvorenog transportnog zadatka.

Tabela 2.12.2. *Opšti model zatvorenog transportnog zadatka–drugi način prikazivanja*

Proizvođači	***Potrošači***				***Kapaciteti***
	P_1	P_2	...	P_n	
I_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}	a_1
I_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2n} x_{2n}	a_2
...	...	...	...	...	...
I_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}	a_m
Potražnja	b_1	b_2	...	b_n	$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

gde su:

- x_{ij} – količina resursa koja treba da se transportuje od I_i do P_j,
- c_{ij} – jedinična cena transporta od I_i do P_j,
- I_i – izvori, proizvođači, ishodišta, centri ponude, otpremne stanice (OS),
- P_j – ponori, potrošači, odredišta, potrošački centri, prijemne stanice (PS),
- a_i – kapacitet i-tog izvora, ishodišta,
- b_j – kapacitet j-tog ponora, odredišta.

Veličine x_{ij}, $i = 1,..,m$, $j = 1,...,n$ čine dopustivi plan prevoženja ako zadovoljavaju ograničenja:

$$\sum_{j=1}^{n} x_{ij} = a_i,\ i = 1,\dots,m \tag{2.12.2.}$$

$$\sum_{i=1}^{m} x_{ij} = b_j,\ j = 1,\dots,n \tag{2.12.3.}$$

$$x_{ij} \geq 0, i = 1,\dots,m, \quad j = 1,\dots,n \tag{2.12.4.}$$

Teorema 2.12.1. Uslov (2.12.1.) je potreban i dovoljan uslov da sistem jednačina (2.11.2.) i (2.12.3.) bude saglasan.

Dokaz. Iz jednačina (2.1.2) i (2.1.3) sledi:

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} b_j,$$

što označava da je uslov (2.12.1.) potreban uslov za saglasnost sistema jednačina (2.12.2.) i (2.12.3.). Da bi se dokazalo da je uslov (2.12.1.) dovoljan uslov saglasnosti sistema jednačina (2.12.2.) i (2.12.3.), dokazaće se da su, u slučaju da važi uslov (2.12.1.), vrednosti promenljivih definisane sa:

$$x_{ij} = \frac{a_i b_j}{\sum_{i=1}^{m} a_i}, \quad i = 1,\dots,m, \quad j = 1,\dots,n$$

rešenje sistema jednačina (2.12.2.) i (2.12.3.). Shodno tome, ako važi uslov (2.12.1.), za ovako definisane vrednosti x_{ij} dobija se uslov (2.12.2.):

$$\sum_{j=1}^{n} x_{ij} = \frac{a_i \sum_{j=1}^{n} b_j}{\sum_{i=1}^{m} a_i} = a_i$$

Analogno se dobija uslov (2.12.3.):

$$\sum_{i=1}^{m} x_{ij} = \frac{b_j \sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} a_i} = b_j .$$

Zbog uslova (2.12.1.) jednačine (2.12.2.) i (2.12.3.), kojih ima $m+n$, nisu nezavisne. Sabirajući jednačine (2.12.2.), a zatim jednačine (2.12.3.), dobija se isti rezultat zbog uslova (2.12.1.), što znači da je broj linearno nezavisnih veza jednak $m+n-1$, pa postoji $\boldsymbol{m+n-1}$ zavisnih promenljivih, koje se nazivaju *bazične promenljive*. Sve ostale su nezavisne i zovu se *slobodne promenljive*. Slobodnih promenljivih ima:

$$n \cdot m - (m+n-1) = (m-1)\cdot(n-1).$$

Bazične promenljive mogu da se izraze pomoću slobodnih promenljivih. Dopustivo rešenje (x_{11}, x_{12},... , x_{mn}) naziva se *bazično rešenje*, ako je u njemu broj pozitivnih veličina x_{ij} ne veći od $m+n-1$, dok su ostale promenljive x_{ij} jednake nuli. To znači da u svakom bazičnom rešenju nema više od $m+n-1$ prevoženja. Bazično rešenje se naziva optimalnim *rešenjem* ako minimizira ukupne troškove prevoza, tj. minimizira funkciju cilja (kriterijuma):

$$F(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\ x_{ij} \qquad (2.12.5.)$$

Matematička formulacija postavljenog transportnog problema sastoji se u nalaženju minimuma funkcije cilja (2.12.5), a da pri tome bude zadovoljen skup ograničenja (2.12.2.) i (2.12.3.), tako da je:

$$\begin{aligned}\min\ F(x) = \ & c_{11}x_{11}+ c_{12}x_{12}+ c_{13}x_{13}+ \dots + c_{1n}x_{1n}+ \\ & c_{21}x_{21}+ c_{22}x_{22}+ c_{23}x_{23}+ \dots + c_{2n}x_{2n}+ \\ & \dots \\ & c_{m1}x_{m1}+ c_{m2}x_{m2}+ c_{m3}x_{m3}+ \dots + c_{mn}x_{mn}+\end{aligned}$$

Ograničenja po redovima (kapaciteti izvora):

$$\begin{aligned} x_{11} + x_{12} + \dots + x_{1n} &= a_1 \\ x_{21} + x_{22} + \dots + x_{2n} &= a_2 \\ &\dots \\ x_{m1} + x_{m2} + \dots + x_{mn} &= a_m \end{aligned}$$

Ograničenja po kolonama (kapaciteti ponora):

$$\begin{aligned} x_{11} + x_{21} + \dots + x_{n1} &= b_1 \\ x_{12} + x_{22} + \dots + x_{n2} &= b_2 \\ &\dots \\ x_{1n} + x_{2n} + \dots + x_{nm} &= b_n. \end{aligned}$$

Pri tome nepoznate vrednosti koje se traže moraju biti nenegativne x_{ij}>=0, zato što se radi o fizičkim veličinama, odnosno o količinama transportovanog materijala.

Teorema 2.12.2. Svaki transportni zadatak (2.12.1.) – (2.12.5.) sadrži optimalno rešenje.

Dokaz. Kako je skup dopustivih rešenja konačan, to funkcija cilja (ukupni troškovi prevoza) mora u jednom od tih rešenja da uzme najmanju vrednost.

Teorema 2.12.3. Ako su veličine a_i i b_j, $i = 1,..., m$, $j = 1,..., n$ celi brojevi, onda su i bazične promenljive u proizvoljnom rešenju takođe celi brojevi.

Formulišući zadatak transportnog problema, koji je zatvorenog tipa, pošlo se od pretpostavke da je ukupna ponuda svih proizvođača jednaka ukupnim potrebama svih potrošača. Ukoliko je ta pretpostavka narušena, tj. da su ukupne ponude svih proizvođača veće nego što je konzumna moć svih potrošača, ili obrnuto, radi se o otvorenom transportnom problemu.

Pretpostavka o ovoj nejednakosti uopšte ne smanjuje opštost transportnog problema i mogućnost njegove primene. Otvoreni transportni model se, uvodenjem jedne nove kolone ili novog reda, jednostavno prevodi u zatvoreni transportni model. Kod svođenja otvorenog na zatvoreni model, postupa se na sledeći način:

Ako je ukupna ponuda proizvođača veća od ukupne tražnje potrošača, tj. ako je:

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \tag{2.12.6.}$$

tada se postojećim odredištima dodaje još jedno, nepostojeće odredište, sa potražnjom koja je jednaka:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \tag{2.12.7.}$$

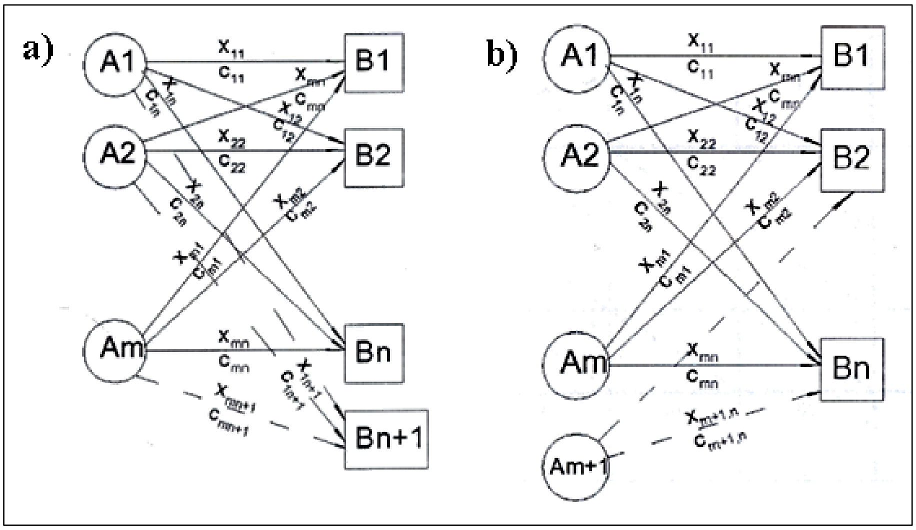
Ukoliko je ukupna tražnja potrošača veća od ukupne ponude proizvođača, tj.:

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \tag{2.12.8.}$$

tada se postojećim proizvođačima dodaje jedan nepostojeći proizvođač, sa ponudom koja je jednaka:

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i \tag{2.12.9.}$$

Šematski prikaz otvorenog transportnog problema, gde su ukupne ponude svih proizvođača veće nego što je konzumna moć svih potrošača, dat je na slici 2.12.2. a), a gde su ukupne ponude svih proizvođača manje nego što je konzumna moć svih potrošača, dat je na slici 2.12.2. b).



Slika 2.12.2. *Otvoreni model transportnog problema. a) ponuda veća od potražnje. b) potražnja veća od ponude (Jovanović, 2005)*

Na ovaj način obezbeđena je tražena jednakost, pa se može rešiti transportni problem. Naravno, transportni troškovi do nepostojećih potrošača, odnosno od nepostojećih proizvođača, uvek će biti jednaki nuli, tj. biće:

$$C_{i,\,n+1} = 0, \quad i = 1,2,...,m \qquad (2.12.10.)$$

ili

$$C_{m+1,\,j} = 0, \quad j = 1,2,...,n.$$

Iz napred formulisanog opšteg modela transportnog problema (TP) očigledno je da se radi o problemu linearnog programiranja. To dalje znači da se ovaj problem može rešiti već poznatom simpleks metodom. Međutim, specijalna struktura transportnog problema omogućuje da se problem reši mnogo jednostavnijim postupkom. Posmatranjem matrice sistema ograničenja transportnog problema, može se videti da su svi koeficijenti uz nepoznate jednaki jedinici. Dalje, kako se svaka nepoznata x_{ij} pojavljuje samo u dve jednačine, to matrica, pored jedinica, sadrži i veliki broj nula, sto dalje pojednostavljuje postupak rešavanja transportnog problema.

Pored primarnog transportnog problema, formulisanog matematičkim modelom (2.12.2.) do (2.12.5.), može se formulisati i njemu odgovarajući ***dualni problem***. Nepoznate u dualnom problemu se označavaju sa:

$$u_i, \quad i = 1, 2,..., m$$
$$v_j, \quad j = 1, 2,..., n$$

Pa model dualnog problema ima *funkciju cilja* (kriterijuma):

$$\max G(x) = \sum_{i=1}^{m} a_i \cdot u_i + \sum_{j=1}^{n} b_j \cdot v_j \qquad (2.12.11.)$$

i sistem *ograničenja*:

$$u_i + v_j \leq c_{ij}, \ i= 1,2,..,m; \ j= 1,2,...,n \qquad (2.12.12.)$$

Kako sistem ograničenja primarnog problema sadrži samo jednačine, to nema nikakvih ograda u pogledu vrednosti dualnih promenljivih, tj. one mogu biti i negativne. Međutim, na osnovu veze, koja postoji između realnih promenljivih jednog problema i izravnjavajućih promenljivih drugog problema, sistem ograničenja (2.12.12.) može se razložiti na dva podsistema. Tako se dobija sledeći sistem ograničenja dualnog problema:

$$\text{za} \quad x_{ij} > 0, \ u_i + v_j = c_{ij}, \qquad (2.12.13.)$$

$$\text{za} \quad x_{ij} = 0, \ u_i + v_j \leq c_{ij}, \qquad (2.12.14.)$$

Kada se levoj strani nejednačine (2.12.14.) doda izravnavajuća dualna promenljiva Δ_{ij}, pa se dobijena jednačina rešava po promenljivoj Δ_{ij}, dobija se:

$$\Delta_{ij} = c_{ij} - (u_i + v_j), \ \text{za} \ x_{ij} = 0. \qquad (2.12.15.)$$

Relacija (2.12.15.) biće korišćena prilikom pronalaženja optimalnog rešenja transportnog problema, pa će se onda ukazati na njen značaj i funkciju.

Zatvoreni model transportnog zadatka je najjednostavniji slučaj transportnih zadataka linearnog programiranja. Druge varijante transportnog zadatka zasnivaju se na zatvorenom modelu.

2.12.2. Određivanje početnog (baznog) rešenja

Rešavanja transportnog zadatka svodi se na određivanje početnog (baznog) rešenja, pomoću jednog od poznatih računskih postupaka – algoritama. To predstavlja prvu etapu u rešavanju transportnog zadatka. Ako se u ovoj prvoj etapi ne dobije optimalno rešenje, prelazi se na drugu etapu. U drugoj etapi se određenim iterativnim postupkom, pomoću algoritma za nalaženje optimalnog rešenja, poboljšava početno rešenje tako što se prelazi sa prvog bazičnog rešenja na bazična rešenja koja su bliža optimalnom rešenju, sve dok se ne dobije optimalno rešenje. Razvijeni su i algoritmi kod kojih nije potrebno pronalaziti početno rešenje, ono je već optimalno.

Postoji više metoda za određivanje početnog bazičnog rešenja, kao što su:

1. Dijagonalni metod – pravilo severozapadnog ugla – gornji levi ugao,
2. Metod minimalnih cena u redovima,
3. Metod minimalnih cena u kolonama,
4. Metod minimalnih cena u matrici – metod (najmanjih) jediničnih koeficijenata,
5. Vogel-ov aproksimativni metod,
6. Vogel – kordin postupak,
7. Metod dvojnog prvenstva – dvostrukog precrtavanja.

2.12.2.1. Dijagonalni metod – metod severozapadnog ugla

Ovaj metod se često naziva i metod “gornji levi ugao” ili “metod severozapadnog ugla”. Detalji ovog metoda najbolje se objašnjavaju na konkretnom zadatku. U tabeli 2.12.3. date su količine robe koje treba otpremiti iz otpremnih stanica (OS), količine robe koje se traže u prijemnim stanicama (PS), kao i cene prevoza jedinice robe od svake otpremne do svake prijemne stanice. Svako polje tabele je označeno sa (i, j), $i=1, ..., m$, $j=1, ..., n$. U levom gornjem uglu svakog takvog polja (i, j) tabele upisana je cena prevoza c_{ij} -prevoza jedinice robe iz i-te otpremne u j-tu prijemnu stanicu. U desnom donjem uglu upisana je vrednost promenljive x_{ij}, tj. upisana je količina robe koju treba prevesti iz i-te otpremne u j-tu prijemnu stanicu. U našem primeru ima $m=4$ otpremnih i $n=5$ prijemnih stanica. Ukupna količina robe koja se nudi u otpremnim stanicama je:

$$\sum_{i=1}^{m} a_i = a_1 + a_2 + a_3 + a_4 = 36 + 23 + 29 + 12 = 100,$$

i jednaka je ukupnoj količini robe koja se traži u prijemnim stanicama:

$$\sum_{j=1}^{n} b_j = b_1 + b_2 + b_3 + b_4 + b_5 = 13 + 24 + 15 + 21 + 27 = 100.$$

Može se zaključiti da se radi o *zatvorenom modelu* transportnog problema, jer važi:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j .$$

Tabela 2.12.3. *Početni problem transportnog zadatka*

OS \ PS	**PS**	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5**	**12**	**1**	**4**	**13**
A_2	**23**	**7**	**8**	**14**	**6**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Prema dijagonalnom metodu najpre se određuje vrednost promenljive x_{11}:

$$x_{11} = \min\{a_1, b_1\} = \min\{36,13\} = 13.$$

Kao što se vidi, prva otpremna stanica A_1 može da isporuči 36 jedinica robe, a prva prijemna stanica traži 13 jedinica te robe. Uzimajući $x_{11} = 13$, podmiruje se potreba prve prijemne stanice u potpunosti. Ovu vrednost za x_{11} se upisuje u donjem desnom uglu polja (1,1), kao što je prikazano u tabeli 2.12.4.

Tabela 2.12.4. *Dijagonalni metod – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** 13	**12**	**1**	**4**	**13**
A_2	**23**	**7**	**8**	**14**	**6**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Otpremna stanica A_1 može da isporuči još 36 – 13 = 23 jedinica robe preostalim prijemnim stanicama. Prijemnoj stanici B_2 se daje najviše što se može, tj. veličina x_{12} se određuje iz jednakosti:

$$x_{12} = \min\{36\text{-}13,24\} = \min\{23,24\} = 23.$$

Na ovaj način količina robe prve otpremne stanice je iscrpljena, pa su veličine x_{13}, x_{14} i x_{15} jednake nuli. Nule se iz praktičnih razloga ne upisuju. Vrednost za x_{12} se upisuje u donjem desnom uglu polja (1,2), kao što je prikazano u tabeli 2.12.5.

Tabela 2.12.5. *Dijagonalni metod – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** 13	**12** 23	**1**	**4**	**13**
A_2	**23**	**7**	**8**	**14**	**6**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Prva prijemna stanica je zadovoljena, a promenljive x_{2j}, j=1,2,3,4,5 se u drugoj vrsti (redu) tabele, određuje na sledeći način:

$x_{21} =$ 0 – jer je prethodno podmirena potreba prve prijemne stanice

$x_{22} = \min\{23, 24\text{-}23\} = \min\{23,1\} = 1;$

$x_{23} = \min\{23\text{-}1, 15\} = \min\{22,15\} = 15;$

$x_{24} = \min\{23\text{-}1\text{-}15, 21\} = \min\{7,21\} = 7;$

Vrednost za x_{22}, x_{23}, x_{24} se upisuje u donjem desnom uglu polja (2,2; 2,3; 2,4), kao što je prikazano u tabelama 2.12.6. – 2.12.8.

Tabele 2.12.6. – 2.12.7. *Dijagonalni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12** **23**	**1**	**4**	**13**
A_2	**23**	**7**	**8** **1**	**14**	**6**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12** **23**	**1**	**4**	**13**
A_2	**23**	**7**	**8** **1**	**14** **15**	**6**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Tabela 2.12.8. *Dijagonalni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12** **23**	**1**	**4**	**13**
A_2	**23**	**7**	**8** **1**	**14** **15**	**6** **7**	**5**
A_3	**29**	**15**	**4**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Druga prijemna stanica je zadovoljena, a promenljive x_{3j}, $j = 1, 2, 3, 4, 5$ se u trećoj vrsti (redu) tabele, određuje na sledeći način:

$$x_{31} = x_{32} = x_{33} = 0;$$
$$x_{34} = \min\{29, 21\text{-}7\} = \min\{29,14\} = 14;$$
$$x_{35} = \min\{29\text{-}14, 27\} = \min\{15,27\} = 15;$$

Vrednost za x_{34} i x_{35} se upisuje u donjem desnom uglu polja (3,4; 3,5), kao što je prikazano u tabelama 2.12.9. i 2.12.10.

Tabele 2.12.9. – 2.12.10. *Dijagonalni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12** **23**	**1**	**4**	**13**
A_2	**23**	**7**	**8** **1**	**14** **15**	**6** **7**	**5**
A_3	**29**	**15**	**4**	**2**	**7** **14**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12** **23**	**1**	**4**	**13**
A_2	**23**	**7**	**8** **1**	**14** **15**	**6** **7**	**5**
A_3	**29**	**15**	**4**	**2**	**7** **14**	**9** **15**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Treća prijemna stanica je zadovoljena, a promenljive x_{4j}, j = 1, 2, 3, 4, 5 se u četvrtoj vrsti (redu) tabele, određuje na sledeći način:

$$x_{41} = x_{42} = x_{43} = x_{44} = 0;$$
$$x_{45} = \min\{12, 27\text{-}15\} = \min\{12, 12\} = 12.$$

Vrednost za x_{45} se upisuje u donjem desnom uglu polja (4,5), kao što je prikazano u tabeli 2.12.11.

Tabela 2.12.11. *Dijagonalni metod – određivanje početnog rešenja*

OS \ PS	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	5 (13)	12 (23)	1	4	13
A_2	**23**	7	8 (1)	14 (15)	6 (7)	5
A_3	**29**	15	4	2	7 (14)	9 (15)
A_4	**12**	6	11	5	16	3 (12)

Nenulte bazične promenljive, bazna polja, koncentrišu se oko glavne dijagonale u transportnoj tabeli, pa je otuda i potekao naziv “dijagonalni metod”, ali isto tako ovaj pravac podseća na pravac severozapad – jugoistok, pa se nekada naziva i “metod severo-zapadnog ugla”. Karakteristično je da popunjavanje tabele bazičnim promenljivim uvek počinje od polja (1,1), a završava se u polju (m,n). Da bi se istakla polja tabele u kojima su na ovaj način dobijene bazične promenljive, početno rešenje, zaokruženim brojevima je označena količina resursa koja treba da se transportuje iz određenih otpremnih centara do određenih prijemnih centara. Kao što se vidi, postoje $m+n-1 = 4+5-1 = 8$ baznih polja, tj. 8 pozitivnih promenljivih, dok su ostale promenljive jednake nuli. Prema tome, ukupni troškovi prevoza iznose:

$$F(x_0) = 5\cdot13 + 12\cdot23 + 8\cdot1 + 14\cdot15 + 6\cdot7 + 7\cdot14 + 9\cdot15 + 3\cdot12 = 870 \text{ n.j.}$$

2.12.2.2. Metod minimalnih cena u redovima

Kod ovog metoda, počinje se od prvog reda gde se uočava minimalna cena i u tom polju se postavlja maksimalna bazična vrednost, što se nastavlja i u ostalim redovima. U primeru koji je dat u tabeli 2.12.12., vidi se da je cena $c_{13} = 1$ najmanja, pa se za promenljivu x_{13} uzima vrednost:

$$x_{13} = \min\{23,15\} = 15.$$

Ovu vrednost za x_{13} se upisuje u donjem desnom uglu polja (1,3) tabele 2.12.12. Na ovaj način, treća prijemna stanica B_3 je zadovoljena, a prva otpremna stanica A_1 ima na raspolaganju još 36 – 15 = 21 jedinicu robe. Treća kolona se isključuje iz daljeg razmatranja, jer su potrebe prijemne stanice B_3 zadovoljene, a postupak se nastavlja sve dok otpremna stanica A_1 ne bude, takođe, zadovoljena.

Tabele 2.12.12. – 2.12.13. *Metod minimalnih cena u redovima – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4	13
A_2	23	7	8	14	6	5
A_3	29	15	4	2	7	9
A_4	12	6	11	5	16	3

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4 21	13
A_2	23	7	8	14	6	5
A_3	29	15	4	2	7	9
A_4	12	6	11	5	16	3

Nakon upisivanja vrednosti $x_{13} = 15$, polje sa najmanjim troškovima u prvom redu, koje je preostalo, je polje x_{14}, za koje je:

$$c_{14} = 4 = \min c_{ij},\ \ 1 \leq j \leq 5;\ \ j \neq 3;\ \ i = 1$$

u njega se upisuje preostala količina robe otpremne stanice A_1, kao što je prikazano u tabeli 2.12.13.:

$$x_{14} = \min\{36\text{-}15, 21\} = \min\{21, 21\} = 21.$$

Ovim je zadovoljena prva otpremna stanica A_1 i četvrta prijemna stanica B_4, pa se prvi red i druga kolona isključuje iz daljeg postupka.

Dalji postupak se nastavlja tako što se najmanja cena transporta po jedinici proizvoda traži u poljima drugog reda. Kako je:

$$c_{25} = 5 = \min c_{ij},\ \ 1 \leq j \leq 5,\ \ \ j \neq 3, 4\ \ \ i = 2$$

to se u donjem desnom uglu polja (2,5) upisuje vrednost bazične promenljive x_{25} = 23, kao što je prikazano u tabeli 2.12.14.

$$\min\{23, 27\} = 23.$$

Ovim je zadovoljena druga otpremna stanica, pa se drugi red isključuje iz daljeg razmatranja. Prijemna stanica B_5 raspolaže još sa kapacitetom od 27 – 23 = 4 jedinica robe.

Među cenama u trećem redu najmanju vrednost ima:

$$c_{32} = 4 = \min c_{ij},\ \ 1 \leq j \leq 5;\ \ j \neq 3, 4;\ \ \ i = 3$$

zato se u donjem desnom uglu polja (3,2) upisuje vrednost bazične promenljive x_{32}= 24, kao što je prikazano u tabeli 2.12.15.:

$$x_{32} = \min\{29, 24\} = 24.$$

Ovim je zadovoljena druga prijemna stanica, pa se druga kolona isključuje iz daljeg razmatranja. Otpremna stanica A_3 raspolaže još sa 29 – 24 = 5 jedinica robe.

Tabele 2.12.14. – 2.12.15. *Metod minimalnih cena u redovima – određivanje početnog rešenja*

OS \ *PS*		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4 21	13
A_2	23	7	8	14	6	5 23
A_3	29	15	4	2	7	9
A_4	12	6	11	5	16	3

OS \ *PS*		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4 21	13
A_2	23	7	8	14	6	5 23
A_3	29	15	4 24	2	7	9
A_4	12	6	11	5	16	3

Nakon upisivanja vrednosti $x_{32} = 24$, polje sa najmanjim troškovima u trećem redu, koje je preostalo, je polje x_{35} , za koje je:

$$c_{35} = 9 = \min c_{ij},\ \ 1 \leq j \leq 5;\ \ j \neq 2,3,4;\ \ i = 3$$

U polje x_{35} se upisuje maksimalna moguća količina preostale robe otpremne stanice A_3, a to je:

$$x_{35} = \min\{29\text{-}24, 27\text{-}23\} = \min\{5,4\} = 4,$$

kao što je prikazano u tabeli 2.12.16.

Tabele 2.12.16. – 2.12.17. *Metod minimalnih cena u redovima – određivanje početnog rešenja*

OS \ *PS*		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4 21	13
A_2	23	7	8	14	6	5 23
A_3	29	15	4 24	2	7	9 4
A_4	12	6	11	5	16	3

OS \ *PS*		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5	12	1 15	4 21	13
A_2	23	7	8	14	6	5 23
A_3	29	15 1	4 24	2	7	9 4
A_4	12	6	11	5	16	3

Na ovaj način su zadovoljene potrebe potrošačkog centra B_5, tako da se peta kolona izostavlja iz daljeg postupka. Međutim, u trećoj otpremnoj stanici A_3 ima na raspolaganju još 29 – 24 – 4 = 1 jedinica robe. Ona se može rasporediti u polje sa najmanjim troškovima koje je preostalo u trećem redu, a to je:

$$c_{31} = 15 = \min c_{ij},\ \ 1 \leq j \leq 5;\ \ j \neq 2,3,4,5;\ \ i = 3$$

tako da se u donjem desnom uglu polja (3,1) tabele 2.12.17. upisuje maksimalna moguća količina preostale robe otpremne stanice A_3, a to je:

$$\min\{13, 29\text{-}24\text{-}4\} = \min\{13,1\} = 1.$$

Na ovaj način je ponuda otpremne stanice A_3 zadovoljena, pa se iz daljeg postupka proračuna izostavlja treći red.

Postupak sa nastavlja popunjavanjem četvrtog reda. Jedino polje u četvrtom redu preko kog može da se izvrši transport je:

$$c_{41} = 6 = \min c_{ij},\ 1 \leq j \leq 5;\ j \neq 2,3,4,5;\ i = 4$$

Otuda sledi da se u donjem desnom uglu polja (1,4) upisuje vrednost bazične promenljive x_{41}=12.

$$x_{41} = \min\{12,13\text{-}1\} = \min\{12,12\} = 12.$$

kao što je prikazano u tabeli 2.12.18.

Tabela 2.12.18. *Metod minimalnih cena u redovima – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	**13**	**24**	**15**	**21**	**27**
A_1	**36**	5	12	1 (15)	4 (21)	13
A_2	**23**	7	8	14	6	5 (23)
A_3	**29**	15 (1)	4 (24)	2	7	9 (4)
A_4	**12**	6 (12)	11	5	16	3

Brojevima koji su zaokruženi je označena količina resursa koja treba da se transportuje iz određenih otpremnih centara do određene prijemne centre. Kao što može da se vidi, u rešenju se nalaze 7 baznih polja (zaokruženi brojevi), dok su ostale promenljive jednake nuli, i nije ispunjen uslov nedegenerisanosti:

$$r = m + n - 1 = 4 + 5 - 1 = 8.$$

Ovde je reč o *degenerisanom* početnom rešenju transportnog problema. Način na koji se rešavaju degenerisani transportni zadaci prikazaće se u posebnom poglavlju.

Prema tome, ukupni troškovi prevoza, u ovom slučaju, iznose:

$$F(x_0) = 1\cdot15 + 4\cdot21 + 5\cdot23 + 15\cdot1 + 4\cdot24 + 9\cdot4 + 6\cdot12 = 433 \text{ n.j.}$$

Nije teško zapaziti da su ukupni transportni troškovi za ovo početno rešenje znatno manji od troškova za rešenje dobijeno pomoću dijagonalnog metoda.

2.12.2.3. Metod minimalnih cena u kolonama

U analiziranoj prvoj koloni se postavlja maksimalni bazni elemenat u polje sa najmanjom jediničnom cenom transporta. Nadalje se postupak nastavlja za preostale kolone. Postupak je ekvivalentan predhodnom metodu minimalnih cena u redovima. Za primer koji je dat u tabeli 2.12.19., uočava se da je u prvoj koloni cena c_{11} =5 najmanja, pa se za promenljivu x_{11} uzima vrednost:

$$x_{11} = \min\{36,13\} = 13.$$

Ova vrednost za x_{11} se upisuje u donjem desnom uglu polja (1,1) tabele 2.12.19. Na ovaj način, prva prijemna stanica B_1 je zadovoljena, a prva otpremna stanica A_1 ima na raspolaganju još 36 – 13 = 23 jedinica robe. Zbog toga se prva kolona isključuje iz daljeg razmatranja.

Postupak se nastavlja tako što se najmanja cena transporta po jedinici proizvoda traži u poljima druge kolone. Pošto je:

$$c_{32} = 4 = \min c_{ij},\ 1 \leq i \leq 4,\ j = 2$$

onda se donjem desnom uglu polja (3,2) upisuje vrednost bazne promenljive x_{32} = 24, kao što je prikazano u tabeli 2.12.20.:

$$\min\{29, 24\} = 24.$$

Na ovaj način je zadovoljena druga prijemna stanica, pa se druga kolona isključuje iz daljeg razmatranja. Otpremna stanica A_3 raspolaže još sa 29 – 24 = 5 jedinica robe.

Tabele 2.12.19. – 2.12.20. *Metod minimalnih cena u kolonama – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 (13)	12	1	4	13
A_2	23	7	8	14	6	5
A_3	29	15	4	2	7	9
A_4	12	6	11	5	16	3

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 (13)	12	1	4	13
A_2	23	7	8	14	6	5
A_3	29	15	4 (24)	2	7	9
A_4	12	6	11	5	16	3

Među cenama u trećoj koloni najmanju vrednost ima:

$$c_{13} = 1 = \min c_{ij},\ 1 \leq i \leq 4,\ j = 3.$$

zato se u donjem desnom uglu polja (1,3) upisuje vrednost bazične promenljive x_{13}=15, kao što je prikazano u tabeli 2.12.21.:

$$x_{13} = \min\{36\text{-}13,15\} = \min\{23,15\} = 15.$$

Ovim je zadovoljena treća prijemna stanica, pa se treća kolona isključuje iz daljeg razmatranja. Otpremna stanica A_1 raspolaže još sa 36 – 13 – 15 = 8 jedinica robe.

Najmanja vrednost cene u četvrtoj koloni je:

$$c_{14} = 4 = \min c_{ij},\ \ 1 \leq i \leq 4,\ \ j = 4$$

Sledi da se u donjem desnom uglu polja (1,4) upisuje vrednost bazične promenljive $x_{14} = 8$, kao što je prikazano u tabeli 2.12.22.:

$$x_{14} = \min\{36\text{-}13\text{-}15,21\} = \min\{8,21\} = 8.$$

Tabele 2.12.21. – 2.12.22. *Metod minimalnih cena u kolonama – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** (13)	**12**	**1** (15)	**4**	**13**
A_2	**23**	**7**	**8**	**14**	**6**	**5**
A_3	**29**	**15**	**4** (24)	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

OS \ PS		B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** (13)	**12**	**1** (15)	**4** (8)	**13**
A_2	**23**	**7**	**8**	**14**	**6**	**5**
A_3	**29**	**15**	**4** (24)	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

Na ovaj način je zadovoljena prva otpremna stanica, pa se prvi red isključuje iz daljeg razmatranja. Prijemna stanica B_4 raspolaže kapacitetom prijema sa još: 21 – 8 =13 jedinica robe. Dalji postupak se nastavlja tako što se u četvrtoj koloni, u poljima koja su preostala, pronalazi polje sa sledećom najmanjom cenom prevoza:

$$c_{24} = 6 = \min c_{ij},\ \ 1 \leq i \leq 4;\ \ i \neq 1;\ \ j = 5$$

Zato se u donjem desnom uglu polja (2,4) upisuje vrednost bazične promenljive x_{24} =13, kao što je prikazano u tabeli 2.12.23.:

$$x_{24} = \min\{23,21\text{-}8\} = \min\{23,13\} = 13.$$

Tako je zadovoljena četvrta prijemna stanica, pa se četrvta kolona isključuje iz daljeg razmatranja. Otpremna stanica A_2 raspolaže kapacitetom još sa: 23 – 13 = 10 jedinica robe. U petoj koloni, najmanja vrednost cene je:

$$c_{45} = 3 = \min c_{ij},\ \ 1 \leq i \leq 4;\ \ i \neq 1;\ \ j = 5$$

Otuda sledi da se u donjem desnom uglu polja (4,5) upisuje vrednost bazične promenljive x_{45} = 12, kao što je prikazano u tabeli 2.12.24.:

$$x_{45} = \min\{12,27\} = 12.$$

Na ovakav način je zadovoljena četvrta otpremna stanica A_2, pa se četvrti red isključuje iz daljeg razmatranja. Prijemna stanica B_5 raspolaže još sa kapacitetom prijema od 27 – 12 = 15 jedinica robe.

Nepopunjena polja koja su preostala u petoj koloni x_{25} i x_{35} se popunjavaju vrednostima promenljivih na sledeći način:

$$x_{25} = \min\{23\text{-}13, 27\text{-}12\} = \min\{10, 15\} = 10$$
$$x_{35} = \min\{12, 27\text{-}12\text{-}10\} = \min\{12, 5\} = 5.$$

Popunjavanje polja x_{25} i x_{35} je prikazano u tabelama 2.12.25. i 2.12.26.

Tabele 2.12.23. – 2.12.24. *Metod minimalnih cena u kolonama – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12**	**1** **15**	**4** **8**	**13**
A_2	**23**	**7**	**8**	**14**	**6** **13**	**5**
A_3	**29**	**15**	**4** **24**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3**

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12**	**1** **15**	**4** **8**	**13**
A_2	**23**	**7**	**8**	**14**	**6** **13**	**5**
A_3	**29**	**15**	**4** **24**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3** **12**

Tabele 2.12.25. – 2.12.26. *Metod minimalnih cena u kolonama – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12**	**1** **15**	**4** **8**	**13**
A_2	**23**	**7**	**8**	**14**	**6** **13**	**5** **10**
A_3	**29**	**15**	**4** **24**	**2**	**7**	**9**
A_4	**12**	**6**	**11**	**5**	**16**	**3** **12**

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** **13**	**12**	**1** **15**	**4** **8**	**13**
A_2	**23**	**7**	**8**	**14**	**6** **13**	**5** **10**
A_3	**29**	**15**	**4** **24**	**2**	**7**	**9** **5**
A_4	**12**	**6**	**11**	**5**	**16**	**3** **12**

Zaokruženim brojevima je označena količina resursa koja treba da se transportuje iz određenih otpremnih do određene prijemne centre. Kao što može da se vidi, u rešenju se nalaze 8 baznih polja (zaokruženi brojevi), dok su ostale promenljive jednake nuli, i ispunjen je uslov nedegenerisanosti:

$$r = m + n - 1 = 4 + 5 - 1 = 8.$$

Ovde je pronađeno *nedegenerisano* početno rešenje transportnog problema, u kome funkcija cilja ima sledeću vrednost:

$$F(x_0) = 5\cdot13 + 1\cdot15 + 4\cdot8 + 6\cdot13 + 5\cdot10 + 4\cdot24 + 9\cdot5 + 3\cdot12 = 417 \text{ n.j.}$$

Može da se uoči da su ukupni transportni troškovi, za ovo početno rešenje, manji od troškova koji su dobijeni pomoću dijagonalnog metoda i metodom minimalnih cena u redovima.

2.12.2.4. Metod minimalnih cena u matrici

Prema metodu minimalnih cena u matrici, ili kako se drugačije zove metod najmanjih jediničnih koeficijenata (cena), za prvu baznu promenljivu se uzima najveće moguće prevoženje u polju tabele, gde je cena prevoza najmanja.

U primeru koji je dat u tabeli 2.12.27., vidi se da je cena c_{22} =1 najmanja, pa se za x_{22} uzima vrednost:

$$x_{22} = \min\{300,260\} = 260.$$

Ovu vrednost za x_{22} se upisuje u donjem desnom uglu polja (2,2) tabele 2.12.27. Na ovaj način, druga prijemna stanica B_2 je zadovoljena, a druga otpremna stanica A_2 ima na raspolaganju još 300 – 260 = 40 jedinica robe. Da se ne bi ispustilo iz vida da otpremna stanica A_2 raspolaže sa još 40 jedinica robe, upisaće se još neraspodeljena količina robe na desnoj strani od tabele u visini druge vrste.

Tabela 2.12.27. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	**4**	**4**	**5**	**6**	**7**	
A_2	**300**	**7**	**1** 260	**3**	**8**	**11**	**40**
A_3	**350**	**2**	**4**	**8**	**5**	**9**	
A_4	**150**	**10**	**7**	**9**	**3**	**10**	

Sada se najmanja od preostalih cena traži u poljima preostalih kolona: prve, treće, četvrte i pete. Kako je:

$$c_{31} = 2 = \min c_{ij},\ \ 1 \le i \le 4;\ \ 1 \le j \le 5;\ \ j \ne 2$$

to se u donjem desnom uglu polja (3,1) upisuje vrednost bazične promenljive x_{31}:

$$x_{31} = \min\{350, 140\} = 140.$$

Ovim je zadovoljena i prva prijemna stanica, pa se prva kolona isključuje iz daljeg razmatranja. Otpremna stanica A_3 raspolaže još sa 350 – 140 = 210 jedinica robe, što se zapisuje desno od tabele u produžetku treće vrste, prikazano u tabeli 2.12.28.

Između cena u trećoj, četvrtoj i petoj koloni najmanju vrednost imaju cene c_{23} i c_{44}:

$$c_{23} = c_{44} = 3 = \min c_{ij};\ \ 1 \le \mathrm{i} \le 4;\ \ \ 1 \le \mathrm{j} \le 5;\ \ \ j \ne 1, \mathrm{j} \ne 2$$

zato se određuju najpre vrednosti promenljive x_{23} i x_{44} i sasvim je svejedno, u ovoj situaciji, kojim redom se to čini.

$$x_{23} = \min\{300\text{-}260,\ 230\} = \min\{40,\ 230\} = 40$$

Ovim je iscrpljena druga otpremna stanica, pa se naznačena količina robe desno od tabele precrtava. Ali, trećoj prijemnoj stanici B_3 potrebno je dostaviti još 230 – 40 = 190 jedinica robe, pa se ta količina upisuje na kraju treće kolone ispod tabele, kao što je prikazano u tabeli 2.12.29.

Tabela 2.12.28. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	4	4	5	6	7	
A_2	**300**	7	1 260	3	8	11	**40**
A_3	**350**	2 140	4	8	5	9	**210**
A_4	**150**	10	7	9	3	10	

Tabela 2.12.29. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	4	4	5	6	7	
A_2	**300**	7	1 260	3 40	8	11	~~**40**~~
A_3	**350**	2 140	4	8	5	9	**210**
A_4	**150**	10	7	9	3	10	
				190			

Kada se odredi za:

$$x_{44} = \min\{150,\ 140\} = 140,$$

u četvrtoj otpremnoj stanici ostaje još 10 jedinica robe koja može da se otpremi, pa se to upisuje na kraju četvrtog reda pored tabele, a prijemna stanica B_4 je popunjena, što je prikazano u tabeli 2.12.30.

Među cenama u trećoj i petoj koloni najmanju vrednost ima cena $c_{13} = 5$, pa stavljamo da je

$$x_{13} = \min\{200,\ 230 - 40\} = 190.$$

Kako je ovim zadovoljena treća prijemna stanica precrtava se razlika (190), koju je upisana na dnu ove kolone u prethodnom koraku. Prva otpremna stanica raspolaže sa još 10 (200 – 190 = 10) jedinica robe, tako da se ta količina upisuje na desnoj strani od tabele u visini prve vrste, kao što je prikazano u tabeli 2.12.31.

Tabela 2.12.30. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	**4**	**4**	**5**	**6**	**7**	
A_2	**300**	**7**	**1** **260**	**3** **40**	**8**	**11**	~~**40**~~
A_3	**350**	**2** **140**	**4**	**8**	**5**	**9**	**210**
A_4	**150**	**10**	**7**	**9**	**3** **140**	**10**	**10**
				190			

Tabela 2.12.31. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	**4**	**4**	**5** **190**	**6**	**7**	**10**
A_2	**300**	**7**	**1** **260**	**3** **40**	**8**	**11**	~~**40**~~
A_3	**350**	**2** **140**	**4**	**8**	**5**	**9**	**210**
A_4	**150**	**10**	**7**	**9**	**3** **140**	**10**	**10**
				~~**190**~~			

Preostaju još polja pete kolone i ona se popunjavaju vrednostima promenljivih x_{15}, x_{35}, i x_{45}, na sledeći način:

$$x_{15} = 10;$$
$$x_{35} = \min\{350\text{-}140,\ 230\text{-}10\} = \min\{210,\ 220\} = 210;$$
$$x_{45} = \min\{150\text{-}140,\ 230\text{-}10\text{-}210\} = \min\{10,10\} = 10;$$

što se može videti i tabelama od 2.12.32. do 2.12.34.

Tabela 2.12.32. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
A_1	**200**	**4**	**4**	**5** **190**	**6**	**7** **10**	~~**10**~~
A_2	**300**	**7**	**1** **260**	**3** **40**	**8**	**11**	~~**40**~~
A_3	**350**	**2** **140**	**4**	**8**	**5**	**9**	**210**
A_4	**150**	**10**	**7**	**9**	**3** **140**	**10**	**10**
				~~**190**~~			

Tabela 2.12.33. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
***A*₁**	**200**	4	4	5 190	6	7 10	~~10~~
***A*₂**	**300**	7	1 260	3 40	8	11	~~40~~
***A*₃**	**350**	2 140	4	8	5	9 210	~~210~~
***A*₄**	**150**	10	7	9	3 140	10	10
				~~190~~			

Tabela 2.12.34. *Metod minimalnih cena u matrici – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	
OS	**1000**	**140**	**260**	**230**	**140**	**230**	
***A*₁**	**200**	4	4	5 190	6	7 10	~~10~~
***A*₂**	**300**	7	1 260	3 40	8	11	~~40~~
***A*₃**	**350**	2 140	4	8	5	9 210	~~210~~
***A*₄**	**150**	10	7	9	3 140	10 10	~~10~~
				~~190~~			

Ovakvom preraspodelom transpotra zadovoljene su sve otpremne i sve prijemne stanice. Dobijeno rešenje je bazično, *nedegenerisano*, jer ima $m+n-1$ = 4+5 – 1 = 8 pozitivnih promenljivih. Ukupni troškovi prevoza su:

$$F(x_0)= 5\cdot190 + 7\cdot10 + 1\cdot260 + 3\cdot40 + 2\cdot140 + 9\cdot210 + 3\cdot140 + 10\cdot10 = 3.990 \text{ n.j.}$$

2.12.2.5. Vogelov aproksimativni metod

Vogelov aproksimativni metod je vrlo efikasan metod pronalaženja početnog rešenja transportnog problema. Rešenje dobijeno ovim metodom ne samo da je blizu optimalnom, već će u nekim slučajevima dati odmah optimalno rešenje.

Postupak za rešavanje je sledeći:

a) Za svaki red i svaku kolonu matrice troškova izračunavaju se razlike između dva najmanja koeficijenta c_{ij} (najmanjeg i neposredno većeg troška). Ta razlika se može tretirati kao “kazna” za nekorišćenje najjeftinije transportne relacije. Ako su u nekoj liniji matrice (red, kolona) dva najmanja troška jednaka, onda je razlika za tu liniju jednaka nuli.

b) U narednom koraku izabere se red ili kolona sa najvećom “kaznom” i on će dobiti prednost pri određivanju vrednosti promenljivih. Kada jedan red, odnosno kolona, dobije prednost nad ostalim, u njemu se u polje s najnižim transportnim troškovima određuje promenljiva sa najvećom mogućom vrednošću. Vrednost promenljive x_{ij} jednaka je ili ponudi a_i ili potražnji b_j, već prema manjoj vrednosti. Ukoliko je $x_{ij}=a_i$ onda je ishodiste a_i (kapacitet nekog proizvođača) potpuno iscrpljeno, pa se taj red u narednoj iteraciji izostavlja. U protivnom je zadovoljeno odrediste b_j (potreba nekog potrošača), pa se u narednoj iteraciji izostavlja ta kolona.

c) Ponovo se preračunavaju razlike za preostale redove i kolone. Ako je izostavljen red, nove "kazne" se računaju za kolone jer su po redovima ostale iste i, obratno, ako je izostavljena kolona, računaju se "kazne" za redove.

d) Postupak računanja "kazni", davanje prednosti redu ili koloni, biranje polja sa najmanjim troškom u izabranom redu (koloni) i određivanje vrednosti nove promenljive ponavlja se dok se ne dobije bazično rešenje problema. Drugim rečima, koraci b) i c) se ponavljaju toliko dugo sve dok se ne rasporede količine a_i na odredišta b_j, čime se dobijaju sve komponente početnog baznog rešenja.

Vrednosti aproksimativnog rešenja (početnog rešenja) po Vogelovom metodu dobijaju se množenjem vrednosti dobijenih količina transporta (promenljivih) s odgovarajućim troškovima transporta i njihovim sabiranjem.

Primer 2.12.1. Potrebno je transportovati robu iz četiri ishodišta (otpremnih stanica) u pet odredišta (prijemnih stanica). Količine raspoložive robe u ishodištima i tražene robe u odredištima, kao i transportni troškovi dati su u tabeli 2.12.35. Kriterijum za izvršenje transporta je minimum ukupnih transportnih troškova.

Rešenje. U tabeli 2.12.35. izračunate su razlike između dva najmanja koeficijenta svakog reda i svake kolone i upisane izvan tabele. Posmatrajući razlike, utvrđuje se da su one najveće i jednake za treću i četvrtu kolonu. Zbog toga potrošači B_3 i B_4 imaju prednost u podmirenju svojih potreba. Najmanji trošak u trećoj i četvrtoj koloni je $c_{23} = c_{34} = 3$. pa treba odrediti vrednost ili promenljivoj x_{23} ili x_{34}. Prednost je data promenljivoj koja dobija veću vrednost, a to je $x_{23} = 30$.

Tabela 2.12.35. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5	
	200	**60**	**50**	**30**	**20**	**40**	
A_1	**65**	8	3	5	5	4	**1**
A_2	**50**	2	5	3 (30)	6	8	**1**
A_3	**40**	4	2	8	3	6	**1**
A_4	**45**	3	6	9	5	3	**0**
		1	**1**	**2**	**2**	**1**	

Pošto je potrošač B_3 podmirio svoje potrebe, treća kolona u tabeli izostavlja se iz daljeg razmatranja. Zbog toga se ponovo računaju razlike za redove (po kolonama ostaju iste). Nove razlike su ispisane, takođe, pored tabele, kao što je prikazano u tabeli 2.12.36. Sada je najveća razlika 3 za drugi red. U drugom redu najmanji koeficijent je $c_{21}=2$. Prema tome, u bazično rešenje se stavlja $x_{21}=20$. Ovim je iscrpljena ponuda drugog proizvođača, pa se drugi red izostavlja iz daljeg razmatranja.

Tabela 2.12.36. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
***A*₁**	**65**	**8**	**3**	**5**	**5**	**4**	**1**
***A*₂**	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**3**
***A*₃**	**40**	**4**	**2**	**8**	**3**	**6**	**1**
***A*₄**	**45**	**3**	**6**	**9**	**5**	**3**	**0**
		1	**1**	**-**	**2**	**1**	

Potrebno je izračunati nove razlike po kolonama. Ovoga puta, one su ostale nepromenjene. U tabeli 2.12.37. je za kolonu B_4 kojoj odgovara najveća razlika (2), izabran najmanji trošak c_{34} i određena je nova bazična promenljiva x_{34}=20. Ovim su podmirene potrebe i potrošača B_4, pa se iz daljeg razmatranja izostavlja i četvrta kolona (pored već izostavljenog drugog reda i treće kolone).

Tabela 2.12.37. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
***A*₁**	**65**	**8**	**3**	**5**	**5**	**4**	**1**
***A*₂**	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**-**
***A*₃**	**40**	**4**	**2**	**8**	**3** 20	**6**	**1**
***A*₄**	**45**	**3**	**6**	**9**	**5**	**3**	**0**
		1	**1**	**-**	**2**	**1**	

Računaje se nove razlike za redove i ispisuju se pored tabele. Najveća razlika (2) odgovara trećem redu. U trećem redu najmanji koeficijent je c_{32}=2, pa je nova bazična promenljiva x_{32}=20, kao što je prikazano u tabeli 2.12.38. Time je iscrpljena ponuda i trećeg proizvođača, pa se iz daljeg razmatranja izostavlja i treći red tabele.

Tabela 2.12.38. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
***A*₁**	**65**	**8**	**3**	**5**	**5**	**4**	**1**
***A*₂**	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**-**
***A*₃**	**40**	**4**	**2** 20	**8**	**3** 20	**6**	**2**
***A*₄**	**45**	**3**	**6**	**9**	**5**	**3**	**0**
		1	**1**	**-**	**-**	**1**	

Dalje, za kolone se izračunavaju nove razlike i ispisuju se pored tabele. Najveća razlika (5) odgovara prvoj koloni. U ovoj koloni, od preostalih koeficijenata, najmanji je c_{41}=3, pa je nova bazična promenljiva x_{41}=40, kao što je prikazano u tabeli 2.12.39. Ovoga puta podmirene su potrebe potrošača B_1, pa iz daljeg razmatranja se izostavlja i prva kolona.

Tabela 2.12.39. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
A_1	**65**	**8**	**3**	**5**	**5**	**4**	**1**
A_2	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**-**
A_3	**40**	**4**	**2** 20	**8**	**3** 20	**6**	**-**
A_4	**45**	**3** 40	**6**	**9**	**5**	**3**	**0**
		5	**1**	**-**	**-**	**1**	

Računaju se nove razlike. Najveća razlika (3) odgovara četvrtom redu i drugoj koloni. Najmanji koeficijent u četvrtom redu i drugoj koloni je c_{12}=c_{45}=3, pa treba odrediti vrednost jednoj od promenljivih x_{12} ili x_{45}. Prednost je data promenljivoj x_{12}=30 jer dobija veću vrednost, kao što je prikazano u tabeli 2.12.40. Ovim su podmirene potrebe i potrošača B_2.

Tabela 2.12.40. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
A_1	**65**	**8**	**3** 30	**5**	**5**	**4**	**1**
A_2	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**-**
A_3	**40**	**4**	**2** 20	**8**	**3** 20	**6**	**-**
A_4	**45**	**3** 40	**6**	**9**	**5**	**3**	**3**
		-	**3**	**-**	**-**	**1**	

Posle izostavljanja druge kolone, u tabeli ostaje samo još peta kolona. Zbog toga nije potrebno računati nove razlike, vrednosti preostalih promenljivih za potrošača B_5 lako se određuju. Tabela 2.12.41. sadrži i te promenljive: x_{15} = 35 i x_{45} = 5.

Tabela 2.12.41. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	
OS	**200**	**60**	**50**	**30**	**20**	**40**	
A_1	**65**	**8**	**3** 30	**5**	**5**	**4** 35	**1**
A_2	**50**	**2** 20	**5**	**3** 30	**6**	**8**	**-**
A_3	**40**	**4**	**2** 20	**8**	**3** 20	**6**	**-**
A_4	**45**	**3** 40	**6**	**9**	**5**	**3** 5	**3**
		-	**-**	**-**	**-**	**1**	

Na ovaj način je određeno početno bazično rešenje po Vogelovom metodu. Njega sačinjavaju sledeće promenljive: $x_{12} = 30$, $x_{15} = 35$, $x_{21} = 20$, $x_{23} = 20$, $x_{32} = 20$, $x_{34} = 20$, $x_{41} = 40$, $x_{45} = 5$.

Vrednost funkcije cilja iznosi:

$$F(x_0)=3\cdot 30+4\cdot 35+2\cdot 20+3\cdot 30+2\cdot 20+3\cdot 20+3\cdot 40+3\cdot 5 = 595 \text{ n.j.}$$

U tabeli 2.12.42. prikazano je rešavanje ovog primera, iterativnim postupkom, tako što se ispisivanje "kazni", za nekorišćenje najjeftinije transportne relacije, odvija izvan tabele u posebnom redu i posebnoj koloni.

Tabela 2.12.42. *Vogel-ov aproksimativni metod – određivanje početnog rešenja*

	PS	B_1	B_2	B_3	B_4	B_5	***Iteracije***						
OS	**200**	**60**	**50**	**30**	**20**	**40**	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**
A_1	**65**	8	3 30	5	5	4 35	1	1	1	1	1	1	1
A_2	**50**	2 20	5	3 30	6	8	1	3	-	-	-	-	-
A_3	**40**	4	2 20	8	3 20	6	1	1	1	2	-	-	-
A_4	**45**	3 40	6	9	5	3 5	0	0	0	0	0	3	3
Iteracije	**I**	1	1	2	2	1							
	II	1	1	-	2	1							
	III	1	1	-	2	1							
	IV	1	1	-	-	1							
	V	5	1	-	-	1							
	VI	-	3	-	-	1							
	VII	-	-	-	-	1							

2.12.2.6. *Vogel – Kordin postupak*

Matematičar Korda je modifikovao Vogel-ov metod uvođenjem prethodne redukci-je matrice troškova u dva koraka:

a) Od svakog elementa reda prvobitne matrice troškova oduzme se najmanji element tog reda. To se uradi za sve redove matrice, što daje najmanje po jednu nulu u svakom redu.
b) Od svakog elementa kolone prethodno redukovane matrice (prema prvom koraku redukcije) oduzme se najmanji erlement date kolone. To se učini za sve kolone, čime se povećava broj nula u matrici.

Dalji postupak se vrši prema Vogel-ovom metodu, ali na temelju dobijene redukovane matrice troškova.

Primer 2.12.2. Vogel-Kordin postupak biće pokazan na primeru gde je potrebno transportovati robu iz četiri otpremnih u pet prijemnih stanica. Svi potrebni podaci dati su u tabeli 2.12.43. Kriterijum za izvršenje transporta je minimum ukupnih transportnih troškova.

Tabela 2.12.43. *Početni podaci za primer urađen Vogel-Kordinim postupkom*

OS \ PS	PS	B_1	B_2	B_3	B_4	B_5
OS	**117**	**17**	**21**	**41**	**14**	**24**
A_1	**25**	**10**	**8**	**9**	**6**	**5**
A_2	**32**	**5**	**6**	**4**	**3**	**8**
A_3	**40**	**9**	**7**	**5**	**4**	**3**
A_4	**20**	**14**	**10**	**8**	**8**	**8**

Rešenje. U tabeli 2.12.44. prikazani su transportni troškovi koji su potrebni za izračunavanje početnog rešenje na osnovu Vogel-Kordinog postupka. Redukovana matrica po redovima prikazana je u tabeli 2.12.45., a redukovana matrica po kolonama u tabeli 2.12.46.

Tabela 2.12.44. *Početna tabela za Vogel-Kordin postupak*

a_i \ b_j	b_1	b_2	b_3	b_4	b_5	*min* c_{ij}
a_1	10	8	9	6	5	**5**
a_2	5	6	4	3	8	**3**
a_3	9	7	5	4	3	**3**
a_4	14	10	8	8	8	**8**

Tabela 2.12.45. *Redukovana tabela po redovima*

a_i \ b_j	b_1	b_2	b_3	b_4	b_5
a_1	5	3	4	1	0
a_2	2	3	1	0	5
a_3	6	4	2	1	0
a_4	6	2	0	0	0
min c_{ij}	2	2	0	0	0

Tabela 2.12.46. *Redukovana tabela po kolonama*

a_i \ b_j	b_1	b_2	b_3	b_4	b_5
a_1	3	1	4	1	0
a_2	0	1	1	0	5
a_3	4	2	2	1	0
a_4	4	0	0	0	0

Nakon obavljene dvostruke redukcije na dobijenu matricu se primenjuje poznata Vogelov aproksimativni metod. Početno rešenje po Vogel-Kordinom postupku prikazano je u tabeli 2.12.47.

Funkcija cilja (kriterijuma) se izračunava tako što se vrednosti za dobijene promenljive unesu u tabelu sa originalnim transportnim troškovima izmnože a nakon toga saberu, kao što je prikazano u tabeli 2.12.48.

Tabela 2.12.47. *Početno rešenje po Vogel-Kordinom postupku–redukovana tabela*

OS \ PS		B_1	B_2	B_3	B_4	B_5	Iteracije I II III IV V VI VII
	117	17	21	41	14	24	
A_1	25	3	1 (21)	4	1 (4)	0	1 1 0 0 0 0 1
A_2	32	0 (17)	1	1 (15)	0	5	0 1 1 1 - - -
A_3	40	4	2	2 (6)	1 (10)	0 (24)	1 1 1 1 1 1 1
A_4	20	4	0	0 (20)	0	0	0 0 0 - - - -
Iteracije	I	3	1	1	0	0	
	II	-	1	1	0	0	
	III	-	1	1	0	-	
	IV	-	0	1	1	-	
	V	-	1	2	0	-	
	VI	-	1	-	0	-	
	VII	-	-	-	0	-	

Tabela 2.12.48. *Početno rešenje po Vogel-Kordinom postupku*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	117	17	21	41	14	24
A_1	25	10	8 (21)	9	6 (4)	5
A_2	32	5 (17)	6	4 (15)	3	8
A_3	40	9	7	5 (6)	4 (10)	3 (24)
A_4	20	14	10	8 (20)	8	8

Transportni troškovi iznose:

$$F(x_0) = 8.21 + 6.4 + 5.17 + 4.15 + 5.6 + 4.10 + 3.24 + 8.20 = 639 \text{ n.j.}$$

Po pravilu Vogel-Kordin postupak daje najbolja početna rešenja transportnog zadatka, od svih metoda koje su do sada obrađene. U većim, složenijim, primerima ta razlika bi bila još izraženija, u korist ovog postupka.

2.12.2.7. Metod dvojnog prvenstva – dvostrukog precrtavanja

U svakoj koloni matrice troškova sa * (zvezdicom) se označi polje sa minimalnim troškovima. Takođe, u svakom redu sa * se označi polje sa minimalnim troškovima. Nakon završenog označavanja moguće su sledeće situacije: neka polja su označena sa dve, neka sa jednom zvezdicom, a neka su neoznačena. Prvo se popunjavaju polja označena sa dve zvezdice sa maksimalno mogućim količinama robe. Iz daljeg razmatranja se izostavljaju kolone ili redovi ili kolone i redovi, zavisno da li su zadovoljena odredišta ili su prazna ishodišta ili su ispunjena oba uslova. Nakon toga se popunjavaju polja označena sa jednom zvezdicom uz izostavljanje iz daljeg razmatranja kolona ili redova. Neoznačena polja se popunjavaju po redosledu minimalnih troškova. Postupak određivanja početnog rešenja po metodu dvojnog prvenstva (dvostrukog precrtavanja) biće prikazan na primeru.

Primer 2.12.3. Potrebno je transportovati robu iz četiri ishodišta (otpremnih stanica) u četiri odredišta (prijemnih stanica). Količine raspoložive robe u ishodištima i tražene robe u odredištima, kao i transportni troškovi dati su u tabeli 2.12.49. Kriterijum za izvršenje transporta je minimum ukupnih transportnih troškova.

Rešenje. U svakoj vrsti tabele se određuje polje sa najmanjom cenom i označava se zvezdicom (*). Zatim se to isto učini i sa kolonama. U nekim poljima naći će se po dve zvezdice, što znači da se u tim poljima nalaze najmanje cene u svojoj vrsti i koloni istovremeno, kao što je prikazano u tabeli 2.12.49.

Tabela 2.12.49. *Početni podaci za primer urađen metodom dvojnog prvenstva*

	PS	***B₁***	***B₂***	***B₃***	***B₄***
OS	**170**	**30**	**35**	**60**	**45**
A₁	**20**	**8**	**10**	**4 ****	**5**
A₂	**50**	**6**	**4 ***	**7**	**3 ****
A₃	**25**	**5 ****	**8**	**9**	**6**
A₄	**75**	**11**	**9**	**10**	**8 ***

U poljima sa dve zvezdice se stavljaju najveće moguće dostave. U ovom primeru to su polja (1,3), (2,4) i (3,1), pa se veličine dostava u tim poljima biraju na sledeći način:

$$x_{13} = \min\{20, 60\} = 20;$$
$$x_{24} = \min\{50, 45\} = 45;$$
$$x_{31} = \min\{25, 30\} = 25.$$

Preraspodela transporta u ovim poljima prikazana je u tabeli 2.12.50. Posle ovog upisivanja dostave, iz daljeg razmatranja se isključuju prva i treća vrsta i četvrta kolona. Na ovaj način, prva A_1 i teća A_3 otpremna stanica su zadovoljene, tako da su kapaciteti prijemnih stanica B_1 još 30-25=5 i B3 još 60-20=40 jedinica robe. Takođe, četvrta prijemna stanica B_4 je zadovoljena, tako da je ponuda otpremne stanice A_2 još 50-45=5 jedinica robe. Da se ne bi ispustili iz vida ovi podaci upisaće se u pomoćnom redu i pomoćnoj koloni pored tabele. Sada se postupak preraspodele transporta (dostava) nastavlja popunjavanjem polja označena sa jednom zvezdicom. Jedino preostalo polje sa jednom zvezdicom, koje se može popunjavati, je polje x_{22}, što je prikazano u tabeli 2.12.51. Kako je:

$$c_{22} = 4 = \min c_{ij}, \quad i \neq 1, i \neq 3, \ j \neq 4,$$

upisuje se u polje:

$$x_{22} = \min\{50-45, 35\} = \min\{5, 35\} = 5$$

Tabela 2.12.50. *Metod dvojnog prvenstva – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	
OS	**170**	**30**	**35**	**60**	**45**	
***A*₁**	**20**	**8**	**10**	**4 **** 20	**5**	
***A*₂**	**50**	**6**	**4 ***	**7**	**3 **** 45	**5**
***A*₃**	**25**	**5 **** 25	**8**	**9**	**6**	
***A*₄**	**75**	**11**	**9**	**10**	**8 ***	
		5		**40**		

Tabela 2.12.51. *Metod dvojnog prvenstva – određivanje početnog rešenja*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	
OS	**170**	**30**	**35**	**60**	**45**	
***A*₁**	**20**	**8**	**10**	**4 **** 20	**5**	
***A*₂**	**50**	**6**	**4 *** 5	**7**	**3 **** 45	~~**5**~~
***A*₃**	**25**	**5 **** 25	**8**	**9**	**6**	
***A*₄**	**75**	**11**	**9**	**10**	**8 ***	
		5	**30**	**40**		

Ovim je iscrpljena druga otpremna stanica A_2, pa se naznačena količina robe desno od tabele precrtava, a u drugoj koloni se upisuje preostali kapacitet prijemne stanice B_2, koja iznosi još 35-5=30. Preostaje još da se raspodeli roba iz četvrte otpremne stanice. Preraspodela robe četvrte otpremne stanice A_4 obaviće se na sledeći način:

$$x_{42} = \min\{75-5,\ 35-5\} = 30;$$
$$x_{43} = \min\{75-5-30,\ 60-20\} = \min\{40,40\} = 40;$$
$$x_{41} = \min\{75,\ 30-25\} = 5.$$

Kompletno urađeno početno rešenje za transportni zadatak, koristeći metodu dvojnog prvenstva, prikazano je u tabeli 2.12.52.

Tabela 2.12.52. *Metod dvojnog prvenstva – konačno početno rešenje*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	
OS	**170**	**30**	**35**	**60**	**45**	
***A*₁**	**20**	**8**	**10**	**4 **** 20	**5**	
***A*₂**	**50**	**6**	**4 *** 5	**7**	**3 **** 45	~~**5**~~
***A*₃**	**25**	**5 **** 25	**8**	**9**	**6**	
***A*₄**	**75**	**11** 5	**9** 30	**10** 40	**8 ***	~~**45**~~, ~~**5**~~
		~~**5**~~	~~**30**~~	~~**40**~~		

Ukupni troškovi prema dobijenom baznom planu prevoženja jednaki su:

$$F(x_0) = 4 \cdot 20 + 4 \cdot 5 + 3 \cdot 45 + 5 \cdot 25 + 11 \cdot 5 + 9 \cdot 30 + 10 \cdot 40 = 1085 \text{ n.j.}$$

2.12.3. Određivanje optimalnog rešenja

Sve metode rešavanja transportnog zadatka proveravaju najpre da li je početno bazično rešenje optimalno ili nije. Ukoliko početno bazično rešenje nije optimalno, svaki od ovih metoda pokazuje kako se prelazi na bolje bazično rešenje, tj. na bazično rešenje koje obezbeđuje smanjenje troškova prevoza.

2.12.3.1. Metod raspodele

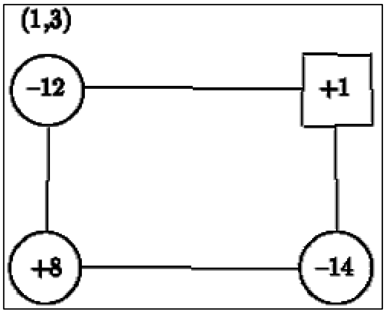
Ovaj metod je nastao u Americi krajem 40-tih godina i sa svojim modifikacijama jedan je od najjednostavnijih metoda za ručno rešavanje transportnog zadatka.

Ovaj metod biće prikazan na primeru u kome je dijagonalnim metodom određeno početno bazično rešenje, tabela 2.12.53. To bazično rešenje je nedegenerisano i ima $m + n - 1 = 4 + 5 - 1 = 8$ pozitivnih bazičnih promenljivih. Ukupni troškovi prevoza po ovom prvom bazičnom rešenju jednaki su $F_1 = 870$ n.j.

Tabela 2.12.53. *Početno rešenje TP određeno dijagonalnim metodom*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	**5** 13	**12** 23	**1**	**4**	**13**
A_2	**23**	**7**	**8** 1	**14** 15	**6** 7	**5**
A_3	**29**	**15**	**4**	**2**	**7** 14	**9** 15
A_4	**12**	**6**	**11**	**5**	**16**	**3** 12

Provera optimalnosti dobijenog početnog rešenja, tj. da li je dobijeni bazični plan optimalan, radi se na sledeći način. Za svako polje tabele, u kome cena nije opisana kružićem, formiraju se tzv. lanci. Lanac predstavlja zatvoreni poligon u čijem je jednom temenu cena polja, za koje se lanac formira, dok su na ostalim temenima poligona cene sa kružićima. Cena posmatranog polja opisana je kvadratićima u lancu. Svi uglovi lanca su pravi. Broj temena svakog lanca je paran i najmanje jednak 4, a najviše $m+n$. Za svako polje tabele bez kružića (*nebazna polja*) može da se formira jedan i samo jedan lanac. Sa leve gornje strane lanca stavlja se uvek oznaka polja za koje se lanac formira. Konstruišu se lanci za sva nebazna polja tabele (polja gde nema kružića). Za polje (1,3) lanac je oblika kao na slici 2.12.3.



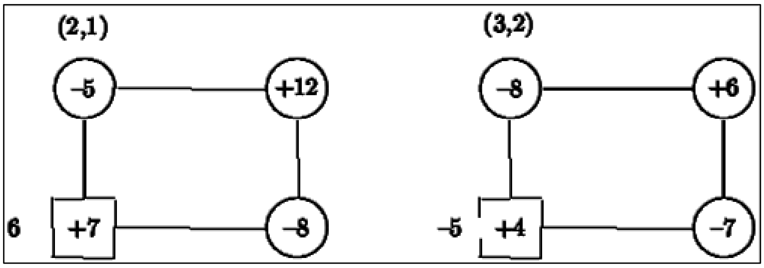
Slika 2.12.3. *Lanac za polje 1,3*

Ispred cene u kvadratiću stavlja se znak +, a ispred cene u lancu (krećući se u proizvoljnu stranu lanca od kvadratića) znak –, zatim +, itd.

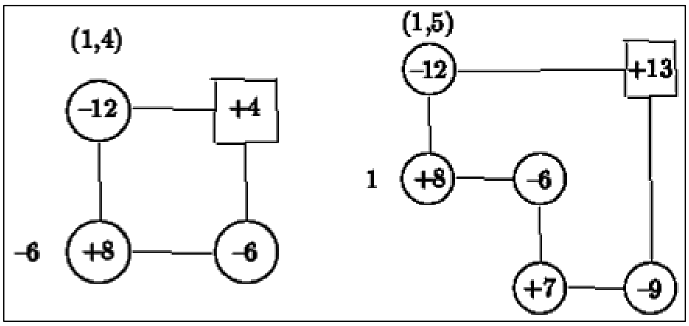
Znaci se menjaju naizmenično ispred cena u temenima lanca. Sve cene u temenima jednog lanca se sabiraju uzimajući u obzir i stavljene znake. Dobijeni zbir se naziva ***karakteristika lanca***. Ona se piše ispod oznake polja za koje je lanac formiran (dole levo). Karakteristika lanca za polje (1,3) je:

$$k_{13} = 1 - 12 + 8 - 14 = -17$$

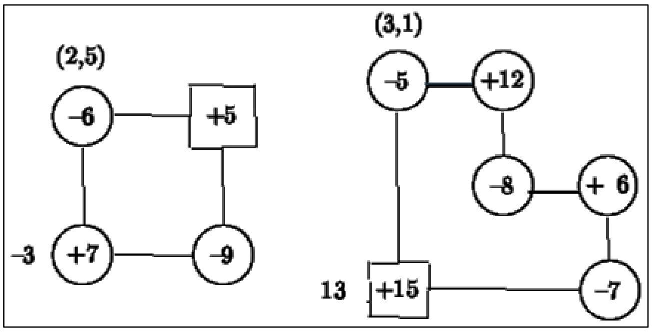
Lanci i njihove karakteristike za ostala polja neoznačena kružićima predstavljeni su slikama od 2.12.4. do 2.12.8.



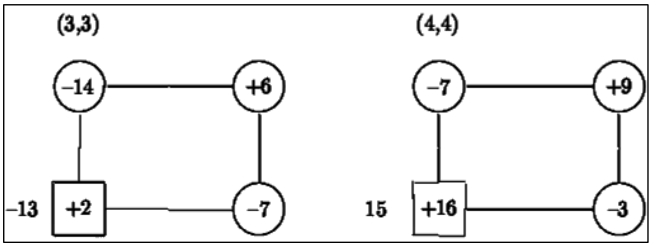
Slika 2.12.4. *Lanac za polja 2,1 i 3,2*



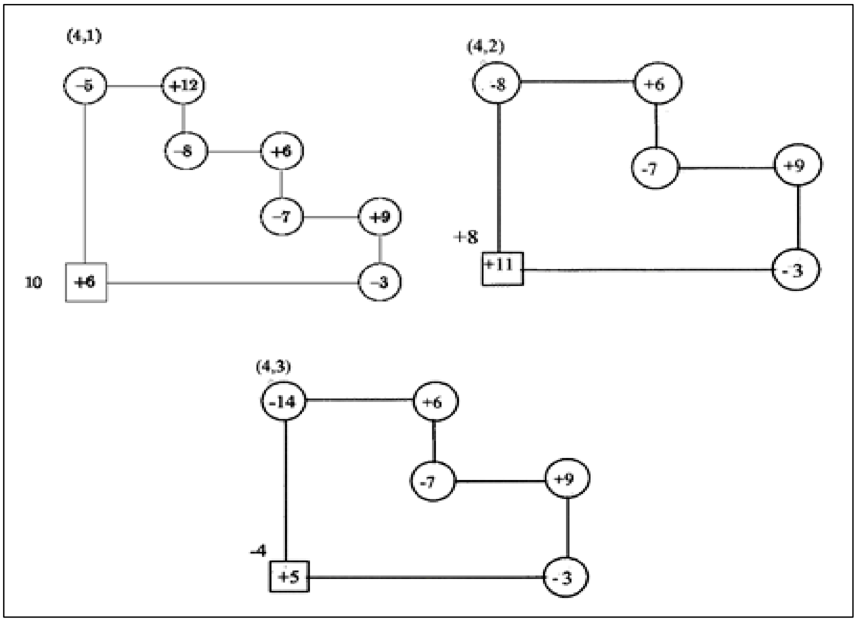
Slika 2.12.5. *Lanac za polja 1,4 i 1,5*



Slika 2.12.6. *Lanac za polja 2,5 i 3,1*



Slika 2.12.7. *Lanac za polja 3,3 i 4,4*



Slika 2.12.8. *Lanac za polja 4,1, 4,2 i 4,3*

Da bi se ustanovila optimalnost početnog baznog plana, ispituju se promene cena prevoza ako se u proizvoljno polje, neoznačeno kružićem, unese jedinična dostava. Pri tome treba voditi računa da jedinice robe za prevoz po vrstama budu jednaki mogućnostima otpremnih stanica, a zbirovi po kolonama jednaki potrebama prijemnih stanica. Tako, na primer, uvodeći jediničnu dostavu u polje (1,3) (x_{13} = 1) obavezno je da se umanji za jedinicu neka druga dostava koja se otprema iz stanice A_1, a takođe da se umanji za jedinicu i dostava trećoj prijemnoj stanici od drugih otpremnih stanica. U konkretnom slučaju umanjuju se dostave u poljima (1,2) i (2,3). Očigledno, smanjenje za jedinicu u polju (2,3) dovodi do uvećanja za jedinicu robe u polju (1,3), ali sada druga stanica B_2 prima jedinicu robe manje. U otpremnoj stanici A_2 našao se višak od jedne jedinice, pa se dostava povećava za jedinicu u polju (2,2). Pri tome, dostava se smanjila za jedinicu u polju čije su cene označene sa –, a povećavala se u poljima čije su cene označene sa +.

Znači, lanac pokazuje polja u kojima mora da se izvrši izmena bazičnih promenljivih, ako se preko polja bez kružića uvede dostava od otpremne do odgovarajuće prijemne stanice. U daljem tekstu se razmatra kakav uticaj na cenu prevoženja imaju ovakve izmene u dostavama. Na primer, ukoliko se uvede jedinica robe za prevoz u polje (1,3), ukupna cena prevoza se povećava za jednu novčanu jedinicu (jer je c_{13}=1). Zatim, ona se povećava još za 8, jer se dodaje jedinica robe u polju (2,2), ali se smanjuje za 12 i 14 novčanih jedinica zbog oduzimanja jedne jedinice robe u poljima (1,2) i (2,3). Prema tome, procena iznosi: 1 – 14 + 8 – 12 = – 17.

Broj –17 je, dakle, karakteristika lanca za polje (1,3) i pokazuje koliko se novčanih jedinica uštedi ako se uvede jedinična dostava u polje (1,3). Ukupno je 6 negativnih karakteristika lanaca za polja gde nema kružića, kao što je prikazano u tabeli 2.12.54.

Tabela 2.12.54. *Određivanje karakteristika lanaca*

OS \ PS	PS	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	36	5 / 13	12 / 23	1 / $\boxed{-17}$	4 / -6	13 / 1
A_2	23	7 / 6	8 / 1	14 / 15	6 / 7	5 / -3
A_3	29	15 / 13	4 / -5	2 / -13	7 / 14	9 / 15
A_4	12	6 / 10	11 / 8	5 / -4	16 / 15	3 / 12

Zaključuje se da pojava najmanje jedne negativne karakteristike pokazuje da bazični plan nije optimalan. Polje (1,3) ima najmanju od negativnih karakteristika (-17) pa znači da promena po lancu polja (1,3) najviše umanjuje ukupnu cenu prevoza po jedinici robe. Zato će se u polje (1,3) staviti najveća moguća dostava; ona je jednaka najmanjoj od dostava u poljima sa negativnim cenama. U našem slučaju 15 jedinica robe se oduzme od dostava polja (1,2) i (2,3) (15=min{15,23}) i dodaje se dostavama u poljima (1,3) i (2,2). Tako se dobija novi bazični plan prevoza. On će biti bolji od početnog bazičnog polja, jer će se ukupni troškovi prevoza smanjiti za $15 \cdot 17 = 255$ novčanih jedinica, tj. troškovi prevoza prema drugom bazičnom planu su: $F_2 = 870 - 255 = 615$ n.j. (novčanih jedinica). Drugi bazični plan (početno rešenje nakon prve iteracije) dat je u tabeli 2.12.55.

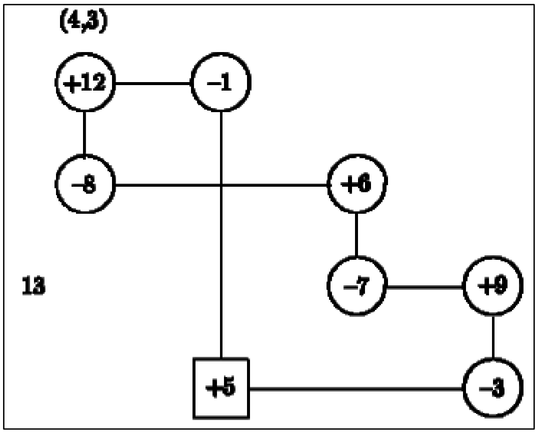
Tabela 2.12.55. *Drugo bazično rešenje*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 13	12 8	1 15	4	13
A_2	23	7	8 16	14	6 7	5
A_3	29	15	4	2	7 14	9 15
A_4	12	6	11	5	16	3 12

Isti postupak provere optimalnosti bazičnog plana se primenjuje i u narednom koraku. Lanci polja bez kružića se lako uočavaju. Vrednosti karakteristika lanaca iznose:

k_{14}= 4-12+8-6 = -6 k_{15}=13-12+8-6+7-9= 1
k_{21}= 7-5+12-8 = 6, k_{23}= 14-8+12-1 = 15, k_{25}= 5-6+7-9 = -3,
k_{31}= 15-7+6-8+12-5 = 13, k_{32} = 4-7+6-8 = -5, k_{33} = 2-1+12-8+6-7=4,
k_{41}= 6-5+12-8+6-7+9-3 = 10, k_{42} = 11-8+6-7+9-3 = 8, k_{44} = 16-7+9-3 = 15

Posebno je zanimljiv lanac polja (4,3), koji je prikazan na slici II-10.



Slika 2.12.9. *Lanac za polje 4,4*

Tabela 2.12.56. prikazuje karakteristike lanaca za nebazna polja posle prve iteracije.

Najmanja negativna karakteristika odgovara lancu polja (1,4). Karakteristika lanca ovog polja je k_{14} = 4–6 +8–12 = –6. Kako je x_{14}=7=min{7,8},ovu količina robe se oduzima od dostava u poljima sa negativnim cenama i dodaje se dostavama u poljima sa pozitivnim cenama u temenima lanca tog polja. Ukupno smanjenje troškova prevoza za treće bazično rešenje iznosi 7·6 = 42 novčane jedinice, pa su ukupni troškovi prevoza prema trećem bazičnom planu (nakon druge iteracije):

$F_3 = F_2 - 6{\cdot}7 = 615 - 42 = 573$ novčane jedinice. Treći bazični plan (početno rešenje nakon druge iteracije) dat je u tabeli 2.12.57.

Tabela 2.12.56. *Određivanje karakteristika lanaca drugog bazičnog rešenja*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 / 13	12 / 8	1 / 15	4 / -6	13 / 1
A_2	23	7 / 6	8 / 16	14 / 15	6 / 7	5 / -3
A_3	29	15 / 13	4 / -5	2 / 4	7 / 14	9 / 15
A_4	12	6 / 10	11 / 8	5 / 13	16 / 15	3 / 12

Tabela 2.12.57. *Treće bazično rešenje*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 / 13	12 / 1	1 / 15	4 / 7	13
A_2	23	7	8 / 23	14	6	5
A_3	29	15	4	2	7 / 14	9 / 15
A_4	12	6	11	5	16	3 / 12

Sada se dobijaju sledeće karakteristike lanaca:

$k_{15} = 13\text{-}4\text{+}7\text{-}9 = 7$, $k_{21} = 7\text{-}5\text{+}12\text{-}8 = 6$, $k_{23} = 14\text{-}1\text{+}12\text{-}8 = 17$,
$k_{24} = 6\text{-}4\text{+}12\text{-}8 = 6$, $k_{25} = 5\text{-}8\text{+}12\text{-}4\text{+}7\text{-}9 = 3$, $k_{31} = 15\text{-}7\text{+}4\text{-}5 = 7$,
$k_{32} = 4\text{-}12\text{+}4\text{-}7 = -11$, $k_{33} = 2\text{-}7\text{+}4\text{-}1 = 6$, $k_{41} = 12\text{-}3\text{+}9\text{-}7\text{+}4\text{-}5 = 4$,
$k_{42} = 11\text{-}3\text{+}9\text{-}7\text{+}4\text{-}12 = 2$, $k_{43} = 5\text{-}3\text{+}9\text{-}7\text{+}4\text{-}1 = 7$, $k_{44} = 16\text{-}3\text{+}9\text{-}7 = 15$.

Kako je karakteristika lanca za polje (3,2) najmanja negativna karakteristika $k_{32} = 4-7+4-12 = -11$, to je potrebno da se izvrši izmena dostava po ovom lancu. Sada je $x_{32} = \min\{14, 1\} = 1$. Četvrti bazični plan (početno rešenje nakon treće iteracije) dat je u tabeli 2.12.58.

Ukupni troškovi prevoza po ovom planu prevoženja su: $F_4 = F_3 - 11{\cdot}1 = 573 - 11 = 562$ novčane jedinice. Najmanju negativnu karakteristiku u ovom bazičnom planu ima lanac polja (2,5): $k_{25} = 5-8+4-9 = -8$. Kako je: $x_{25} = 15 = \min\{x_{35}, x_{22}\}$, dobija se da je funkcija cilja: $F_5 = F_4 - 15{\cdot}8 = 562 - 120 = 442$ n.j. Peti bazični plan (početno rešenje nakon četvrte iteracije) dat je u tabeli 2.12.59.

Tabela 2.12.58. *Četvrto bazično rešenje*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 / 13	12	1 / 15	4 / 8	13
A_2	23	7	8 / 23	14	6	5
A_3	29	15	4 / 1	2	7 / 13	9 / 15
A_4	12	6	11	5	16	3 / 12

Tabela 2.12.59. *Peto bazično rešenje*

OS \ PS		B_1	B_2	B_3	B_4	B_5
	100	13	24	15	21	27
A_1	36	5 / 13	12	1 / 15	4 / 8	13
A_2	23	7	8 / 8	14	6	5 / 15
A_3	29	15	4 / 16	2	7 / 13	9
A_4	12	6	11	5	16	3 / 12

Najmanja negativna karakteristika u petom baznom planu ima lanac polja (2,4): k_{24}= 6–7+4–8=–5. Kako je x_{24} = min{x_{22}, x_{34}}= 8, to su troškovi prevoza po šestom baznom planu (nakon pete iteracije) jednaki: $F_6 = F_5 - 5{\cdot}8$ = 442–40=402. Šesti bazični plan (početno rešenje nakon pete iteracije) dat je u tabeli 2.12.60.

Najmanju negativnu karakteristiku u šestom baznom planu ima lanac polja (3,3): k_{33}=2–7+4–1 = –2, pa kako je x_{33} = min{x_{34}, x_{13}} = 5, to su ukupni troškovi prevoza: $F_7 = F_6 - 2{\cdot}5$= 402–10= 392 n.j. Sedmi bazični plan dat je u tabeli 2.12.61.

Tabela 2.12.60. *Šesto bazično rešenje*

OS \ PS	*PS*	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	5 / **13**	12	1 / **15**	4 / **8**	13
A_2	**23**	7	8	14	6 / **8**	5 / **15**
A_3	**29**	15	4 / **24**	2	7 / **5**	9
A_4	**12**	6	11	5	16	3 / **12**

Tabela 2.12.61. *Sedmo bazično rešenje – prvi optimalni plan*

OS \ PS	*PS*	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	5 / **13**	12	1 / **10**	4 / **13**	13
A_2	**23**	7	8	14	6 / **8**	5 / **15**
A_3	**29**	15	4 / **24**	2 / **5**	7	9
A_4	**12**	6	11	5	16	3 / **12**

Kako su sve karakteristike lanaca polja bez kružića pozitivne, to znači da je plan prevoza optimalan. Na taj način minimalni troškovi prevoza su: $F_{min} = F_7$ = 392 n.j. Polje (2,1) u poslednjoj tabeli ima karakteristiku koja je 0, pa ako se po lancu polja izvrši maksimalno moguća zamena dostava, (x_{21} = min{x_{24}, x_{11}} = 8)), to se ukupna cena prevoza (F_{min} =392) neće promeniti. To znači da zadatak ima dva optimalna plana. Drugi optimalni plan je prikazan u tabeli 2.12.62.

Tabela 2.12.62. *Drugi optimalni plan*

OS \ PS	*PS*	B_1	B_2	B_3	B_4	B_5
OS	**100**	**13**	**24**	**15**	**21**	**27**
A_1	**36**	5 / **5**	12	1 / **10**	4 / **21**	13
A_2	**23**	7 / **8**	8	14	6	5 / **15**
A_3	**29**	15	4 / **24**	2 / **5**	7	9
A_4	**12**	6	11	5	16	3 / **12**

2.12.3.2. Metod koeficijenata - potencijala

Metod koeficijenata predstavlja uprošćeni metod raspodele koji je razvio američki matematičar Dantzig. Ovde ne treba konstruisati lance za sva polja neobeležena kružićima, jer se pomoću “koeficijenata” vrsta i kolona neposredno dobijaju lanci sa negativnim karakteristikama. Provera optimalnosti baznog plana izvodi se samo pomoću koeficijenata vrsta i kolona. Ti koeficijenti se biraju tako da njihov zbir bude jednak ceni u kružiću polja koje se nalazi u preseku vrste i kolone čiji se koeficijenti sabiraju, tj. ako se od cene u kružiću oduzmu koeficijenti posmatrane vrste, dobijaju se koeficijenti odgovarajuće kolone i obrnuto. Koeficijenti mogu biti pozitivni, negativni ili jednaki nuli.

Postupak za nalaženje optimalnog rešenja pomoću metoda koeficijenata ima sledeći tok:

1) Početni program (rešenje) se odredi po nekom od razrađenih postupaka.
2) Za angažovane rute u početnom baznom rešenju (polja opterećena pozitivnim komponentama početnog rešenja) formira se tabela jediničnih troškova transporta c_{ij}.
3) Svakom redu i svakoj koloni matrice troškova se dodeljuje po jedan pozitivan broj – potencijal ili simpleksni multiplikator. Potencijali redova su u_i (i= l, 2,..., m), a potencijali kolona su v_j (j=1, 2,..., n). Potencijali u_j i v_j se biraju tako da je njihova suma jednaka odgovarajućoj vrednosti troška angažovane rute c_{ij}, odnosno:

   $$u_i + v_j = c_{ij} \qquad (2.12.16.)$$

   pri čemu indeksi (i, j) označavaju opterećeno polje u matrici troškova. Obzirom na to da u trenutku određivanja vrednosti potencijala u_j i v_j postoji (n+m) nepoznatih potencijala, a (n+m-1) opterećenih polja, odnosno jednačina, jednoj nepoznatoj se može dati proizvoljna vrednost. Obično se potencijal prvog reda u_1 izjednačava sa nulom, a ostale vrednosti potencijala jednoznačno se izračunavaju iz jednačine $u_i + v_j = c_{ij}$. U praksi se sreće slučaj da se, zbog nedostatka jedne nepoznate, potencijalu koji se najveći broj puta pojavljuje u sistem jednačina dodeljuje vrednost nula.
4) Nakon utvrđivanja vrednosti za potencijale u_j i v_j izračunavaju se vrednosti c'_{ij} za prazna polja u matrici troškova prema obrascu:

   $$c'_{ij} = u_i + v_j \qquad (2.12.17.)$$

5) Vrednosti c'_{ij} se upoređuju s vrednostima odgovarajućih troškova c_{ij} u početnoj tabeli troškova. Ako su za sva polja prvobitni troškovi c_{ij} veći ili jednaki od dobijenih odgovarajućih troskova c'_{ij}, onda je dobijeno rešenje optimalno i ne može se dalje poboljšavati. Polja kod kojih je zadovoljen uslov $c'_{ij} - c_{ij} > 0$ pružaju mogućnost formiranja boljeg rešenja. Kriterijum za izbor polja čija komponenta vektora X ulazi u bazu je:

   $$\max[(c'_{ij} - c_{ij}) > 0]. \qquad (2.12.18.)$$

Da se postojeća ravnoteža početnog rešenja po redovima i kolonama (ravnoteža ponude i potražnje) ne bi poremetila, postupak uvođenja nove komponente x_{rk} na mestu sa najvećom razlikom $c'_{ij} - c_{ij} > 0$ i izlaska iz baze neke od komponenata prvobitnog rešenja mora biti izveden unutar jednog mnogougaonika koji se dobija zamišljenim kretanjem topa u šahu.

Počinje se od polja (r,k), uz moguća skretanja samo na angažovanim (opterećenim poljima). Mnogougaonik se završava u posmatranom polju (r, k), zahvativši manji ili veći broj angazovanih polja, odnosno komponenata x_{ij} početnog rešenja. Temena dobijenog mnogougaonika se označavaju naizmenicno sa (+) i (–), pri čemu je teme (r, k) oznaceno sa (+) jer se u to polje dovodi određena (maksimalno moguća) količina tereta. Među količinama sa negativnim temenima izdvaja se najmanja i dodaje količinama čija su temena sa znakom (+), a oduzima od količina čija su temena sa znakom (–).

Rezultat je promena količina na zahvaćenim temenima, ali je pri tome najvažnije da je izabrano polje (r, k) dobilo količinu koja je cirkulisala kroz mnogougaonik, a teme čija je količina cirkulisala ostalo je bez opterećenja, odnosno ispalo je iz programa. Na opisani način se može za svako nepopunjeno polje konstruisati samo jedan jedini mnogougaonik.

6) Sa novodobijenim baznim rešenjem se postupa prema koracima 2), 3), 4) i 5) i postupak se ponavlja sve dok se u nekom ponovljenom koraku 5) ne utvrdi da ne postoji nijedno polje sa $c'_{ij} - c_{ij} > 0$. To je znak da je dobijeno optimalno rešenje.

Primer 2.12.4. Za određivanje optimalnog rešenja metodom potencijala koriste se polazni podaci kao u tabeli 2.12.63. To su podaci iz primera koji su korišćeni za određivanje početnog rešenja kod metoda Vogel-Kordinog postupka.

Tabela 2.12.63. *Početni podaci za primer urađen Vogel-Kordinim postupkom*

	PS	B_1	B_2	B_3	B_4	B_5
OS	**117**	**17**	**21**	**41**	**14**	**24**
A_1	**25**	**10**	**8**	**9**	**6**	**5**
A_2	**32**	**5**	**6**	**4**	**3**	**8**
A_3	**40**	**9**	**7**	**5**	**4**	**3**
A_4	**20**	**14**	**10**	**8**	**8**	**8**

Rešenje.

1) Za početno rešenje izabraćemo rešenje koje je dobijeno metodom severozapadnog ugla, kao što je prikazano u tabeli 2.12.64. Dobijeno je nedegenerisano rešenje X={17,8,13,19,22,14,4, 20} sa tačno $(n+m-1) = 8$ nenegativnih komponenata vektora X. Troškovi transporta prema dobijenom rešenju iznose: $F(x_0)$=10·17+ 8·8+ 6·13+ 4·19+ 5·22+ 4·14+ 3·4+ 8·20 = 726 n.j.

Tabela 2.12.64. *Početno rešenje dobijeno metodom severozapadnog ugla*

OS \ PS		B_1	B_2	B_3	B_4	B_5
OS	**117**	**17**	**21**	**41**	**14**	**24**
A_1	25	10 17	8 8	9	6	5
A_2	32	5	6 13	4 19	3	8
A_3	40	9	7	5 22	4 14	3 4
A_4	20	14	10	8	8	8 20

2) U tabeli 2.12.65. dati su troškovi za polja koja su angažovana u početnom programu. Ti troškovi c_{ij} su zaokrućeni kako bi se razlikovali od troskova c'_{ij} koji se dobijaju sabiranjem odgovarajućih potencijala u_i i v_j.

Tabela 2.12.65. *Troškovi za početno rešenje*

OS \ PS		B_1	B_2	B_3	B_4	B_5	u_i
OS	**117**	**17**	**21**	**41**	**14**	**24**	
A_1	25	10	8	6	5	4	0
A_2	32	8 +3	6	4	3	2	-2
A_3	40	9	7	5	4	3	-1
A_4	20	14	12 +2	10 +2	9 +1	8	4
v_j		10	8	6	5	4	

3) Potencijali se određuju tako što se stavi da je $u_1=0$, a zatim se računa: $v_1=c_{11}-u_1=10$; $v_2=c_{12}-u_1 = 8$ i tako redom dok se ne odrede svi potencijali.
4) Vrednosti c'_{ij} za neangažovana polja (nebazna polja) računaju se prema izrazu: $c'_{ij}=u_i+v_j$, npr. c_{21}=10+2= 8; c_{31}=10+1= 9.
5) Primenom kriterijuma $max[(c'_{ij} – c_{ij})> \theta]$ bira se polje za ulazak u bazu. Uslov zadovoljava polje (a_2,b_1) sa vrednoscu c'_{21}= 8 i u to polje treba dovesti količinu θ=13 koja predstavlja minimalnu količinu na negativnim temenima mnogougaonika. U tabeli 2.12.66. izvršeno je poboljšanje početnog programa tako što je polju (a_2,b_1) dodeljena vrednost θ =13 a polje (a_2,b_2) ostaje bez 13 jedinica težine robe i ono napušta bazu. Količina robe u poljima (a_1,b_1) i (a_2,b_2) je $x_{11}=4$ i $x_{12}=21$ jedinica težine, kao što je prikazano u tabeli 2.12.66.

Tabela 2.12.66. *Troškovi nakon prvog poboljšanja*

OS \ PS		B_1	B_2	B_3	B_4	B_5	u_i
OS	**117**	**17**	**21**	**41**	**14**	**24**	
A_1	25	10 -θ 4	8 +θ 21	6	5	4	0
A_2	32	8 +θ 13	6 -θ 0	4 19	3	2	-2
A_3	40	9	7	5 22	4 14	3 4	-1
A_4	20	14	12	10	9	8 20	4
v_j		10	8	6	5	4	

6) Ponavljanjem koraka 2), 3), 4) i 5) dobija se novo bazno rešenje, kao što je prikazano u tabeli 2.12.67.

Tabela 2.12.67. *Troškovi nakon drugog poboljšanja*

	PS	B_1	B_2	B_3	B_4	B_5	u_i
OS	**117**	**17**	**21**	**41**	**14**	**24**	
A_1	**25**	**10** -θ **0**	**8** **21**	**9**	**8** +θ **4**	**7**	**0**
A_2	**32**	**5** +θ **17**	**3**	**4** -θ **15**	**3**	**2**	**-5**
A_3	**40**	**6**	**4**	**5** +θ **26**	**4** -θ **10**	**3** **4**	**-4**
A_4	**20**	**11**	**9**	**10**	**9**	**8** **20**	**1**
v_j		**10**	**8**	**8**	**8**	**7**	

Nakon ponovljene celokupne procedure (ponavljanje koraka 2, 3, 4 i 5) dobija se novo bazno rešenje, kao što je prikazano u tabeli 2.12.68.

Tabela 2.12.68. *Troškovi nakon trećeg poboljšanja*

	PS	B_1	B_2	B_3	B_4	B_5	u_i
OS	**117**	**17**	**21**	**41**	**14**	**24**	
A_1	**25**	**8**	**8** **21**	**7**	**6** **4**	**5**	**0**
A_2	**32**	**5** **17**	**5**	**4** **15**	**3**	**2**	**-3**
A_3	**40**	**6**	**6**	**5** -θ **6**	**4** **10**	**3** +θ **24**	**-2**
A_4	**20**	**11**	**11**	**10** +θ **20**	**9**	**8** -θ **0**	**3**
v_j		**8**	**8**	**7**	**6**	**5**	

Vrednosti potencijala i veličina c'_{ij} dobijene u sledećoj iteraciji date su u tabeli 2.12.69.

Tabela 2.12.69. *Troškovi u četvrtoj iteraciji*

	PS	B_1	B_2	B_3	B_4	B_5	u_i
OS	**117**	**17**	**21**	**41**	**14**	**24**	
A_1	**25**	**8**	**8** **21**	**7**	**6** **4**	**5**	**0**
A_2	**32**	**5** **17**	**4**	**4** **15**	**3**	**2**	**-3**
A_3	**40**	**6**	**6**	**5** **6**	**4** **10**	**3** **24**	**-2**
A_4	**20**	**9**	**9**	**8** **20**	**7**	**6**	**1**
v_j		**8**	**8**	**7**	**6**	**5**	

Kako nijedno polje ne zadovoljava kriterijum za ulazak u bazu, to znači da je dobijeno rešenje optimalno. Vrednost dobijenog optimalnog rešenja jednaka je:

$$F(x_1) = 17\cdot 5+21\cdot 8+ 15\cdot 4+6\cdot 5+20\cdot 8+4\cdot 6+ 10\cdot 4+24\cdot 3 = 639 \text{ novčanih jedinica}$$

Ušteda koja je postignuta u odnosu na početno rešenje, koje je dobijeno metodom severozapadnog ugla, je:

$$\Delta F = F(x_0) - F(x_1) = 726 - 639 = 87 \text{ novčanih jedinica}$$

Primer 2.12.5. Za transportni zadatak koji je postavljen u tabeli 2.12.70. bazično rešenje je određeno metodom najmanje jedinične cene u matrici.

Tabela 2.12.70. *Početno rešenje problema transporta*

	PS	**B$_1$**	**B$_2$**	**B$_3$**	**u$_i$**
OS	**400**	**140**	**100**	**160**	***kv***
A$_1$	**90**	2	5	2 / **90**	**0**
A$_2$	**200**	4 / **30**	1 / **100**	5 / **70**	**3**
A$_3$	**110**	3 / **110**	6	8	**2**
v$_j$	***kk***	**1**	**-2**	**2**	

U tabeli II-72 sa ***kk*** označavaju se koeficijenti kolona, a sa ***kv*** koeficijenti vrste. Najpre će se za koeficijent vrste u_1 odrediti vrednost nula, tj. $u_1=0$. Ostale koeficijente vrsta i kolona se određuju pomoću izraza:

$$c_{ij} = u_i + v_j$$

koji važi za sve cene označene kružićima u kojima važi $x_{ij} > 0$. Na taj način se dobija:

$$c_{13} = u_1 + v_3 \quad \Rightarrow \quad v_3 = c_{13} - u_1 = 2 - 0 = 2$$
$$c_{23} = u_2 + v_3 \quad \Rightarrow \quad u_2 = c_{23} - v_3 = 5 - 2 = 3$$
$$c_{22} = u_2 + v_2 \quad \Rightarrow \quad v_2 = c_{22} - u_2 = 1 - 3 = -2$$
$$c_{21} = u_2 + v_1 \quad \Rightarrow \quad v_1 = c_{21} - u_2 = 4 - 3 = 1$$
$$c_{31} = u_3 + v_1 \quad \Rightarrow \quad u_3 = c_{31} - v_1 = 3 - 1 = 2$$

Provera optimalnosti plana kod ovog metoda zasniva se na sledećoj teoremi:

Teorema 2.12.4. Ako je za sva bazična polja plana ispunjeno:

$$c_{ij} = u_i + v_j \qquad (2.12.19.)$$

a za slobodna polja

$$c_{ij} \geq u_i + v_j \qquad (2.12.20.)$$

onda je bazični plan optimalan.

Dokaz: Označimo sa $\{x_{ij}\}$ plan prevoza i sistem koeficijenata vrsta i kolona sa (u_i, v_j). Ukupni troškovi prevoza su:

$$F = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} (u_i + v_j) x_{ij}$$

Za plan $\{x'_{ij}\}$ ovi troškovi glase:

$$F' = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij}'$$

Promenljive x'_{ij} se u nekim poljima poklapaju sa x_{ij} iz plana $\{x_{ij}\}$ ali su zato u nekim poljima gde su $x_{ij} = 0$, one pozitivne.

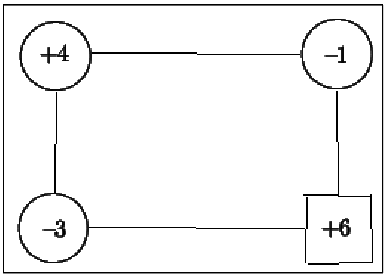
U poljima gde se x_{ij} i x'_{ij} poklapaju ispunjeno je $u_i + v_j = c_{ij}$, a u poljima gde je $x_{ij}= 0$, $x'_{ij} > 0$ ispunjeno je $u_i + v_j < c_{ij}$ pa sledi:

$$F' = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\ x_{ij}' \geq \sum_{i=1}^{m}\sum_{j=1}^{n}(u_i + v_j)x_{ij}' \geq \sum_{i=1}^{m}\sum_{j=1}^{n}(u_i + v_j)x_{ij} = F$$

što znači da se izmenom plana $\{x_{ij}\}$ troškovi prevoza ne mogu umanjiti tj. plan $\{x_{ij}\}$ sa koeficijentima koji zadovoljavaju uslove (2.12.16.) i (2.12.17.) daju optimalan plan.

Sada kriterijum optimalnosti glasi: ***Ako su razlike $\Delta_{ij} = c_{ij} - (u_i + v_j)$ nenegativne za nebazična polja, onda je bazični plan optimalan.***

Može se još pokazati da su ove razlike jednake karakteristikama lanaca polja sa neoznačenim cenama. Konkretno lanac polja na primer (3,2) ima oblik koji je dat na slici 2.12.10.



Slika 2.12.10. *Lanac za polje 3,2*

$$k_{32} = c_{32} - (u_3 + v_2) = 6- (2\text{-}2) = 6.$$

Izračunavaju se sve karakteristike nebazičnih polja:

$$k_{11} = 2 - 1 - 0 = 1$$
$$k_{12} = 5 + 2 - 0 = 7$$
$$k_{32} = 6 + 2 - 2 = 6$$
$$k_{33} = 8 - 2 - 2 = 4$$

U našem primeru ne javlja se nijedna negativna karakteristika, što govori da je dobijeni bazični plan, metodom najmanjih jediničnih cena u matrici, optimalan. Bazični plan se može poboljšavati sve dok se pojavljuje makar jedna negativna karakteristika.

Ukupna cena prevoženja u našem primeru je:

$$F_{\min} = 2 \cdot 90 + 4 \cdot 30 + 1 \cdot 100 + 5 \cdot 70 + 3 \cdot 110 = 1080 \text{ n.j.}$$

2.12.3.3. Rešavanje transportnog problema pomoću programskih paketa

U daljem tekstu, na konkretnim primerima, biće objašnjeni postupci rešavanja transportnog problema. Pri tome, koristiće se sledeći programski paketi: *LINDO*, *LINGO* i *QM for Windows*.

Primer 2.12.6. Robu iz četiri skladišta S_i (i=1, 2, 3, 4) potrebno je dostaviti u tri preduzeća P_j (j=1, 2, 3). Količina robe u skladištima (izvorima), potrebna količina u preduzećima (ponorima) i cena transporta jedinice robe iz određenog skladišta u određeno preduzeće dati su u tabeli 2.12.71. Potrebno je rešiti zadatak korišćenjem LINDO programskog paketa.

Tabela 2.12.71. *Početni podaci transportnog problema*

Preduzeća / *Skladišta*	P_1 (140)	P_2 (35)	P_3 (105)
S_1 (70)	50	60	0
S_2 (105)	40	20	15
S_3 (70)	30	45	20
S_4 (35)	35	40	25

Rešenje: Optimalno rešenje transportnog problema korišćenjem LINDO programskog paketa se dobija tako što se najpre, za postavljeni problem, napiše matematički model.

Funkcija cilja je:

$$\min F(x) = 50x_{11}+60x_{12}+40x_{21}+20x_{22}+15x_{23}+30x_{31}+45x_{32}+20x_{33}+35x_{41}+40x_{42}+25x_{43}$$

Ograničenja po redovima su:

$$\begin{aligned} x_{11}+x_{12}+x_{13}&=70 \\ x_{21}+x_{22}+x_{23}&=105 \\ x_{31}+x_{32}+x_{33}&=70 \\ x_{41}+x_{42}+x_{43}&=35 \end{aligned}$$

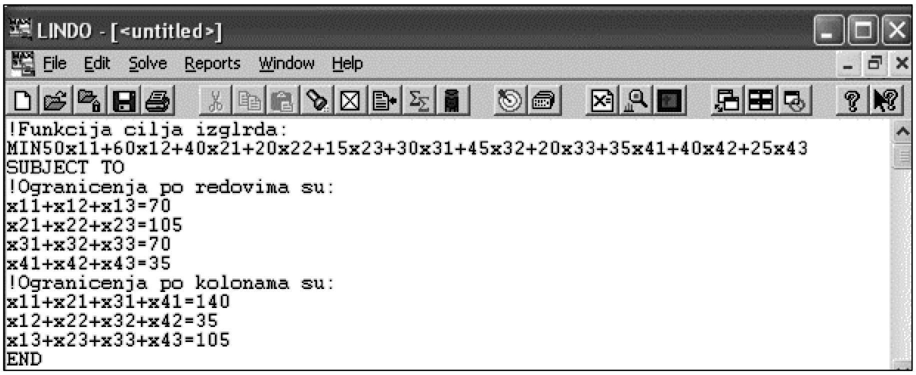
Ograničenja po kolonama su:

$$\begin{aligned} x_{11}+x_{21}+x_{31}+x_{41}&=140 \\ x_{12}+x_{22}+x_{32}+x_{42}&=35 \\ x_{13}+x_{23}+x_{33}+x_{43}&=105 \end{aligned}$$

Ovako formulisan matematički model problema potrebno je upisati u radni prostor programskog paketa LINDO, na sledeći način:

```
MIN 50x11+60x12+40x21+20x22+15x23+30x31+45x32+20x33+35x41+40x42+25x43
SUBJECT TO
x11+x12+x13=70
x21+x22+x23=105
x31+x32+x33=70
x41+x42+x43=35
x11+x21+x31+x41=140
x12+x22+x32+x42=35
x13+x23+x33+x43=105
END
```

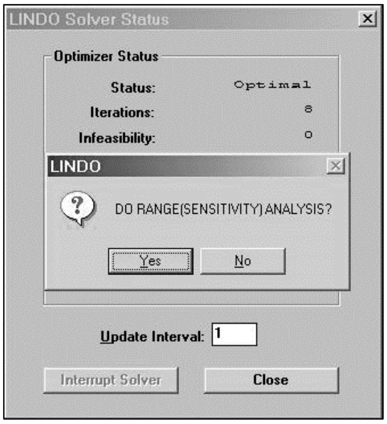
Izgled ekrana sa postavkom za rešavanje transportnog zadatka, dat je na slici 2.12.11.



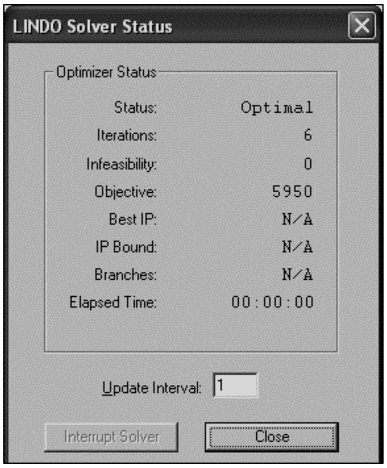
Slika 2.12.11. *Izgled postavljenog TP u LINDO programu*

Izborom opcije Solve, ili preko ikone Solve, dobija se rešenje sledećim redom:

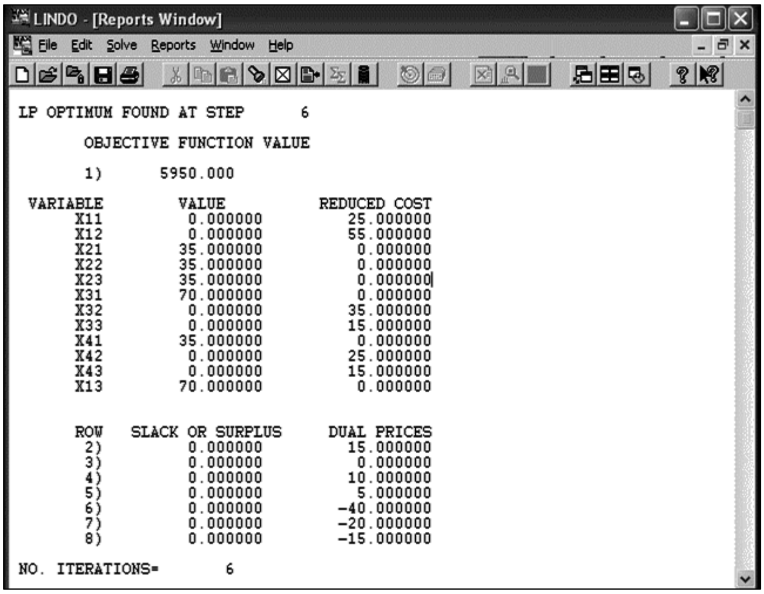
- Prvo se otvara dijalog prozor, slika 2.12.12., gde izborom opcije Yes potvrđuje da se želi i postoptimalna analiza rešenja.
- Zatim se dobij prikaz stanja LINDO rešavača, slika 2.12.13., na kome vidi da je rešenje optimalno, dobijeno u 6 iteracija, da je nepodesnost (Infeasibility) modela 0, da je vrednost funkcije cilja (minimum) 5950, dok se ostali parametri odnose na celobrojno programiranje (Integer Programmng), koji imaju oznaku N/A – Not Available, obzirom na to da celobrojno programiranje nije korišćeno.
- Zatvaranjem ovog prozora, može se pogledati rešenje, koje je dato u Reports Window, kao što je prikazano slikom 2.12.14.



Slika 2.12.12. *Dijalog prozor sa upitom za postoptimalnu analizu*



Slika 2.12.13. *Status LINDO rešavača*



LP OPTIMUM FOUND AT STEP 6

OBJECTIVE FUNCTION VALUE

1) 5950.000

VARIABLE	VALUE	REDUCED COST
X11	0.000000	25.000000
X12	0.000000	55.000000
X21	35.000000	0.000000
X22	35.000000	0.000000
X23	35.000000	0.000000
X31	70.000000	0.000000
X32	0.000000	35.000000
X33	0.000000	15.000000
X41	35.000000	0.000000
X42	0.000000	25.000000
X43	0.000000	15.000000
X13	70.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	15.000000
3)	0.000000	0.000000
4)	0.000000	10.000000
5)	0.000000	5.000000
6)	0.000000	-40.000000
7)	0.000000	-20.000000
8)	0.000000	-15.000000

NO. ITERATIONS= 6

Slika 2.12.14. *Rešenje TP korišćenjem LINDO programa dato u Reports Window*

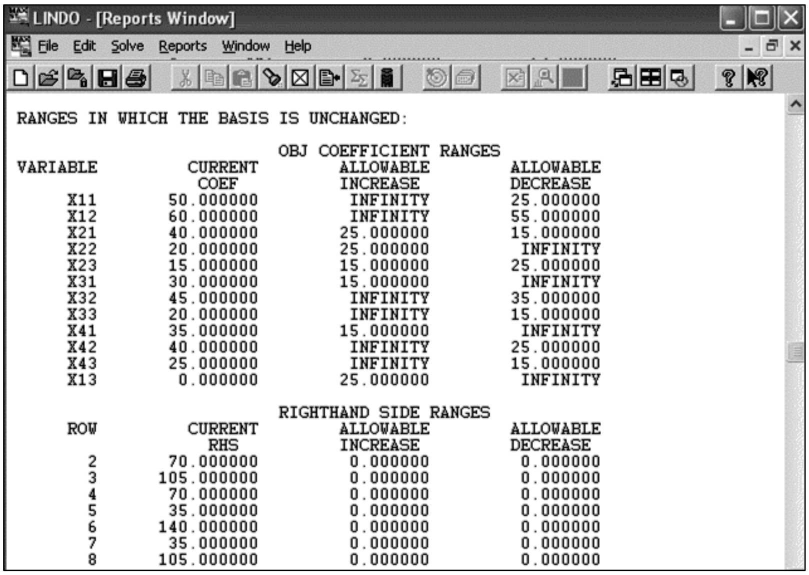
U prvom redu vse idi da je nađeno optimalno rešenje u šestom koraku, zatim da je vrednost funkcije cilja 5950, za vrednosti promenljivih (čija su imena navedena u koloni VARIABLE, a vrednosti u koloni VALUE):

X11	**0**.000000
X12	**0**.000000
X13	**70**.000000
X21	**35**.000000
X22	**35**.000000
X23	**35**.000000
X31	**70**.000000
X32	**0**.000000
X33	**0**.000000
X41	**35**.000000
X42	**0**.000000
X43	**0**.000000

U koloni SLACK OR SURPLUS su date vrednosti dopunskih promenljivih (LINDO koristi simplex algoritam rešavanja problema).

Kolona DUAL PRICES (dualne ili cene iz senke) nam govori o tome kako bi se uvođenje resursa koji nisu u rešenju odrazilo na funkciju cilja, odnosno koliko treba da se plati da bi se uveli dodatni resursi u rešenje. Ovo su ujedno i vrednosti promenljivih dualnog modela.

Ukoliko je izabrana opcija analize parametara (analiza osetljivosti – senzitivnosti) dobija se izveštaj o tome, kao što je prikazano na slici 2.12.15.



RANGES IN WHICH THE BASIS IS UNCHANGED:

OBJ COEFFICIENT RANGES

VARIABLE	CURRENT COEF	ALLOWABLE INCREASE	ALLOWABLE DECREASE
X11	50.000000	INFINITY	25.000000
X12	60.000000	INFINITY	55.000000
X21	40.000000	25.000000	15.000000
X22	20.000000	25.000000	INFINITY
X23	15.000000	15.000000	25.000000
X31	30.000000	15.000000	INFINITY
X32	45.000000	INFINITY	35.000000
X33	20.000000	INFINITY	15.000000
X41	35.000000	15.000000	INFINITY
X42	40.000000	INFINITY	25.000000
X43	25.000000	INFINITY	15.000000
X13	0.000000	25.000000	INFINITY

RIGHTHAND SIDE RANGES

ROW	CURRENT RHS	ALLOWABLE INCREASE	ALLOWABLE DECREASE
2	70.000000	0.000000	0.000000
3	105.000000	0.000000	0.000000
4	70.000000	0.000000	0.000000
5	35.000000	0.000000	0.000000
6	140.000000	0.000000	0.000000
7	35.000000	0.000000	0.000000
8	105.000000	0.000000	0.000000

Slika 2.12.15. *Rezultati analize osetljivosti za transportni zadatak*

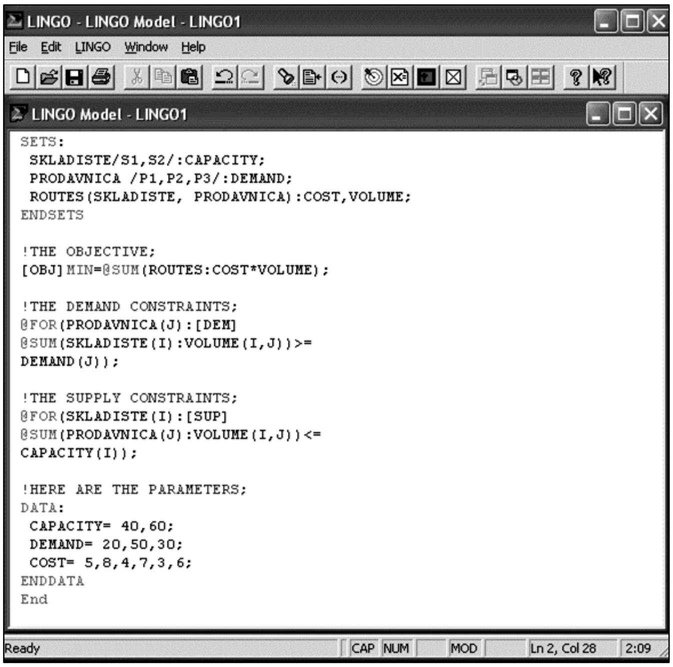
Ova analiza nam govori o intervalu u kome mogu da se menjaju parametri sa desne strane ograničenja (Righthand Side – RHS) i koeficijenti uz promenljive u funkciji cilja (OBJ Coefficient), a da dobijeno optimalno rešenje ostane nepromenjeno.

Primer 2.12.7. Potrebno je izvršiti trnsport robe iz dva skladišta (S_1, S_2) do tri prodavnice (P_1, P_2 i P_3). Na skladištima se nalaze 40 i 60, a prodavnicama je potrebno 20, 50 i 30 jedinica robe, respektivno. Troškovi transporta, količina robe u skladištima i prodavnicama dati su u tabeli 2.12.72. Potrebno je problem rešiti korišćenjem LINGO programskog paketa.

Tabela 2.12.72. *Početni podaci za problem transportnog zadatka*

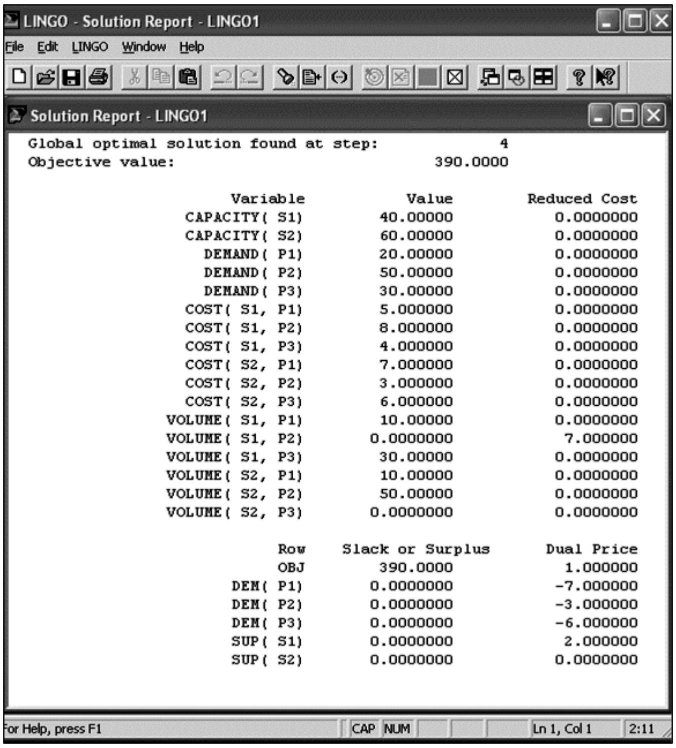
Prodavnice / ***Skladišta***	P_1	P_2	P_3	***Kapacitet***
S_1	5	8	4	40
S_2	7	3	6	60
Potražnja	20	50	30	

Rešenje: Kod LINGO programa je ovde korišćen Help meni u kome se nalaze mnogi rešeni primeri raznih vrsta zadataka. Put komandi je sledeći: File→Open→Sample→Widgets. Otvara se primer rešenog zadatka transportnog problema. Jedino što treba da se uradi je da se zamene postojeći podaci (prekucaju) podacima zadatka koji treba da se reši. Postavka zadatka za rešavanje transportnog zadatka pomoću LINGO programskog paketa, nakon unosa podataka u model je prikazano je na slici 2.12.16., a dobijeno rešenje na slici 2.12.17.




```
SETS:
 SKLADISTE/S1,S2/:CAPACITY;
 PRODAVNICA /P1,P2,P3/:DEMAND;
 ROUTES(SKLADISTE, PRODAVNICA):COST,VOLUME;
ENDSETS

!THE OBJECTIVE;
[OBJ]MIN=@SUM(ROUTES:COST*VOLUME);

!THE DEMAND CONSTRAINTS;
@FOR(PRODAVNICA(J):[DEM]
@SUM(SKLADISTE(I):VOLUME(I,J))>=
DEMAND(J));

!THE SUPPLY CONSTRAINTS;
@FOR(SKLADISTE(I):[SUP]
@SUM(PRODAVNICA(J):VOLUME(I,J))<=
CAPACITY(I));

!HERE ARE THE PARAMETERS;
DATA:
 CAPACITY= 40,60;
 DEMAND= 20,50,30;
 COST= 5,8,4,7,3,6;
ENDDATA
End
```

Slika 2.12.16. *Postavka zadatka u LINGO programu*



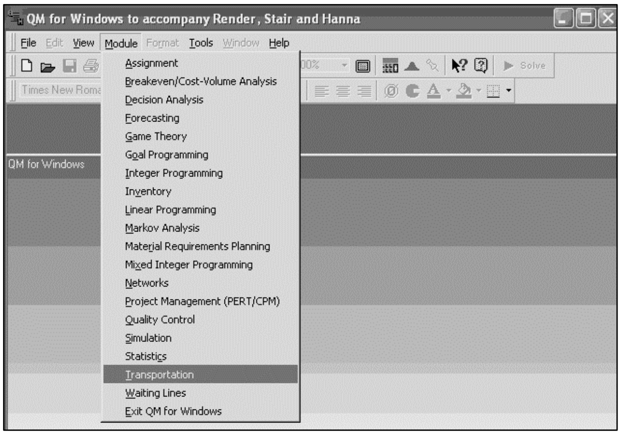

```
Global optimal solution found at step:          4
Objective value:                         390.0000

                 Variable         Value      Reduced Cost
           CAPACITY( S1)       40.00000         0.0000000
           CAPACITY( S2)       60.00000         0.0000000
             DEMAND( P1)       20.00000         0.0000000
             DEMAND( P2)       50.00000         0.0000000
             DEMAND( P3)       30.00000         0.0000000
           COST( S1, P1)       5.000000         0.0000000
           COST( S1, P2)       8.000000         0.0000000
           COST( S1, P3)       4.000000         0.0000000
           COST( S2, P1)       7.000000         0.0000000
           COST( S2, P2)       3.000000         0.0000000
           COST( S2, P3)       6.000000         0.0000000
         VOLUME( S1, P1)       10.00000         0.0000000
         VOLUME( S1, P2)      0.0000000          7.000000
         VOLUME( S1, P3)       30.00000         0.0000000
         VOLUME( S2, P1)       10.00000         0.0000000
         VOLUME( S2, P2)       50.00000         0.0000000
         VOLUME( S2, P3)      0.0000000         0.0000000

                      Row   Slack or Surplus   Dual Price
                      OBJ       390.0000         1.000000
                 DEM( P1)      0.0000000        -7.000000
                 DEM( P2)      0.0000000        -3.000000
                 DEM( P3)      0.0000000        -6.000000
                 SUP( S1)      0.0000000         2.000000
                 SUP( S2)      0.0000000        0.0000000
```

Slika 2.12.17. *Rešenje zadatka dobijeno LINGO programom*

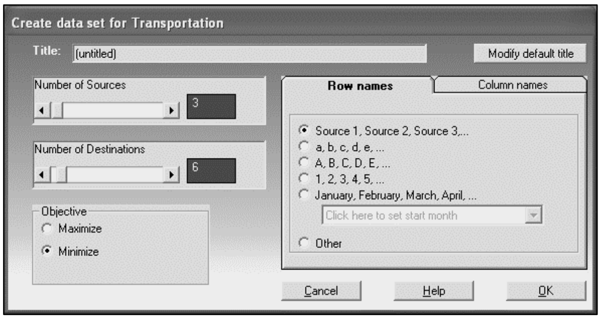
U prvom redu vse idi da je nađeno optimalno rešenje u četvrtom koraku, zatim da je vrednost funkcije cilja 390, za vrednosti promenljivih (čija su imena navedena u koloni Volume (S_i, P_j), a vrednosti u koloni VALUE):

X11 1**0**.000000
X12 **0**.000000
X13 **30**.000000
X21 **10**.000000
X22 **50**.000000
X23 **0**.000000

Primer 2.12.8. Kod programskog paketa QM for Windows ide se na komandu Module, gde se za rešavanje transportnog problema bira opcija “Transportation” (slika 2.12.18. Nakon toga se pokreće program za proračinavanje ovog zadatka. U meniju “File” bira se opcija “New”, nakon čega se pokreće prozor za kreiranje podataka za konkretni transportni zadatak “Create date set for Transportation”, kao što je prikazano na slici 2.12.19.

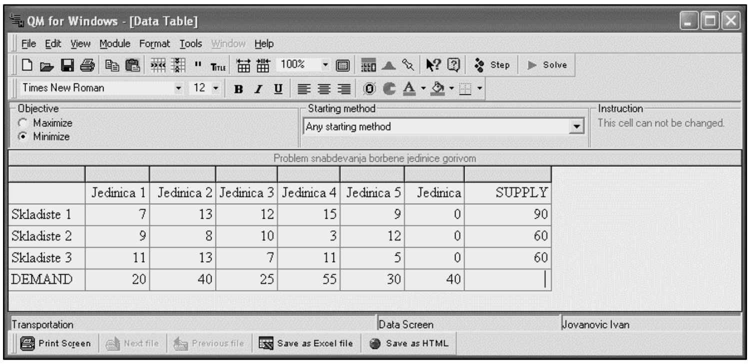


Slika 2.12.18. *Ponuda menija “Module” u QM for Windows programu*



Slika 2.12.19. *Prozor za kreiranje podataka za transportni problem*

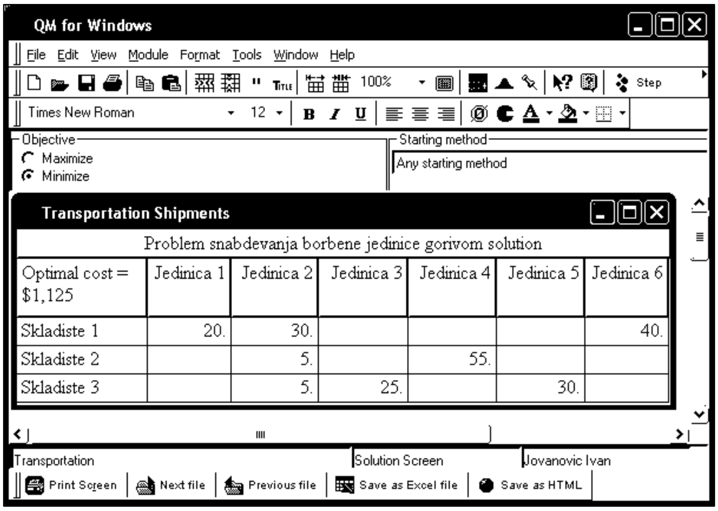
U polju “Number of Sourses” se unosi broj izvora, skladiša, centara otpreme. U primeru koji se ovde rešava, u pitanju su tri skladišta goriva. U polju “Number of Destinations” se upisuje broj ponora, potrošača, prijemnih centara. Konkretno u primeru egzistiraju šet jedinice koje očekuju transportovano gorivo. Kod opcije “Objective” odabiramo opciju “Minimize” jer je funkcija cilja minimalna vrednost transportnih troškova. Na paletama “Row names” i “Column names” ponuđene su opcije za ispisivanje imena redova i kolona ili data mogućnost da sami kreiramo ime birajući opciju “Other”. Nakon potvrde komandom “OK” pojavljuje se tabela za unos podataka za konkretan transportni problem, slika 2.12.20.



Slika 2.12.20. *Tabela za unošenje podataka transportnog problema*

U tabelu se unose podaci za jedinične transportne troškove, kao što je dato u postavci zadatka. U koloni koja je imenovana kao "Jedinica 6" unose se nule, jer je to veštačka kolona koja je posledica transportnog problema otvorenog tipa. U zadatku je dato da su kapaciteti ponude veći od konzumne moći potrošača.

U koloni "Supply" se upisiju količine ponude skladišta, a u redu "Demand" količine potražnje jedinica na terenu. Komandom "Solve" ili tasterom "Enter" daje se nalog za izračunavanje. Rezultat se prikazuje tabelarno, slika 2.12.21.



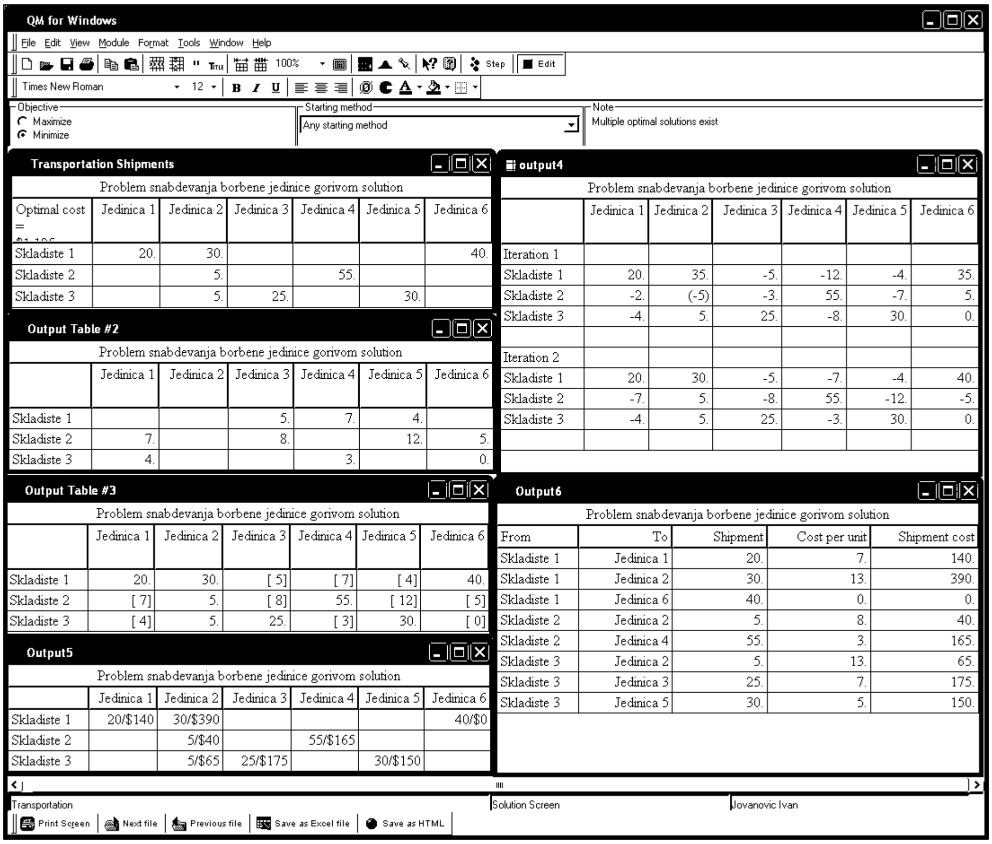
Slika 2.12.21. *Rezultat proračuna programa "QM for Windows"*

Glavna opcija za prikazivanje rezultata "Transportation Shipment" se automatski prikazuje na ekranu. Može se uočiti da je program izbacio konačno rešenje transportnog problema, gde su prikazane količine transportovane robe od tri skladišta do pet (šest) jedinica koje se nalaze na terenu. Takođe se može uočiti, u gornjem levom polju vrednost funkcije cilja, tj. minimalnu cenu transporta, koja je ovde izražena u dolarima (po "default"-u).

U meniju Window nalazi se ukupno šest vrsta izveštaja za ovaj transportni problem, kao što je prikazano na slici 2.12.22. Pored pomenutog izveštaja za transport robe "Transportation Shipment", tu se još nalaze:

Output table #2:	Marginal costs (marginalni – kritični troškovi)
Output table #3:	Final solution table (završna tabela)
Output #4:	Iterations (iteracije)

Output #5: Shipments with costs (troškovi prevoza)
Output #6: Shipping list (spisak otpremljene robe sa cenom)



Transportation Shipments — Problem snabdevanja borbene jedinice gorivom solution

Optimal cost =	Jedinica 1	Jedinica 2	Jedinica 3	Jedinica 4	Jedinica 5	Jedinica 6
Skladiste 1	20.	30.				40.
Skladiste 2		5.		55.		
Skladiste 3		5.	25.		30.	

Output Table #2 — Problem snabdevanja borbene jedinice gorivom solution

	Jedinica 1	Jedinica 2	Jedinica 3	Jedinica 4	Jedinica 5	Jedinica 6
Skladiste 1			5.	7.	4.	
Skladiste 2	7.		8.		12.	5.
Skladiste 3	4.			3.		0.

Output Table #3 — Problem snabdevanja borbene jedinice gorivom solution

	Jedinica 1	Jedinica 2	Jedinica 3	Jedinica 4	Jedinica 5	Jedinica 6
Skladiste 1	20.	30.	[5]	[7]	[4]	40.
Skladiste 2	[7]	5.	[8]	55.	[12]	[5]
Skladiste 3	[4]	5.	25.	[3]	30.	[0]

Output5 — Problem snabdevanja borbene jedinice gorivom solution

	Jedinica 1	Jedinica 2	Jedinica 3	Jedinica 4	Jedinica 5	Jedinica 6
Skladiste 1	20/$140	30/$390				40/$0
Skladiste 2		5/$40		55/$165		
Skladiste 3		5/$65	25/$175		30/$150	

output4 — Problem snabdevanja borbene jedinice gorivom solution

	Jedinica 1	Jedinica 2	Jedinica 3	Jedinica 4	Jedinica 5	Jedinica 6
Iteration 1						
Skladiste 1	20.	35.	-5.	-12.	-4.	35.
Skladiste 2	-2.	(-5)	-3.	55.	-7.	5.
Skladiste 3	-4.	5.	25.	-8.	30.	0.
Iteration 2						
Skladiste 1	20.	30.	-5.	-7.	-4.	40.
Skladiste 2	-7.	5.	-8.	55.	-12.	-5.
Skladiste 3	-4.	5.	25.	-3.	30.	0.

Output6 — Problem snabdevanja borbene jedinice gorivom solution

From	To	Shipment	Cost per unit	Shipment cost
Skladiste 1	Jedinica 1	20.	7.	140.
Skladiste 1	Jedinica 2	30.	13.	390.
Skladiste 1	Jedinica 6	40.	0.	0.
Skladiste 2	Jedinica 2	5.	8.	40.
Skladiste 2	Jedinica 4	55.	3.	165.
Skladiste 3	Jedinica 2	5.	13.	65.
Skladiste 3	Jedinica 3	25.	7.	175.
Skladiste 3	Jedinica 5	30.	5.	150.

Slika 2.12.22. *Kompletan izveštaj iz menija Window*

2.12.3.4. Otvoreni model transportnog problema

Otvoreni problemi transporta nastaju kada nije ispunjen uslov da je:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \qquad (2.12.21.)$$

U takvim slučajevima neophodno je uvesti fiktivno ishodište, tj. fiktivno odredište.

a) Ukoliko je:

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \qquad (2.12.22.)$$

uvodi se fiktivno orediste B_{n+1} sa zahtevanom količinom:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \tag{2.12.23.}$$

Troškovi transporta iz ishodišta u fiktivno odredište su nula, tj:

$$c_{i\,n+1} = 0;\ 1 \le i \le m \tag{2.12.24.}$$

b) Ukoliko je:

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \tag{2.12.25.}$$

uvodi se fiktivno ishodiste A_{m+1} sa ponudom od:

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i \tag{2.12.26.}$$

Troskovi transporta od fiktivnog ishodišta do bilo kog odredišta jednaki su nuli, tj:

$$C_{m+1\,j} = 0;\ 1 \le j \le n \tag{2.12.27.}$$

Nakon svođenja početnog problema na odnos $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, za rešavanje se može primeniti neki od objašnjenih postupaka.

Primer 2.12.9. Prilikom izvršenja jednog taktičkog zadatka za snabdevanje gorivom mogu se koristiti tri izvora za snabdevanje sa količinama: 90, 60 i 60 tona. Količine goriva koje jedinice zahtevaju su: 20, 40, 25, 55 i 30 tona, respektivno. Odrediti plan snabdevanja gorivom vodeći računa da troškovi snabdevanja budu minimalni. Jedinični troškovi dati su u tabeli 2.12.73. Odrediti ukupne troškove pri izvršenju dobijenog plana snabdevanja.

Tabela 2.12.73. *Početni podaci za transportni problem*

OS \ PS	PS	B_1	B_2	B_3	B_4	B_5
OS	$210 \neq 170$	**20**	**40**	**25**	**55**	**30**
A_1	**90**	**7**	**13**	**12**	**15**	**9**
A_2	**60**	**9**	**8**	**10**	**3**	**12**
A_3	**60**	**11**	**13**	**7**	**11**	**5**

Rešenje. Ukupna količina goriva kojom raspolažu izvori snabdevanja iznosi 210 tona, a jedinice potražuju 170 tona. Pošto je količina goriva u izvorima snabdevanja veća od količine koju jedinice traže neophodno je formirati fiktivno olrediste, odnosno u matrici troškova uvesti fiktivnu kolonu sa transportnim troškovima jednakim nuli. Količina goriva koja se dodeljuje fiktivnom odredištu jednaka je:

$$B_6 = \sum_{i=1}^{3} a_i - \sum_{j=1}^{5} b_j = 210 - 170 = 40\ t$$

Početno rešenje biće određeno pomoću Vogelovog aproksimativnog metoda, kao što je prikazano u tabeli 2.12.74.

Tabela 2.12.74. *Početno rešenje po Vogel-ovog aproksimativnog metoda*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	***B*₆**	***Iteracije***
OS	210=210	**20**	**40**	**25**	**55**	**30**	**40**	**I II III IV V VI**
***A*₁**	**90**	7 / 20	13 / 35	12	15	9	0 / 35	7 7 7 2 2 6
***A*₂**	**60**	9	8	10	3 / 55	12	0 / 5	3 8 - - - -
***A*₃**	**60**	11	13 / 5	7 / 25	11	5 / 30	0	5 5 5 2 6 2
Iteracije I		2	5	3	8	4	0	
II		2	5	3	-	4	0	
III		4	0	5	-	4	0	
IV		4	0	5	-	4	-	
V		4	0	-	-	4	-	
VI		4	0	-	-	-	-	

Troškovi transporta za početno rešenje iznose 1150 novčanih jedinica. Poboljšanje početnog rešenja biće izvršeno pomoću metoda potencijala. Vrednosti potencijala $\boldsymbol{u_i}$ i $\boldsymbol{v_j}$ za dato početno rešenje date su u tabeli 2.12.75.

Tabela 2.12.75. *Vrednosti potencijala u prvoj iteraciji*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	***B*₆**	v_j
OS	210=210	**20**	**40**	**25**	**55**	**30**	**40**	
***A*₁**	**90**	7 / 20	13 / -θ 35	7	3	5	0 / +θ 35	**0**
***A*₂**	**60**	7	13 / +θ +5	7	3 / 55	5	0 / -θ 5	**0**
***A*₃**	**60**	7	13 / 5	7 / 25	3	5 / 30	0	**0**
u_i		**7**	**13**	**7**	**3**	**5**	**0**	

Uslov za ulazak u bazu ispunjava polje (a_2,b_2) i u njega se dovodi kolicina θ=5 tona goriva. Rešenje dobijeno nakon izvršene preraspodele goriva po temenima mnogougaonika dato je u tabeli 2.12.76.

Tabela 2.12.76. *Rešenje nakon prve iteracije*

	PS	***B*₁**	***B*₂**	***B*₃**	***B*₄**	***B*₅**	***B*₆**	v_j
OS	210=210	**20**	**40**	**25**	**55**	**30**	**40**	
***A*₁**	**90**	7 / 20	13 / 30	7	8	5	0 / 40	**0**
***A*₂**	**60**	2	8 / 5	2	3 / 55	0	-5	**-5**
***A*₃**	**60**	7	13 / 5	7 / 25	8	5 / 30	0	**0**
u_i		**7**	**13**	**7**	**8**	**5**	**0**	

Kako nijedno polje ne zadovoljava uslov za ulazak u bazu to je dobijeno rešenje optimalno. Ukupni troškovi transporta iznose: $F(x)$= 1.125 novčanih jedinica. Ovaj primer je urađen i korišćenjem programa "QM for Windows" u poglavlju 2.12.3.3.

2.12.3.5. Degeneracija u transportnom problemu

Transportni problem sadrži $m+n-1$ linearno nezavisnih jednačina koje čine bazu vektorskog prostora, pa na osnovu toga svako nedegenerisano bazično moguće rešenje ima tačno $m+n-1$ pozitivnih promenljivih x_{ij}. Svako bazično rešenje koje ima manje od $m+n-1$ pozitivnih promenljivih x_{ij} naziva se ***degenerisano*** rešenje. Do degeneracije dolazi kada istovremeno budu podmireni i ishodište i odredište. Pojava degeneracije onemogućava primenu postupka traženja optimalnog rešenja. Degeneracija je relativno česta pojava u transportnom problemu, ali se rešava na lak način. U slučaju degeneracije, zbog manjeg broja pozitivnih promenljivih x_{ij}, ne može se neposredno koristiti modifikovani metod. Zapravo, na osnovu pozitivnih promenljivih x_{ij} određujemo dualne promenljive u_i i v_j. Ukoliko se ne formira potreban broj jednačina tipa (2.12.16.), otkazuje modifikovani metod, i ne mogu se izračunati sve dualne promenljive. Potrebno je da se izvorni podaci o količinama u ishodištu i odredištu modifikuju na sledeći način:

$$\begin{aligned} a_i &= a_i + \varepsilon, \ \forall i \\ b_j &= b_j \quad \text{za } j = 1, 2, ..., n-1 \qquad (2.12.28.) \\ b_j &= b_j + m\,\varepsilon \ \text{ za } j = n \end{aligned}$$

Primer 2.12.10. Dat je problem transporta robe iz tri skladišta (A_1, A_2, A_3) na četiri korisničke lokacije (B_1, B_2, B_3, B_4,). Potrebno je odrediti optimalan plan transporta uz minimalne ukupne transportne troškove. Neophodni podaci za izradu plana dati su u tabeli 2.12.77.

Tabela 2.12.77. *Početni podaci transportnog problema*

	PS	***B₁***	***B₂***	***B₃***	***B₄***
OS	45=45	**5**	**15**	**15**	**10**
A₁	**15**	**10**	**5**	**20**	**11**
A₂	**25**	**12**	**7**	**9**	**20**
A₃	**5**	**5**	**14**	**16**	**18**

Rešenje: Početno rešenje dobijeno metodom dvojnog prvenstva, prikazano je u tabeli 2.12.78. Početno rešenje je degenerisano jer vektor rešenja ima četiri pozitivne komponente, umesto potrebnih $(n+m-1)$=6. Određivanje početnog rešenja biće ponovljeno sa prethodno modifikovanim količinama a_i i b_j. Ceo postupak sa rešenjem dat je u tabeli 2.12.79.

***Tabela 2.12.78.** Početno rešenje metodom dvojnog prvenstva*

	PS	*B_1*	*B_2*	*B_3*	*B_4*
OS	45=45	**5**	**15**	**15**	**10**
A_1	**15**	**10**	**5 ** 15**	**20**	**11 ***
A_2	**25**	**12**	**7 ***	**9 * 15**	**20 10**
A_3	**5**	**5 ** 5**	**14**	**16**	**18**

***Tabela 2.12.79.** Početno rešenje sa modifikovanim vrednostima a_i i b_j*

	PS	*B_1*	*B_2*	*B_3*	*B_4*
OS	45=45	**5**	**15**	**15**	**10+ 3ε**
A_1	**15 + ε**	**10**	**5 ** 15**	**20**	**11 * + ε**
A_2	**25 + ε**	**12**	**7 ***	**9 * 15**	**20 10+ ε**
A_3	**5 + ε**	**5 ** 5**	**14**	**16**	**18 + ε**

Rešenje dobijeno na ovaj način nije degenerisano i omogućava primenu postupka optimizacije, što će biti i urađeno. Poboljšanje početnog rešenja biće izvršeno metodom potencijala. Vrednosti potencijala za početno rešenje kao i postupak preraspodele robe po temenima mnogougaonika dati su u tabeli 2.12.80. Poboljšano rešenje je dato u tabeli 2.12.81.

***Tabela 2.12.80.** Prvo poboljšanje početnog rešenja*

	PS	*B_1*	*B_2*	*B_3*	*B_4*	**u_i**
OS	45=45	**5**	**15**	**15**	**10+ 3ε**	
A_1	**15 + ε**	**-2**	**5 -θ 15**	**0**	**11 +θ + ε**	**0**
A_2	**25 + ε**	**7**	**14 +θ +7**	**9 15**	**20 -θ 10+ ε**	**9**
A_3	**5 + ε**	**5 5**	**12**	**7**	**18 + ε**	**7**
v_j		**-2**	**5**	**0**	**11**	

***Tabela 2.12. 81.** Rešenje nakon prvog poboljšanja*

	PS	*B_1*	*B_2*	*B_3*	*B_4*	**u_i**
OS	45=45	**5**	**15**	**15**	**10+ 3ε**	
A_1	**15 + ε**	**-2**	**5 5- ε**	**7**	**11 10+ 2ε**	**0**
A_2	**25 + ε**	**0**	**7 10+ ε**	**9 15**	**13**	**2**
A_3	**5 + ε**	**5 5**	**12**	**14**	**18 + ε**	**7**
v_j		**-2**	**5**	**7**	**11**	

Nakon sprovedenog postupka utvrđuje se da je dobijeno rešenje optimalno. Veličine *ε* ne smetaju, jer se u optimalnom rešenju mogu zanemariti. Ukupni troškovi transporta iznose:

$$F(x) = 5 \cdot 5 + 5 \cdot 5 + 10 \cdot 7 + 15 \cdot 9 + 10 \cdot 11 = 365 \text{ novčanih jedinica.}$$

Primer 2.12.11. Iz tri stovarišta (S_1, S_2, S_3) je potrebno snabdeti četiri potrošača (P_1, P_2, P_3, P_4) robom *A*. Raspoložive količine svakog stovarišta, potrebe potrošača, kao i transportni troškovi jedinice robe *A*, dati su u tabeli 2.12.82.

Iz tabele se lako vidi da su potrebe potrošača (20+20+20+20=80) veće od raspoloživih količina koje tri stovarišta mogu isporučiti (20+30+25=75). Prema tome, radi se o otvorenom transportnom problemu. Uvodi se još jedan red u tabelu, novo stovarište sa ponudom od 5 jedinica robe *A* (80-75 =5), čime ose bezbeđuje jednakost između ponude stovarišta i potreba potrošača. Troškovi transporta jedinice robe *A* od novog stovarišta do svih potrošača jednaki su nuli.

Posle ovog proširenja, problem je lako rešiti. Početno rešenje, pronađeno pomoću Vogel-ove aproksimativnog metoda, prikazano je u tabeli 2.12.83.

Tabela 2.12.82. *Početni podaci transportnog problema*

Stovarište	***Potrošači***				***Ponuda***
	P_1	***P_2***	***P_3***	***P_4***	
S_1	7	8	4	8	**20**
S_2	5	3	5	6	**30**
S_3	9	5	7	9	**25**
Potrebe	**20**	**20**	**20**	**20**	

Tabela 2.12.83. *Početno rešenje dobijeno Vogel-ovim aproksimativnim metodom*

OS \ ***PS***		***P_1***	***P_2***	***P_3***	***P_4***	***Iteracije***
		20	**20**	**20**	**20**	**I II III IV V**
S_1	**20**	**7**	**8**	**4** **20**	**8**	**3 3 - - -**
S_2	**30**	**5** **20**	**3** **10**	**5**	**6**	**2 2 2 3 -**
S_3	**25**	**9**	**5** **10**	**7**	**9** **15**	**2 2 2 2 2**
S_4	**5**	**0**	**0**	**0**	**0** **5**	**0 - - - -**
Iteracije		**5**	**3**	**4**	**6**	
I		**2**	**2**	**1**	**2**	
II		**4**	**2**	**-**	**3**	
III		**-**	**2**	**-**	**3**	
IV		**-**	**5**	**-**	**9**	
V						

Rešenje je degenerisano. Ima ukupno šest pozitivnih promenljivih x_{ij}, a nedegenerisano rešenje mora imati m+n-1= 4+4-1= 7 promenljivih većih od nule. Za primenu modifikovanog metoda, tj. za određivanje dualnih promenljivih u_i i v_j, mora da se ima još jedna pozitivna promenljivu. Koju vrednost može dobiti nova promenljiva i na koje mesto u tabeli je treba upisati? Novoj promenljivoj određuje se vrednost ε. Ova vrednost promenljive treba da omogući postizanje sledeća dva cilja:

a) vrednost ε je pozitivna i na osnovu toga možemo formirati potrebnu jednačinu za izračunavanje dualnih promenljivih i
b) vrednost ε je vrlo mala, tako da se ona može zanemariti u bazičnom rešenju.

Nešto je teže odabrati koja promenljiva će dobiti vrednost ε. Najbolje je taj izbor izvršiti prilikom određivanja vrednosti dualnih promenljivih u_i i v_j. U tabeli 2.12.84. je to prikazano.

Određeno je da je $u_1 = 0$, pa je, pomoću promenljive $x_{13} > 0$, dobijeno da je $v_3 = 4$. U prvom redu nema više pozitivnih promenljivih, pa bi postupak trebalo nastaviti preko treće kolone i promenljive v_3. Međutim, i treća kolona nema drugih pozitivnih promenljivih, pa se ovaj postupak ne može nastaviti. To je zbog toga što je rešenje degenerisano. Sada se mora izabrati promenljiva kojoj će se dati vrednost ε. Pošto su određene samo dualne promenljive u_i i v_j, to mora biti promenljiva iz prvog reda ili treće kolone. Može se bilo kojoj promenljivoj iz prvog reda i treće kolone dodeliti vrednost ε. Međutim, bolje je sada uzeti u obzir i dodatni kriterijum. Tako je određeno da to bude promenljiva sa najmanjim koeficijentom c_{ij}. To je $c_{43} = 0$, pa je u tabeli 2.12.84. promenljiva $x_{43} = \varepsilon$. Ona se u daljem postupku rešavanja problema tretira kao i ostale pozitivne promenljive, pa nije teško odrediti i ostale dualne promenljive u_i i v_j, a na osnovu njih i vrednost promenljivih Δ_{ij} ($\Delta_{ij} = c_{ij}-u_i-v_j$). U tabeli 2.12.83. upisane su samo negativne vrednosti promenljivih Δ_{ij}.

Tabela 2.12.84. *Određivanje potencijala u_i i v_j i dodeljivanje vrednosti ε*

OS \ *PS*	***PS***	P_1	P_2	P_3	P_4	u_i
OS		**20**	**20**	**20**	**20**	
S_1	**20**	7	8	4 20	8	**0**
S_2	**30**	5 20	3 10	5 -2	6 -1	**3**
S_3	**25**	9	5 10	7 +θ -2	9 -θ 15	**5**
S_4	**5**	0	0	0 -θ +ε	0 +θ 5	**-4**
v_j		**2**	**0**	**4**	**4**	

Napomena. Zbog promenljive $x_{43} = \varepsilon$, trebalo bi povećati vrednost slobodnih članova S_4 i P_3 za ε. To se ne čini jer je napomenuto da se u proračunima ε zanemaruje.

Rešenje iz tabele 2.12.84. nije optimalno. Pošto je $\Delta_{23} = \Delta_{33} = -2$, jedna od promenljivih x_{23} ili x_{33} u narednoj iteraciji dobija pozitivnu vrednost. Proizvoljno je uzeto da to bude x_{33}. U tabeli 2.12.84. oznakama (+θ) i (-θ) označene su potrebne promene u postojećem rešenju. Zapaža se da od promenljivih, označenih sa (-θ), najmanju vrednost ima $x_{43} = \varepsilon$. To znači da će promenljiva x_{33} u narednoj iteraciji imati vrednost ε . Ovom promenom neće se menjati vrednost ostalih promenljivih jer, i prilikom oduzimanja ε od neke vrednosti i prilikom dodavanja ε nekoj promenljivoj, ε se zanemaruje. Došlo je samo do seljenja vrednosti ε. Zbog toga će se promeniti vrednosti dualnih promenljivih. Sve ove promene date su u tabeli 2.12.85.

Tabela 2.12.85. *Rešenje nakon prvog poboljšanja - prve iteracije*

OS \ *PS*	***PS***	P_1	P_2	P_3	P_4	u_i
OS		**20**	**20**	**20**	**20**	
S_1	**20**	4	2	4 20	6	**0**
S_2	**30**	5 20	3 -θ 10	5	7 +θ -1	**1**
S_3	**25**	7	5 +θ 10	7 +ε	9 -θ 15-ε	**3**
S_4	**5**	-2	-4	-2	0 5+ε	**-6**
v_j		**4**	**2**	**4**	**6**	

U tabeli 2.12.85. je promenljiva $\Delta_{24} = -1$, što znači da nije pronađeno optimalno rešenje. Promenljiva x_{24} u narednoj iteraciji dobija pozitivnu vrednost. Novo bazično rešenje nalazi se u tabeli 2.12.86.

U tabeli 2.12.86. nema negativnih promenljivih Δ_{ij}, pa je pronađeno optimalno rešenje. Ono je degenerisano i čine ga sledeće promenljive: $x_{13}=20$, $x_{21}=20$, $x_{24}=10$, $x_{32}=20$, $x_{34}=5$, $x_{45}= 5$, a minimalna vrednost funkcije kriterijuma iznosi 345 n.j.

Pošto se radi o otvorenom transportnom problemu, potrebno je objasniti ulogu promenljive x_{44}. Četvrto stovarište je nepostojeće, pa potrošač koji deo svojih potreba podmiruje odatle, u stvari, neće podmiriti svoje potrebe u potpunosti. Kako je $x_{44} = 5$, pa četvrti potrošač neće dobiti ovih 5 jedinica robe *A*.

Tabela 2.12.86. *Konačno rešenje nakon druge iteracije*

OS \ PS		P_1	P_2	P_3	P_4	u_i
		20	20	20	20	
S_1	20	5	2	4 / 20	6	0
S_2	30	5 / 20	3	4	6 / 10	0
S_3	25	7	5 / 20	7 / $+\varepsilon$	9 / $5-\varepsilon$	3
S_4	5	-1	-4	-2	0 / $5+\varepsilon$	-6
v_j		5	2	4	6	

2.12.3.6. Maksimalna vrednost funkcije kriterijuma

Transportni problem je formulisan kao problem minimuma, odnosno kao problem u kome se traži rešenje koje će obezbediti da funkcija kriterijuma uzme minimalnu vrednost. Funkcija kriterijuma u transportnom problemu najčešće označava ukupne transportne troškove, pa je bilo logično tražiti njenu minimalnu vrednost. Moguće je, medutim, formulisati transportni problem u kome će se tražiti takvo rešenje koje će obezbediti da funkcija kriterijuma dostigne svoju maksimalnu vrednost. U tim zadacima ekonomsko značenje koeficijenata c_{ij} iz funkcije kriterijuma je takvo da je logično tražiti maksimalnu vrednost funkcije.

Nema velike razlike u postrupku rešavanja problema minimuma, odnosno maksimuma. Matematički modeli su isti. Razlike postoje pri pronalaženju početnog rešenja i u kriterijumu za ocenu optimalnosti, odnosno izbora promenljive koja će ući u naredno rešenje.

Kod pronalaženja početnog rešenja:

- Kod **dijagonalnog metoda** ne menja se ništa u postupku pronalaženja početnog rešenja. Uostalom, taj metod i ne vodi računa o kriterijumu optimalnosti, pa će početno rešenje biti isto, bez obzira koju vrednost funkcije tražimo.
- Menja se kriterijum za pronalaženje početnog rešenja po **metodu jediničnih koeficijenata**. Kod traženja minimalne vrednosti funkcije kriterijuma, ovaj metod polazi od *najmanjih koeficijenata* c_{ij} u matrici troškova. Pošto je izmenjeno ekonomsko značenje koeficijenata c_{ij}, menja se i ovaj kriterijum. Sada se početno rešenje pronalazi polazeći od *najvećih koeficijenata* c_{ij}, pa se prednost u transportu daje tim relacijama.
- Kod **Vogel-ovog aproksimativnog metoda** ima razlike u postupku pronalaženja početnog rešenja. Za probleme u kojima se traži minimalna vrednost funkcije računaju se razlike između *dva najmanja koeficijenta* c_{ij}, pa se prednost daje redu ili koloni kojima odgovara najveća razlika. U problemima u kojima se traži maksimalna vrednost funkcije kriterijuma traže se najveće

razlike između *dva najveća koeficijenta* c_{ij}, pa se prednost daje redu ili koloni sa najvećom razlikom.

Nema nikakvih promena u postupku određivanja **optimalnog rešenja** za probleme maksimuma u odnosu na postupak koji se koristi kod problema minimuma. Dualna promenljiva Δ_{ij}, koju određujemo pomoću relacije:

$$\Delta_{ij} = c_{ij} - (u_i + v_j) \qquad (2.12.29.)$$

i ovde ima dvojaku ulogu: služi kao kriterijum za ocenu optimalnosti pronađenog rešenja i kao kriterijum za izbor promenljive koja ulazi u naredno bazično rešenje. Za probleme u kojima se traži maksimalna vrednost funkcije kriterijuma pronađeno je optimalno rešenje samo ako su sve dualne promenljive $\Delta_{ij} \leq 0$. Takođe, u naredno rešenje ulazi promenljiva x_{ij} kojoj odgovara najveća pozitivna vrednost dualne promenljive Δ_{ij}.

Primer 2.12.12. Četiri fabrike (F_1, F_2, F_3 i F_4) proizvode robu A i mogu isporučiti potrošačima sledeće količine: F_1 90, F_2 90, F_3 100 i F_4 80 jedinica. Potrošačima su potrebne sledeće količine robe A: potrošaču P_1 120, P_2 40, P_3 60 i potrošaču P_4 140 jedinica. Ukoliko se plaćanje robe izvrši odmah, proizvođači su potrošačima ponudili određene popuste u ceni, što se može videti iz tabele 2.12.87. Potrošači su prihvatili da robu plate odmah i žele da sami sačine plan snabdevanja robom A, koji će im obezbediti maksimalni ukupni popust u ceni.

Tabela 2.12.87. *Procenti popusta u ceni*

	P_1	P_2	P_3	P_4
F_1	9	12	13	18
F_2	6	10	8	14
F_3	12	10	15	16
F_4	10	15	12	14

Rešenje: Početno rešenje prikazano u tabeli 2.12.88., pronađeno je pomoću metoda najvećeg koeficijenta c_{ij} u transportnoj tabeli. Najveći koeficijent u tabeli je $c_{14} = 18$, pa je vrednost promenljive $x_{14} = 90$. Isporučena je celokupna ponuda fabrike F_1, pa se iz daljeg razmatranja isključuje prvi red. U preostalom delu tabele najveći koeficijent je $c_{34} = 16$, a vrednost promenljive je $x_{34} = 50$. Podmirene su ukupne potrebe potrošača P_4. Isključuje se četvrta kolona iz daljeg razmatranja i nastavlja se sa određivanjem vrednosti preostalih promenljivih. To su najpre, za koeficijente $c_{33} = c_{42} = 15$, promenljive $x_{33} = 50$ i $x_{42} = 40$. Sada iz razmatranja treba isključiti treći red i drugu kolonu. U preostalom delu tabele najveći koeficijent je $c_{43} = 12$, pa je vrednost promenljive $x_{43} = 10$. Pošto se iz razmatranja isključi i treća kolona, ostaju neizmirene samo potrebe potrošača P_1, pa je lako odrediti vrednost preostalih promenljivih: $x_{21} = 90$ i $x_{41} = 30$.

Tabela 2.12.88. *Početno rešenje dobijeno metodom najvećih koeficijenata*

OS \ PS	360=360	P_1 120	P_2 40	P_3 60	P_4 140
F_1	90	9	12	13	18 / 90
F_2	90	6 / 90	10	8	14
F_3	100	12	10	15 / 50	16 / 50
F_4	80	10 / 30	15 / 40	12 / 10	14

Iznos funkcije kriterijuma, za bazno rešenje, dobijeno metodom najvećih koeficijenata, je:

$$\max F(x) = 4.730 \text{ n.j.}$$

Rešenje je nedegenerisano, pa je lako, pomoću relacije:

$$c_{ij} = u_i + v_j \qquad (2.12.30.)$$

izračunati dualne promenljive u_i i v_j. Vrednosti ovih promenljivih upisane su u poslednji red i poslednju kolonu tabele 2.12.89. Određena je i vrednost dualnih promenljivih Δ_{ij}. Obzirom na kriterijum optimalnosti, pozitivan uticaj na vrednost funkcije kriterijuma imaju samo promenljive $\Delta_{ij}>0$, pa su u tabeli 2.12.89. upisane pozitivne vrednosti za $\Delta_{24} = 5$ i $\Delta_{44} = 1$.

Tabela 2.12.89. *Početno rešenje dobijeno metodom najvećih koeficijenata*

	PS	**P_1**	**P_2**	**P_3**	**P_4**	**u_i**
OS	**360=360**	**120**	**40**	**60**	**140**	
F_1	**90**	9	12	13	18 / 90	0
F_2	**90**	6 -θ / 90	10	8	14 +θ / +5	-9
F_3	**100**	12	10	15 +θ / 50	16 -θ / 50	-2
F_4	**80**	10 +θ / 30	15 / 40	12 -θ / 10	14 / +1	-5
v_j		**15**	**20**	**17**	**18**	

Bazično rešenje iz tabele 2.12.89. nije optimalno. Najveću pozitivnu vrednost ima dualna promenljiva $\Delta_{24} = 5$, pa će promenljiva x_{24} u narednoj iteraciji imati pozitivnu vrednost 10. Potrebne promene rešenja naznačene su, takođe, u tabeli 2.12.89. Novo bazično rešenje nalazi se u tabeli 2.12.90.

Tabela 2.12.90. *Bazično rešenje nakon prve iteracije*

	PS	**P_1**	**P_2**	**P_3**	**P_4**	**u_i**
OS	**360=360**	**120**	**40**	**60**	**140**	
F_1	**90**	9	12	13	18 / 90	0
F_2	**90**	6 -θ / 80	10	8	14 +θ / 10	-4
F_3	**100**	12 +θ / +4	10	15 / 60	16 -θ / 40	-2
F_4	**80**	10 / 40	15 / 40	12	14	0
v_j		**10**	**15**	**17**	**18**	

Funkcija kriterijuma za ovo rešenje ima vrednost 4.780 n.j. I ovo rešenje nije optimalno zato što postoji promenljiva $\Delta_{31} = 4 > 0$. U naredno rešenje ulazi promenljiva x_{31} i dobija vrednost 40. Potrebne promene rešenja naznačene su, takođe, u tabeli 2.12.90. Novo bazično rešenje pronađeno je u tabeli 2.12.91.

Bazično rešenje iz tabele 2.12.91. je optimalno rešenje i čine ga promenljive: x_{14}=90, x_{21}=40, x_{24}=50, x_{31}=40, x_{33}=60, x_{41}= 40, x_{42}=40. Vrednost funkcije kriterijuma, koja označava najveći mogući ukupni popust u ceni, iznosi: $\max F(x) = 4940$ n.j. Pošto je promenljiva Δ_{13}=0, postoji još jedno optimalno rešenje, sa istom vrednošću funkcije kriterijuma, koje se dobija III iteracijom (tabela 2.12.92.) i koje je prikazano u tabeli 2.12.93.

Tabela 2.12.91. *Bazično rešenje nakon druge iteracije – prvo optimalno rešenje*

	PS	P_1	P_2	P_3	P_4	u_i
OS	360=360	120	40	60	140	
F_1	90	9	12	13	18 90	0
F_2	90	6 40	10	8	14 50	-4
F_3	100	12 40	10	15 60	16	2
F_4	80	10 40	15 40	12	14	0
v_j		10	15	13	18	

Tabela 2.12.92. *Treća iteracija – nakon prvog optimalnog rešenja*

	PS	P_1	P_2	P_3	P_4	u_i
OS	360=360	120	40	60	140	
F_1	90	9	12	13 +θ	18 -θ 90	0
F_2	90	6 -θ 40	10	8	14 +θ 50	-4
F_3	100	12 +θ 40	10	15 -θ 60	16	2
F_4	80	10 40	15 40	12	14	0
v_j		10	15	13	18	

Tabela 2.12.93. *Bazično rešenje nakon treće iteracije – drugo optimalno rešenje*

	PS	P_1	P_2	P_3	P_4	u_i
OS	360=360	120	40	60	140	
F_1	90	9	12	13 40	18 50	0
F_2	90	6	10	8	14 90	-4
F_3	100	12 80	10	15 20	16	2
F_4	80	10 40	15 40	12	14	0
v_j		10	15	13	18	

2.12.4. Metod raspoređivanja

Problemi raspoređivanja ili, kako se često nazivaju problemi asignacije, predstavljaju specijalan slučaj transportnog problema. Suština problema je u tome da se na optimalan način rasporedi n ljudi za obavljanje n poslova, tako da funkcija kriterijuma dostigne optimalnu vrednost.

Pri tome se polazi od sledećeg:

a) jedan posao može biti dodeljen samo jednom čoveku i jedan čovek može primiti samo jedan posao,
b) poznata je efikasnost *i*-tog izvršioca na *j*-toj aktivnosti (svi izvršioci nisu podjednako efikasni za obavljanje pojedinih poslova).

Efikasnost radnika za obavljanje pojedinih poslova može se meriti na različite načine. Ona može biti iskazana vremenom (satima) potrebnim svakom radniku za izvršenje svih poslova, pa će optimalna raspodela ljudi na poslove podrazumevati izvršenje svih poslova za najkraće moguće vreme.

Efikasnost može biti iskazana i količinom proizvodnje svakog radnika na pojedinim poslovima, pa će optimalni raspored podrazumevati izvršenje svih poslova uz maksimalnu proizvedenu količinu.

U probleme koji se mogu rešavati metodom raspoređivanja dolazi i raspored mašina za obavljanje pojedinih poslova, prijem kandidata na konkursu, najkraći ukupni putevi, najniži ukupni troškovi, najveći dohodak i drugo.

Kako se radi o specijalnom slučaju transportnog problema, problemi raspoređivanja se mogu rešavati i transportnim metodama i simpleks metodom. Međutim, struktura ovih problema i neke njihove karakteristike omogućile su da se formuliše poseban algoritam za njihovo rešavanje. H. W. Kuhn je 1955. razvio jedan postupak za rešavanje problema raspoređivanja i nazvao ga *„mađarskim metodom“*. Ovaj metod za rešavanje problema raspoređivanja koristi njegov dualni problem.

Formiranje matematičkog modela i izlaganje mađarskog metoda ilustrovano je na primeru.

2.12.4.1. Opšti model

Opšti model asignacije biće objašnjen kroz primer gde pet radnika treba da izvrše pet različitih poslova. Svaki od pet radnika zna da izvrši sve poslove, ali istovremeno može da radi samo jedan posao. Različita je individualna efikasnost ovih radnika za obavljanje pojedinih poslova i ona je izražena u satima rada, a data je u tabeli 2.12.94. Raspodelu poslova na radnike treba izvršiti tako da ukupno vreme izvršenja svih poslova bude najkraće.

Tabela 2.12.94. *Efikasnost radnika za obavljanje pojedinih poslova*

Radnici	***Poslovi***				
	P_1	P_2	P_3	P_4	P_5
R_1	14	9	12	8	16
R_2	8	7	9	9	14
R_3	9	11	10	10	12
R_4	10	8	8	6	14
R_5	11	9	10	7	13

Nije poznato koji će posao raditi svaki radnik, pa će se označiti sa:

x_{ij} – i-ti radnik koji će raditi j-ti posao,
c_{ij} – vreme potrebno i-tom radniku za obavljanje j-*tog* posla.

Pošto jedan radnik može raditi samo jedan posao, to mora biti:

x_{ij} = 1 ako i-ti radnik dobije j-ti posao,
x_{ij} = 0 u protivnom slučaju (ako i-ti radnik ne dobije posao).

Usvaja se da, u opštem slučaju, ima n radnika (i = 1,... , n) i n poslova (j = 1,2,... ,n). Sada se može formirati opšti model problema raspoređivanja.

Potrebno je pronaći minimalnu vrednost funkcije kriterijuma, kao što je prikazano formulom (2.12.31.):

$$F(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} \qquad (2.12.31.)$$

uz sledeća ograničenja:

– uslov da jedan radnik može da obavlja samo jedan posao, pa zbir u svakom redu može biti samo 1, kao što je prikazano formulom (2.12.32.).

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,3,...,n \qquad (2.12.32.)$$

– uslov da jedan posao može biti dodeljen jednom radniku, pa zbir u svakoj koloni, takođe, može biti samo 1, kao što je prikazano formulom (2.12.33.).

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,...,n. \qquad (2.12.33.)$$

– Promenljive mogu uzimati samo vrednosti: x_{ij}=1 ako i-ti radnik dobije j-ti posao, x_{ij}=0 u protivnom slučaju (ako i-ti radnik ne dobije posao), kao što je prikazano formulom (2.12.34.).

$$x_{ij} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \qquad (2.12.34.)$$

Relacijama (2.12.31.) do (2.12.34.) formulisan je opšti model problema raspoređivanja. Očigledno da je problem ekvivalentan transportnom problemu u kome svako ishodište raspolaže samo jednom jedinicom proizvoda za transport, a potrebe odredišta, takođe, iznose po jednu jedinicu proizvoda, tj.

$$a_{\mathrm{i}} = b_j = 1 \qquad \text{za svako } i \text{ i svako } j.$$

Postoje, međutim, i druge razlike između ova dva modela:

– matrica koeficijenata c_{ij} u problemu rasporedivanja mora biti kvadratna, tj. mora da ima n redova i n kolona,

- promenljive u problemu raspoređivanja mogu uzimati ili vrednost jedan ili vrednost nula,
- problem raspoređivanja ima svojstvo da je svako njegovo bazično moguće rešenje degenerisano jer ima samo n bazičnih promenljivih $x_{ij} = 1$.

Problem raspoređivanja je po formi linearni model, pa se i za njega može formirati odgovarajući dualni model. Njegov dualni model može se formulisati na sledeći način:

Potrebno je pronaći maksimalnu vrednost funkcije kriterijuma, kao što je prikazano formulom (2.12.35.):

$$G = \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j \qquad (2.12.35.)$$

uz ograničenja koja su prikazana formulom (2.12.36.):

$$u_i + v_j \leq c_{ij}, i,j = 1, 2, 3, ..., n \qquad (2.12.36.)$$

2.12.4.2. Rešavanje problema raspoređivanja

Mađarski metod, koji se koristi za rešavanje problema raspoređivanja, temelji se na teoremi o broju *nezavisnih nula* u matrici. Pojam nezavisnih nula u matrici se određuje na sledeći način. Ako se u matrici odabere nula tako da svakom redu, ili koloni, pripadne najviše po jedna takva nula, onda se odabrane nule nazivaju nezavisnim nulama.

Teorema o broju nezavisnih nula u matrici tvrdi: *Ako se u nekoj matrici nalaze i nule, onda je maksimalan broj nezavisnih nula, koje se mogu među njima odabrati, jednak minimalnom broju linija koje pokrivaju sve nule.*

Ova teorema omogućava da se problem raspoređivanja može formulisati i na sledeći način: U matrici iz tabele 2.12.93. treba odrediti pet nezavisnih nula tako da zbir odgovarajućih elemenata bude minimalan. Naravno, nezavisne nule se mogu odrediti pošto se izvrši transformacija elemenata matrice i tabele 2.12.93.

Transformacija matrice koeficijenata c_{ij} zasniva se na sledećem stavu: *Optimalno rešenje problema raspoređivanja neće se promeniti ako se svakom elementu jednog reda, odnosno jedne kolone, matrice oduzme (ili doda) jedan isti broj.*

Na osnovu prethodnog stava, uvek se može izvršiti takva transformacija koeficijenata c_{ij} kojom će se obezbediti da matrica u svakom redu i svakoj koloni sadrži najmanje po jednu nulu. Time su stvoreni uslovi za primenu mađarskog metoda.

2.12.4.3. Minimalna vrednost funkcije kriterijuma

- **Kvadratna matrica koeficijenata**

U pomenutom primeru (tabela 2.12.93.) potrebno je pronaći takvo rešenje koje će obezbediti da funkcija kriterijuma postigne svoju minimalnu vrednost. Pet je poslova i pet radnika, što znači da je matrica koeficijenata c_{ij} kvadratna. Postupak rešavanja problema mađarskim metodom odvija se u okviru nekoliko koraka i može se opisati na sledeći način:

Korak 1. U ovom koraku vrši se transformacija koeficijenata matrice efikasnosti radnika, c_{ij}, u nove koeficijente, $\bar{c}_{ij}$, pri čemu je:

$$\bar{c}_{ij} = c_{ij} - u_i - v_j \qquad (2.12.37.)$$

Transformacija se vrši tako što se u svakom redu matrice odabere najmanji koeficijent

$$u_i = \min_j c_{ij}\,, \qquad i = 1, 2, ..., n \qquad (2.12.38.)$$

i oduzme od svih elemenata odgovarajućeg reda. Zatim se u novoj matrici pronađe najmanji element u svakoj koloni:

$$v_j = \min_i (c_{ij} - u_i)\,, \qquad j = 1, 2, ..., n \qquad (2.12.39.)$$

pa se oduzme od svih elemenata kolone. Posle ovih transformacija, svaki red i svaka kolona matrice koeficijenata $\bar{c}_{ij}$ mora imati najmanje po jednu nulu.

U našem primeru, u tabeli 2.12.93., najmanji elementi po redovima su:

$$u_1 = 8, \quad u_2 = 7, \quad u_3 = 9, \quad u_4 = 6, \quad u_5 = 7.$$

Posle izvršenih oduzimanja, dobija se matrica koja je prikazana u tabeli 2.12.95.

Tabela 2.12.95. *Transformacija početnih koeficijenata c_{ij} - po redovima*

6	1	4	0	8
1	0	2	2	7
0	2	1	1	3
4	2	2	0	8
4	2	3	0	6

U matrici iz tabele 2.12.94. treća i peta kolona nemaju nule, pa se određuju najmanji elementi po kolonama. To su:

$$v_1 = 0, \quad v_2 = 0, \quad v_3 = 1, \quad v_4 = 0, \quad v_5 = 3.$$

Nakon njihovog oduzimanja po kolonama, dobija se transformisana matrica koja je prikazana u tabeli 2.12.96. Ova matrica sadrži bar po jednu nulu u svakom redu i svakoj koloni.

Tabela 2.12.96. *Transformacija početnih koeficijenata c_{ij} - po kolonama*

6	1	3	0	5
1	0	1	2	4
0	2	~~0~~	1	~~0~~
4	2	1	~~0~~	5
4	2	2	~~0~~	3

Korak 2. U ovom koraku se dređuju nezavisne nule u matrici. Polazi se od reda u kome je samo jedna nula, a to je prvi red. Njenim zaokruživanjem se označava da je ona nezavisna. U četvrtoj koloni postoje još dve nule. One moraju biti zavisne, pa se zato one precrtavaju.

I drugi red ima samo jednu nulu (u drugoj koloni), koja će se proglasiti za nezavisnu i koja se zaokružuje. U drugoj koloni nema više nula, pa se prelazi na naredni red sa jednom nulom. Pošto nema takvog reda, onda će se jedna nula iz trećeg reda proglasiti za nezavisnu. Proizvoljno se određuje da je nula iz prve kolone trećeg reda nezavisna, pa se ona taokružuje, a ostale nule će biti zavisne, pa se one precrtavaju.

Kada je broj nezavisnih nula jednak broju redova, odnosno broju kolona, nađeno je optimalno rešenje. Mesta nezavisnih nula određuju promenljive čija je vrednost jedan. U našem primeru nema pet nezavisnih nula, što znači da nije nađeno optimalno rešenje i da postupak treba nastaviti.

Korak 3. U ovom koraku se vrši novu transformacija koeficijenata matrice. Zato se određuje minimalni broj linija koje će pokriti sve nule u matrici. Transformacija se zatim vrši tako što se najmanji nepokriveni elemenat oduzme od svih nepokrivenih elementa, a doda svim elementima koji leže na preseku dveju linija.

Za određivanje sistema linija kojima će se pokriti sve nule u matrici treba koristiti sledeći postupak:

a) označe se svi redovi koji nemaju nezavisnu nulu,
b) zatim se precrtaju sve kolone koje imaju nule u označenim redovima,
c) označe se redovi koji imaju nezavisnu nulu u precrtanim kolonama,
d) postupci pod b) i c) ponavljaju se dokle god je to moguće,
e) na kraju se precrtaju svi neoznačeni redovi.

Opisani postupak je korektno vođen u sledećem slučaju:

- po završenom postupku sve nule u matrici moraju biti precrtane,
- broj linija kojima su sve nule precrtane jednak je broju nezavisnih nula,
- nezavisna nula ne sme se naći na preseku linija, i
- svaka linija sme prolaziti samo kroz jednu nezavisnu nulu.

Za matricu iz tabele 2.12.96. se može konstruisati sistem linija pokrivanja na sledeći način: Četvrti i peti red nemaju nezavisne nule, pa se oni označavaju strelicom. Označeni redovi imaju zavisne nule u četvrtoj koloni, pa se precrtava četvrta kolona (isprekidana linija). U četvrtoj koloni je precrtana nezavisna nula iz prvog reda. Zato se označava i prvi red (duple strelice). U prvom redu nema više zavisnih nula, pa je postupak označavanja redova i precrtavanja kolona završen. Precrtavaju se neoznačeni drugi i treći red. Dobijeni rezultat je prikazan u tabeli 2.12.97.

Tabela 2.12.97. *Precrtavanje redova i kolona u prvoj iteraciji*

6	1	3	0	5
1	0	1	2	4
0	2	~~0~~	1	~~0~~
4	2	1	~~0~~	5
4	2	2	~~0~~	3

Sada se određuje najmanji nepokriveni elemenat u tabeli 2.12.97. To je elemenat $\bar{c}_{12} = \bar{c}_{43} = 1$. Kada se oduzme 1 od svakog neprecrtanog elementa i doda 1 svakom elementu na preseku linija, dobija se novu matrica, kao što je prikazano u tabeli 2.12.98.

Tabela 2.12.98. *Transformisana matrica nakon prve iteracije*

5	~~0~~	2	0	4
1	0	1	3	4
0	2	~~0~~	2	~~0~~
3	1	0	~~0~~	4
3	1	1	~~0~~	2

Korak 4. Ovaj korak sastoji se u ponavljanju koraka 2 i 3 dok se ne dobije optimalno rešenje. Prema tome, vrši se kategorizacija nula iz tabele 2.12.98.

Drugi red ima samo jednu nulu i ona je nezavisna. Usled toga je nula u prvom redu i drugoj koloni zavisna, i ona se zaokružuje. Sada je u prvom redu ostala još jedna nula, pa će se ona proglasiti za nezavisnu i precrtaće se. U prvom redu ima još jedna nula. To je nula iz četvrte kolone i ona se proglašava za nezavisnu i zaokružuje se. To će zahtevati da nule u četvrtom i petom redu četvrte kolone budu zavisne, pa se one precrtavaju. U četvrtom redu preostala je još jedna nula iz treće kolone. Ona je nezavisna, a nula u trećem redu i trećoj koloni je zavisna. Postoje neoznačene nule još u trećem redu. Nula iz prve kolone se bira za nezavisnu, pa je nula iz pete kolone zavisna. Nezavisne nule se zaokružuju, a zavisne precrtavaju.

U tabeli 2.12.98. nije pronađeno optimalno rešenje jer postoje samo četiri nezavisne nule, pa se postupak nastavlja ponavljanjem koraka 3 i 4. Potrebno je odrediti sistem linija kojim će se pokriti sve nule u tabeli 2.12.98.

Peti red nema nezavisne nule i on je označen (strelica). U označenom redu se pronalazi zavisna nula. To je nula iz četvrte kolone, pa je precrtana četvrta kolona (isprekidana linija). Sada je u četvrtoj koloni precrtana i nezavisna nula iz prvog reda, pa se označava i prvi red (dupla strelica). Potom se, u označenom prvom redu, precrtava druga kolona jer ima zavisnu nulu u prvom redu. Opet je precrtana i nezavisna nula iz drugog reda, pa se označava drugi red (dupla strelica). Time je postupak označavanja i precrtavanja završen. Ostaje da se precrtaju svi neoznačeni redovi. Tako je konstruisan sistem linija precrtavanja, kao što je prikazano u tabeli 2.12.99.

Tabela 2.12.99. *Precrtavanje redova i kolona u drugoj iteraciji*

⇉	5	~~0~~	2	0	4
⇉	1	0	1	3	4
	0	2	~~0~~	2	~~0~~
	3	1	0	~~0~~	4
→	3	1	1	~~0~~	2

Najmanji nepokriveni elemenat u tabeli 2.12.99., je $\bar{c}_{21} = \bar{c}_{23} = \bar{c}_{53} = 1$. Kada se 1 oduzme od svakog neprecrtanog elementa i doda svakom elementu na preseku linija, dobija se tabela 2.12.100.

Tabela 2.12.100. *Transformisana matrica nakon druge iteracije*

4	0	1	~~0~~	3
0	~~0~~	~~0~~	3	3
~~0~~	3	~~0~~	3	0
3	2	0	1	4
2	1	~~0~~	0	1

U tabeli 2.12.100. se dređuju nezavisne nule. U četvrtom redu postoji samo jedna nula i ona je nezavisna, pa se ona zaokružuje. Usled toga, sve nule treće kolone moraju biti zavisne i one su precrtane. Sada peti red ima samo jednu nulu u četvrtoj koloni i ona je nezavisna. Zato će nula iz prvog reda četvrte kolone biti zavisna, pa sada i prvi red ima jednu nulu, koju se proglašava za nezavisnu. Zbog toga je nula u drugom redu druge kolone zavisna. U drugom redu ostala je neoznačena nula iz prve kolone, pa je proglašena za nezavisnu. Pošto se precrta nula iz trećeg reda prve kolone, ostaje poslednja nula u petoj koloni trećeg reda, koju je proglašena za nezavisnu.

Dobijeno je pet nezavisnih nula, što znači da je rešenje iz tabele 2.12.100. optimalno. Optimalno rešenje sačinjavaju sledeće promenljive:

$$x_{12} = 1 \quad x_{21} = 1 \quad x_{35} = 1 \quad x_{43} = 1 \quad x_{54} = 1$$

Optimalno rešenje je predstavljeno u matrici sa polaznim parametrima, i izgleda kao što je prikazano u tabeli 2.12.101.

Minimalna vrednost funkcije kriterijuma iznosi:

$$F(x) = 9+8+12+8+7= 44 \text{ sati.}$$

Tabela 2.12.101. *Optimalni raspored radnika*

Radnici	Poslovi				
	P_1	P_2	P_3	P_4	P_5
R_1	14 0	9 **1**	12 0	8 0	16 0
R_2	8 **1**	7 0	9 0	9 0	14 0
R_3	9 0	11 0	10 0	10 0	12 **1**
R_4	10 0	8 0	8 **1**	6 0	14 0
R_5	11 0	9 0	10 0	7 **1**	13 0

Optimalno rešenje može se objasniti na sledeći način:
Radnik R_1 dobiće posao P_2, radnik R_2 posao P_1, radnik R_3 posao P_5, radnik R_4 posao P_3 i radnik R_5 posao P_4. Oni će sve poslove obaviti za 44 sata rada i to je najmanji broj sati za obavljanje svih poslova.

- **Nekvadratna matrica koeficijenata**

Pokazaće se na jednom primeru kako se rešavaju problemi raspoređivanja u kojima broj redova nije jednak broju kolona. U pitanju je jedno transportno preduzeće koje ima u jednom trenutku 4 slobodna kamiona. Kamioni se nalaze u garažama: G_1, G_2, G_3 i G_4, koje su u različitim mestima. Potrebno je uputiti po jedan kamion na pet različitih utovarnih mesta: M_1, M_2, M_3, M_4 i M_5. Rastojanja garaža od utovarnih mesta su različita. Tabela 2.12.102. sadrži podatke o rastojanjima (c_{ij}) između utovarnih mesta i garaža.

Potrebno je odrediti iz kojih garaža i na koja utovarna mesta treba, uputiti kamione, pa da pređeni put svih kamiona bude najmanji, kao i koje utovarno mesto neće dobiti kamion.

Tabela 2.12.102. *Rastojanja garaža od utovarnih mesta*

Utovarna mesta	Garaže			
	G_1	G_2	G_3	G_4
M_1	12	11	12	13
M_2	9	16	10	13
M_3	11	10	9	10
M_4	15	13	12	13
M_5	11	14	11	15

Problem se rešava tako što će se proširiti matrica iz tabele 2.12.102. sa još jednom kolonom, tj. nepostojećom garažom, čija je udaljenost od svih utovarnih mesta jednaka nuli. Na taj način se dobija kvadratna matrica, pa se problem rešava na poznati način. Nova, proširena matrica je data u tabeli 2.12.103.

Tabela 2.12.103. *Proširena početna matrica*

12	11	12	13	0
9	16	10	13	0
11	10	9	10	0
15	13	12	13	0
11	14	11	15	0

Proširenjem matrice novom kolonom, u svakom redu matrice već ima po jedna nula. Zato se transformacija koeficijenata c_{ij} nastavlja po kolonama. Pošto se odrede najmanji koeficijenti po kolonama i oduzmu se od ostalih, dobija se tabela 2.12.104.

Tabela 2.12.104. *Precrtavanje redova i kolona u prvoj iteraciji*

3	1	3	3	0
0	6	1	3	$\cancel{0}$
2	0	$\cancel{0}$	$\cancel{0}$	$\cancel{0}$
6	3	3	3	$\cancel{0}$
2	4	2	5	$\cancel{0}$

U tabeli 2.12.104. izvršena je i kategorizacija nula. Kako je broj nezavisnih nula manji od pet, određen je i sistem linija kojima su prekrivene sve nule u matrici. Najmanji neprecrtani elemenat je $\bar{c}_{12} = 1$. Kada se on oduzme od neprecrtanih elemenata i doda elementima na preseku linija, dobija se vrednost kao što je prikazano u tabeli 2.12.105.

Tabela 2.12.105. *Precrtavanje redova i kolona u drugoj iteraciji*

2	0	2	2	$\cancel{0}$
0	6	1	3	1
2	$\cancel{0}$	0	$\cancel{0}$	1
5	2	2	2	0
1	3	1	4	$\cancel{0}$

Još uvek nije proriađeno optimalno rešenje. Određen je sistem linija kojima su precrtane sve nule u tabeli 2.12.105. Najmanji neprecrtani koeficijent je $\bar{c}_{51} = \bar{c}_{53} = 1$. Nakon izvršene transformacije dobijena je tabela 2.12.106.

Tabela 2.12.106. *Transformisana matrica nakon druge iteracije*

2	0	2	2	1
0	6	1	3	2
2	$\cancel{0}$	$\cancel{0}$	0	2
4	1	1	1	0
$\cancel{0}$	2	0	3	$\cancel{0}$

Dobijeno je pet nezavisnih nula, što znači da je rešenje iz tabele 2.12.106. optimalno, a njega sačinjavaju sledeće promenljive:

$$x_{12} = 1 \quad x_{21} = 1 \quad x_{34} = 1 \quad x_{45} = 1 \quad x_{53} = 1$$

Optimalno rešenje je predstavljeno u matrici sa polaznim parametrima, pa izgleda kao u tabeli 2.12.107. Minimalna vrednost funkcije kriterijuma iznosi:

$$F(x) = 11+9+10+0+11= 41.$$

Tabela 2.12.107. *Optimalni raspored kamiona*

Utovarna mesta	*Kamioni*				
	G_1	G_2	G_3	G_4	G_5
M_1	12 0	11 **1**	12 0	13 0	0 0
M_2	9 **1**	16 0	10 0	13 0	0 0
M_3	11 0	10 0	9 0	10 **1**	0 0
M_4	15 0	13 0	12 0	13 0	0 **1**
M_5	11 0	14 0	11 **1**	15 0	0 0

Optimalno rešenje objašnjavamo na sledeći način: Utovarno mesto M_1 dobija kamion iz garaže G_2, mesto M_2 iz garaže G_1, mesto M_3 iz G_4 i mesto M_5 iz garaže G_3. Utovarno mesto $M4$ treba da dobije kamion iz nepostojeće garaže, što znači da ono neće dobiti kamion. Minimalna vrednost funkcije kriterijuma označava najkraći put koji će preći svi kamioni od garaža do utovarnih mesta.

2.12.4.4. Maksimalna vrednost funkcije kriterijuma

Metodom raspoređivanja mogu se rešavati i problemi u kojima se traži maksimalna vrednost funkcije kriterijuma. Prikazaće se na jednom primeru kako se rešavaju ovi problemi. U pitanju je dorada proizvoda P koja se može izvršiti na 4 različite mašine. Preduzeće je obučilo 4 radnika za rad na ovim mašinama. Jedan radnik će raditi samo sa jednom mašinom. Provera stručne sposobnosti radnika izvršena je tako što je svaki radnik radio po jedan sat na svakoj mašini i za to vreme proizveo sledeću količinu proizvoda P – tabela 2.12.108.

Tabela 2.12.108. *Proizvedena količina proizvoda P*

Mašine	*Radnici*			
	R_1	R_2	R_3	R_4
M_1	6	9	9	11
M_2	4	5	11	8
M_3	9	5	12	7
M_4	8	10	13	8

Potrebno je izvršiti takav raspored radnika na mašinama da oni ostvare maksimalnu proizvodnju proizvoda *P*. Ovaj problem maksimuma se rešava na sledeći način:

1) Za svaki red tabele 2.12.108. odabere se najveći koeficijent c_{ij} i oduzme od ostalih koeficijenata iz tog reda. Tako se dobija tabela 2.12.109.
2) Svi koeficijenti iz tabele 2.12.109. pomnože se sa -1 i nastavi se sa rešavanjem problema tražeći minimalnu vrednost funkcije kriterijuma. Dalji tok rešavanja problema dat je u tabelama 2.12.110. do 2.12.113.

Tabela 2.12.109. *Transformacija početnih koeficijenata c_{ij} - po redovima*

-5	-2	-2	0
-7	-6	0	-3
-3	-7	0	-5
-5	-3	0	-5

Tabela 2.12.110. *Prevođenje problema tipa max u tip min*

5	2	2	0
7	6	0	3
3	7	0	5
5	3	0	5

Tabela 2.12.111. *Transformacija početnih koeficijenata c_{ij} - po kolonama*

	2	0	2	0
→	4	4	0	3
	0	5	0	5
→	2	1	0	5

Tabela 2.12.112. *Transformisana matrica nakon prve iteracije*

2	0	3	0
3	3	0	2
0	5	1	5
1	0	0	4

Tabela 2.12.113. *Optimalan plan proizvodnje proizvoda P*

Mašine	***Radnici***			
	R_1	***R_2***	***R_3***	***R_4***
M_1	6 / 0	9 / 0	9 / 0	11 / **1**
M_2	4 / 0	5 / 0	11 / **1**	8 / 0
M_3	9 / **1**	5 / 0	12 / 0	7 / 0
M_4	8 / 0	10 / **1**	13 / 0	8 / 0

Tabela 2.12.113. sadrži optimalno rešenje: Na mašini M_1 radiće radnik R_4, na mašini M_2 radnik R_3, na mašini M_3 radnik R_1 i na mašini M_4 radnik R_2. Njihova maksimalna proizvodnja proizvoda *P*, za jedan sat rada, je 41 komad, tj. $maxF(x)$=11+11+9+10=41.

2.12.4.5. Rešavanje transportnog problema mađarskim metodom

Optimalno rešenje transportnog problema, pored metoda raspodele i metoda koeficijenata (potencijala), može se pronaći i korišćenje ***mađarskog metoda***. Pošto su osnove mađarskog metoda objašnjene u poglavlju 2.12.4., ovde će se na jednom primeru pokazati kako se vrši njeno prilagođavanje za rešavanje transportnog problema. U pitanju je proizvođač koji proizvodi robu *A* na četiri proizvodna mesta. Robu je potrebno dostaviti do pet potrošačkih centara. Svi potrebni podaci dati su u tabeli 2.12.114.

Tabela 2.12.114. *Početni podaci*

Proizvođač	***Potrošački centri***				
	***P*₁ (60)**	**P₂ (50)**	***P*₃ (30)**	**P₄ (20)**	***P*₅ (40)**
***M*₁ (65)**	8	3	5	5	4
***M*₂ (50)**	2	5	3	6	8
***M*₃ (40)**	4	2	8	3	6
***M*₄ (45)**	3	6	9	5	3

U prvom koraku se vrši transformacija koeficijenata c_{ij}, prema relaciji (2.12.38.):

$$u_1 = 3 \quad u_2 = 2 \quad u_3 = 2 \quad u_4 = 3$$

a, prema relaciji (2.12.39.):

$$v_1 = 0 \quad v_2 = 0 \quad v_3 = 1 \quad v_4 = 1 \quad v_5 = 0$$

Transformisani koeficijenti se unose u tabelu 2.12.115. sa ostalim podacima o transportnom problemu.

Tabela 2.12.115. *Transformisana tabela po redovima i kolonama*

Proizvođač	***Potrošački centri***				
	***P*₁ (60)**	**P₂ (50)**	***P*₃ (30)**	**P₄ (20)**	***P*₅ (40)**
→ ***M*₁ (65)**	5	0 **50**	1	1	1
~~***M*₂ (50)**~~	0 **50**	3	~~0~~	3	6
→ ***M*₃ (40)**	2	~~0~~	5	0 **20**	4
~~***M*₄ (45)**~~	0 **10**	3	5	1	0 **35**

U drugom koraku se vrši kategorizacija nula. Ovde treba imati u vidu da će svaka nezavisna nula u matrici predstavljati onoliko redova, odnosno kolona, kolika je vrednost slobodnog člana posmatranog reda, odnosno kolone. Drugim rečima, slobodni članovi a_i i b_j određuju „multiplicitet“ nezavisne nule u preseku reda i kolone, pri čemu je multiplicitet nezavisne nule jednak manjem od ovih slobodnih članova. Zaokruženi brojevi u tabeli označavaju multiplicitet nezavisnih nula.

Nezavisne nule za tabelu 2.12.115. određene su na sledeći način. Prvi red ima samo jednu nulu i ona je nezavisna, pa se ona zaokružuje (krug). Njen multiplicitet je 50, a određen je na osnovu manjeg slobodnog člana prvog reda i druge kolone. Nula iz trećeg reda druge kolone je zavisna, pa se ona precrtava (linija), sada treći red ima samo jednu nulu, $\bar{c}_{34} = 0$. Multiplicitet ove nule je 20. Drugi i četvrti red imaju po dve nule, pa se nastavlja sa drugim redom. Nula u prvoj koloni ima veći multiplicitet, pa se određuje da je ona nezavisna, sa multiplicitetom 50. Druga nula u drugom redu je sada zavisna. Započelo se sa prvom kolonom. Njen multiplicitet nije ispunjen, pa će nula u četvrtom redu biti nezavisna sa preostalim multiplicitetom od 10. Konačno, nula iz pete kolone četvrtog reda je nezavisna sa multiplicitetom 35. Ovo još uvek ne predstavlja moguće rešenje jer nisu zadovoljena sva ograničenja.

Sada se prelazi na korak 3. U ovom koraku treba odrediti sistem linija prekrivanja nula u matrici kako bi se izvršila nova transformacija koeficijenata. Sistem linija za tabelu 2.12.115. određen je na sledeći način:
Nule u prvom i trećem redu imaju manje multiplicitete od vrednosti odgovarajućih slobodnih članova ovih redova. Zato su prvi i treći red označeni kao redovi koji nemaju nezavisne nule (strelice). Zatim su precrtane kolone (isprekidane linije) koje imaju nule u ovim redovima, a to su druga i četvrta kolona. Postupak označavanja redova je završen, pa se svi neoznačeni redovi precrtavaju.

Najmanji neprecrtani element u matrici je $\bar{c}_{13} = \bar{c}_{15} = 1$. Pošto se sve neprecrtane elemente matrice smanje za 1, a sve elemente sa preseka linija povećaju za 1, dobijaju se koeficijenti iz tabele 2.12.116. Rešenje iz tabele 2.12.116. je moguće rešenje, pa je ono, prema tome, i optimalno rešenje.

Tabela 2.12.116. *Optimalno rešenje*

Proizvođač	*Potrošački centri*				
	***P_1* (60)**	**P_2 (50)**	***P_3* (30)**	**P_4 (20)**	***P_5* (40)**
***M_1* (65)**	4	0 **30**	0 **30**	1	0 **5**
***M_2* (50)**	0 **50**	4	~~0~~	4	6
***M_3* (40)**	1	0 **20**	4	0 **20**	3
***M_4* (45)**	0 **10**	4	5	2	0 **35**

Treba napomenuti još da svi transformisani koeficijenti c_{ij} iz tabele 2.12.115. imaju iste vrednosti kao i dualne promenljive Δ_{ij} iz klasičnog transportnog problema, a koje se mogu odrediti za optimalno rešenje iz tabele 2.12.115.

2.13. CELOBROJNO PROGRAMIRANJE

Kod mnogih zadataka postoji zahtev da jedna, više ili sve promenljive moraju biti celobrojne, odnosno da uzmu vrednosti koje su celi brojevi. To je slučaj kada se traži optimalno rešenje koje se odnosi na broj proizvoda, broj angažovanih radnika, broj mašina koje treba kupiti, broj delova koje treba proizvesti i dr.

Transportni problem i problem rasporedjivanja po definiciji zadovoljavaju uslove celobrojnosti.

Prema prirodi zadataka, mogu se odvojiti tri tipa celobrojnog programiranja:

- čisto celobrojno programiranje, gde su sve promenljive celobrojne,
- mešovito celobrojno programiranje, gde su samo neke promenljive celobrojne, i
- binarno programiranje (0-1 programiranje), gde su sve ili neke promenljive binarne (uzimaju vrednosti 1 ili 0).

Kod čistog celobrojnog programiranja razmatra se sledeci matematički model

$$\begin{pmatrix}\min\\ \max\end{pmatrix} C^T X$$
$$Ax=B \qquad (2.13.1.)$$
$$x\geq 0,\ x\in Z^n$$

gde Z^n oznacava skup n-torki sa celobrojnim koordinatama, $b\in Z^m$, $c\in Z^n$.

Ako se izostavi uslov celobrojnosti dobija se linearna relaksacija problema

$$\begin{pmatrix}\min\\ \max\end{pmatrix} \mathrm{C^T X}$$
$$Ax=B \qquad (2.13.2.)$$
$$x\geq 0$$

Kod binarnog celobrojnog programiranja razmatra se sledeci matematički model

$$\begin{pmatrix}\min\\ \max\end{pmatrix} \mathrm{C^T X}$$
$$Ax=B \qquad (2.13.3.)$$
$$x\geq 0,$$
$$x\in\{0,1\}^n$$

Linearna relaksacija problema binarnog programiranja može izraziti na sledeći način:

$$\begin{pmatrix}\min \\ \max\end{pmatrix} C^T X$$

$$Ax=B \qquad (2.13.4.)$$

$$0 \le x \le 1$$

Metode celobrojnog programiranja se zasnivaju na inteligentnoj pretrazi dopustivog skupa. Najvažniji agoritmi za dobijanje celobrojnog rešenja su:

a) tačni agoritmi (metode odsecanja, metode grananja i odsecanja i metode grananja i ograničavanja), i
b) heuristički algoritmi (lokalno pretraživanje i simulacija kaljenja).

2.13.1. Tačni algoritmi

- **Metode odsecanja**

Metode odsecanja polaze od nekog necelobrojnog rešenja i dodavanjem novog ograničenja vrši se odsecanje deo konveksnog skupa mogućih rešenja zajedno sa tačkom u kojoj je bilo necelobrojno optimalno rešenje, pri čemu se ovim odsecanjem ne gubi ni jedno celobrojno rešenje. Nakon toga, ponovo se dodaje novo ograničnje i ceo postupak se ponavlja sve dok se ne dobije optimalno, celobrojno rešenje.

Broj iteracija zavisi od načina kako se biraju dodatna ograničenja, jer praktično postoji neograničen broj ograničenja koja se mogu dodati pri svakoj iteraciji. Shodno tome, Gomori (Gomory) je razvio algoritam (Gomorijev algoritam), pomoću koga se može dobiti optimalno celobrojno rešenje. To je bio prvi algoritam kojim se dolazi do optimalnog rešenja pri konačnom broju iteracija. Postupak je sledeći:

Korak 1: Posmatrati nultu vrstu odgovarajuće simplex tabele i u njoj odrediti *i* za koje je $\overline{b_i}$ (vrednost *b* kod optimalne simpleks tabele) maksimalno.

Korak 2: Napraviti rez

$$\sum \left(\overline{a_{ij}} - \lfloor \overline{a_{ij}} \rfloor\right) \cdot x_j + x_{n+1} = \lfloor \overline{bi} \rfloor - \overline{b_i} \qquad (2.13.5.)$$

i dodati ga na kraj tabele.

Korak 3: Primenom dualnog simplex algoritma naći rešenje problema tako što se pivot bira na sledeći način:

a) početi sa kolonama *j = 1,..., n+1* kod kojih je lex pozitivan, i
b) izabrati bilo koju kolonu *l* gde je $b_i < 0$ kao pivot vrstu. Primeniti dualni simplex algoritam.

- **Metode grananja i odsecanja**

Ove metode se baziraju na formiranju linearne nejednačine koja će zadovoljavati sva celobrojna dopustiva rešenja početnog problema, dok necelebrojna neće. Problem se prvo relaksira ukidanjem uslova celobrojnosti i nalazi se optimalno, necelobrojno rešenje (tačku). Potom metod nalazi hiperravan koja razdvaja tu tačku od svih dopustivih celobrojnih rešenja. Ova hiperravan je linearna nejednačina i njenim uključivanjem u ograničenja dobija se modifikovani linearni problem. Dati postupak se ponavlja sve dok se ne dobije celobrojno rešenje. Za rešavanje linearnog programiranja se koristi Simplex metod, tako što se malo menja oblik početne jednačine.

- **Metode grananja i ograničavanja**

Kod ovih metode se problemi razlažu na prostije potprobleme (grananje), pri čemu se mnogi od ovih potproblema ne rešavaju ako se proceni da nemaju bolje rešenje od trenutno najboljeg rešenja. Trenutno najboljeg rešenja se može dobiti neposrednom proverom ili nekim heurističkim postupcima u toku algoritma. U daljem postupku se vrši rešavanje svakog potproblema. Optimalna vrednost problema jednaka je trenutnom rekordu M ukoliko je M konačan broj (M je najbolje pronađeno rešenje). Algoritam ovih metoda se posle konačno mnogo iteracija završava nalaženjem optimalnih vrednosti i rešenja polaznog problema.

- **Programski paketi za dobijanje celobrojnog programiranja**

Programski paketi imaju mogućnost rešavanja i celobrojnih problema linearnog programiranja.

Primer 2.13.1. Dat je matematički problem linearnog programiranja:

$$\begin{aligned}
&\max F(x) = 450x_1 + 425x_2 + 226x_3 + 350x_4\\
&6x_1 + 3x_2 + 9x_3 + 9x_4 \leq 150\\
&7x_1 + 9x_2 + 2x_3 + 5x_4 \leq 130\\
&8x_1 + 2x_2 + 7x_3 + 6x_4 \leq 160\\
&x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

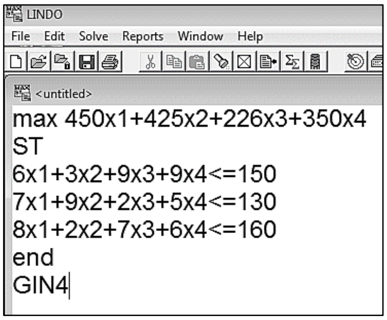
Problem rešiti pomoću programskih paketa LINDO, LINGO i QM for Windows. Pri tome, treba odrediti rešenje kod kojih su:

a) sve promenljive celobrojne,
b) x_1 i x_4 celobrojne promenljive, a x_2 i x_3 ne moraju biti celobrojne.

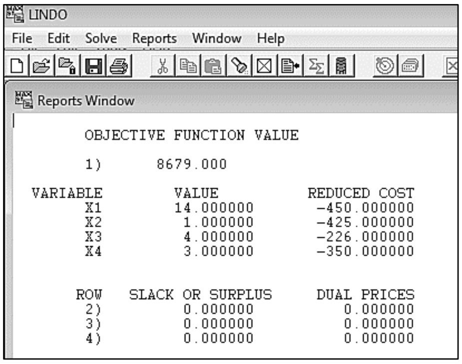
Rešenje.

- LINDO program

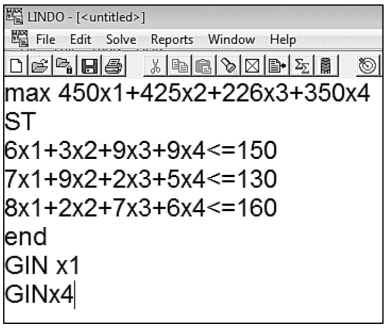
Lindo program koristi komandu GIN za dobijanje celobrojnog rešenja problema, koja se unosi ispod komande END. Ako se traži da sve promenljive budu celobrojne onda komanda ima oblik GIN *k*, gde je *k* broj promenljivih, U našem zadatku ima 4 promenljivih (x_1, x_2, x_3, x_4), tako da komanda ima sledeći oblik: GIN 4 (slika 2.13.1.). Dobijeno rešenje je prikazano na slici 2.13.2. Kada se traži da rešenje bude mešovito onda se komandom GIN označava promenljive koje trebaju da uzmu celobrojne vrednosti (u našem zadatku bi bilo GIN x_1 ispod komande END, a ispod njega bi bila komanda GIN x_4) – slika 2.13.3. Dobijeno rešenje je prikazano na slici 2.13.4.



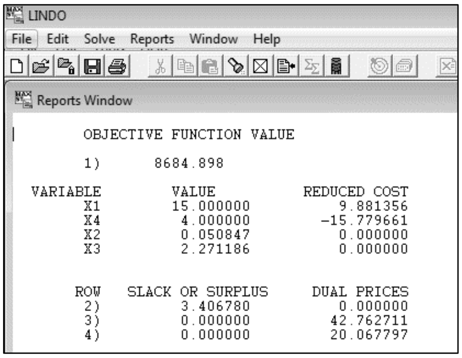
Slika 2.13.1. *Postavka zadatka kod celobrojnog problema (Lindo)*



Slika 2.13.2. *Rešenje zadatka celobrojnog problema (Lindo)*



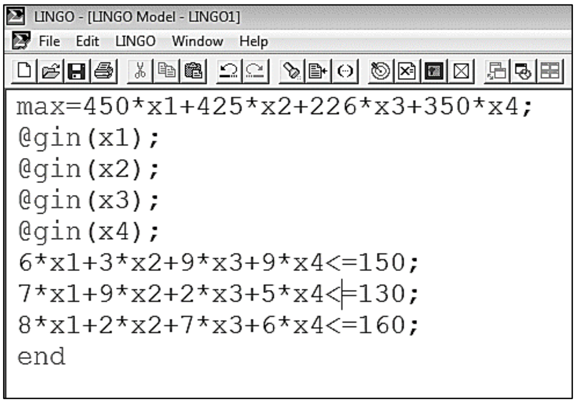
Slika 2.13.3. *Postavka zadatka kod mešovitog celobrojnog problema (Lindo)*



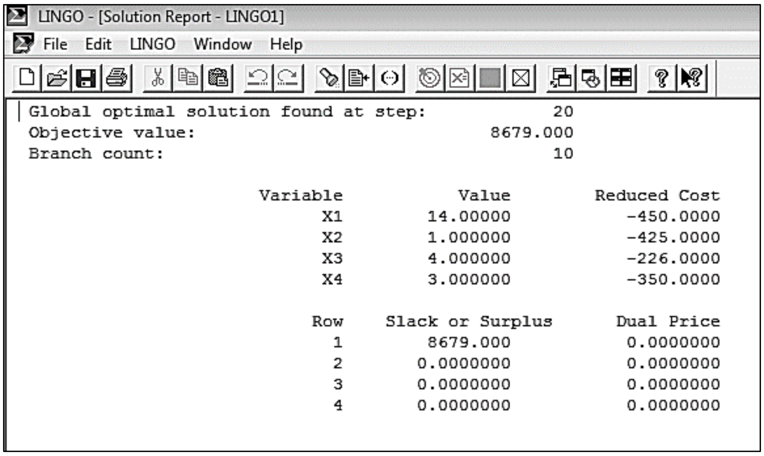
Slika 2.13.4. *Rešenje zadatka mešovitog celobrojnog problema (Lindo)*

- LINGO program

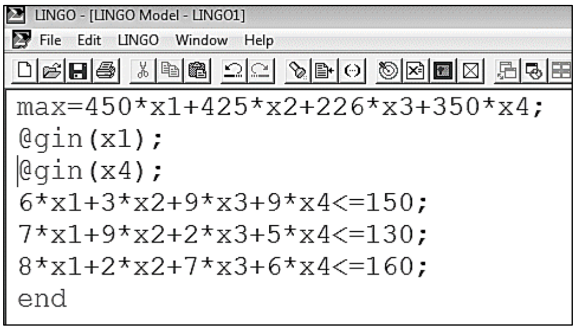
Lingo program takođe koristi komandu GIN za dobijanje celobrojnog rešenja problema, koja se unosi ispod funkcije cilja. Ova komanda se daje za svaku promenljivu za koju se traži celobrojno rešenje. Komanda ima sledeći oblik: @GIN(x_j); (slike 2.13.5. i 2.13.7.). Dobijeno rešenje je prikazano na slikama 2.13.6. i 2.13.8.



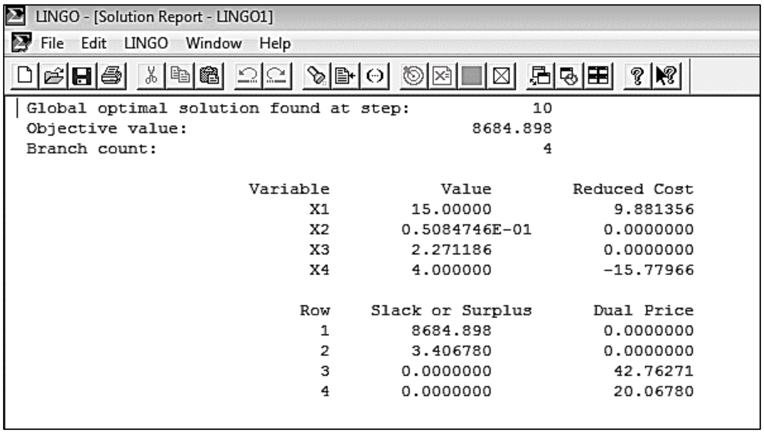
Slika 2.13.5. *Postavka zadatka kod celobrojnog problema (Lingo)*



Slika 2.13.6. *Rešenje zadatka celobrojnog problema (Lingo)*



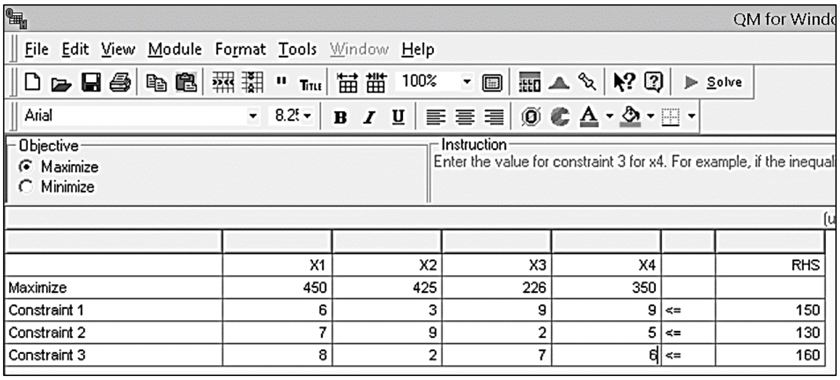
Slika 2.13.7. *Postavka zadatka kod mešovitog celobrojnog problema (Lingo)*



Slika 2.13.8. *Rešenje zadatka mešovitog celobrojnog problema (Lingo)*

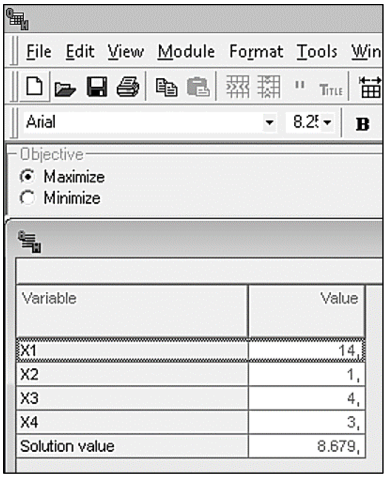
- QM for Windows program

Kod programa QM for Windows u meniju “Module” (slika 2.10.5.) bira se opcija “Integer programming” za dobijanje celobrojnog rešenja problema. Nakon toga definišu se parametri problema i unose se koeficijenti matematičkog modela. Na slici 2.13.9. prikazani su ulazni podaci problema , a na slici 2.13.10. prikazano je rešenje problema.



	X1	X2	X3	X4		RHS
Maximize	450	425	226	350		
Constraint 1	6	3	9	9	<=	150
Constraint 2	7	9	2	5	<=	130
Constraint 3	8	2	7	6	<=	160

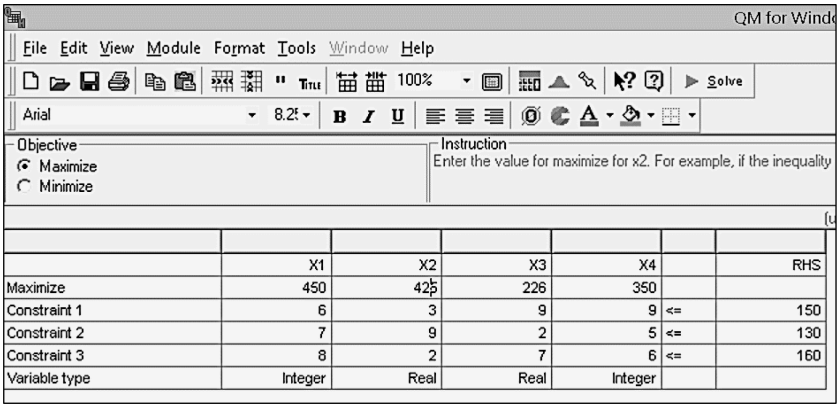
Slika 2.13.9. *Postavka zadatka kod celobrojnog problema (QM for Windows)*



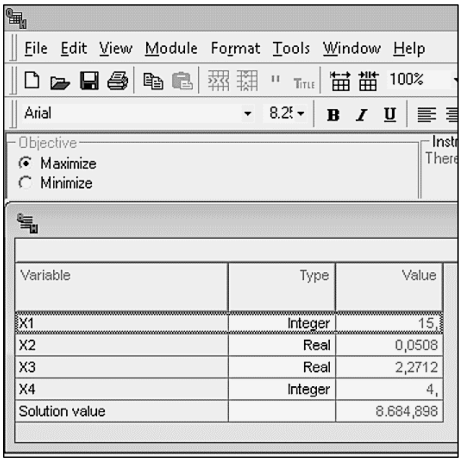
Variable	Value
X1	14,
X2	1,
X3	4,
X4	3,
Solution value	8.679,

Slika 2.13.10. *Rešenje zadatka celobrojnog problema (QM for Windows)*

Kada se traži da pojedine promenljive uzmu cele brojeve koristi se komanda “Mixed Integer programming” u meniju “Module”. Nakon ubacivanja koeficijenata, u zadnjem redu tabele (slika 2.13.11.) bira se opcija za dobijanje celobrojnog rešenja problema (integer), ili necelobrojnog (real). Postoji i opcija za 0/1 programiranje. Na slici 2.13.12. prikazano je rešenje problema.



Slika 2.13.11. *Postavka zadatka kod mešovitog celobrojnog problema (QM for Windows)*



Slika 2.13.12. *Rešenje zadatka mešovitog celobrojnog problema (QM for Windows)*

2.13.2. Heuristički algoritmi

Heurističke metode omogućavaju dobijanje suboptimalnih rešenja, ali bez garancija u vezi njihove tačnosti. Njihov cilj je dobijanje brzih rešenja. Nekada one mogu biti najbolji izbor za rešavanje velikih ili teških problema za koje tačni algoritmi ne mogu da nađu razumno rešenje u vremenu. Danas postoji veći broj heurističkih više metoda.

- **Lokalno pretraživanje**

Ova metoda polazi od bilo kog necelobrojnog rešenja i ispituje sva susedna rešenja koja su celobrojna. Ako se nađe bolje susedno rešenje, ono se izabere i ponavlja se postupak. U suprotnom, algoritam se zaustavlja. Ovde je veoma važna definicija susedstva (okoline necelobrojnog rešenja). Ona zavisi od vrste problema koje se rešava i predstavlja glavnu razliku između različitih algoritama lokalnog pretraživanja.

- **Simulacija kaljenja**

To je metod koji pokušava da pobegne od lokalnog minimuma x, dozvoljavajući pri tome sa određenom veravotnoćom da se izabere lošije rešenje u odnosu na x iz njegovog susedstva.

Pri proračunu, ova metoda koristi parametar T koji se naziva temperatura. Verovatnoća izbora lošijeg rešenja je bliža jedinici za velike vrednosti parametra T, a bliža nuli za male vrednosti T. Simulacija kaljenja je iterativni metod i parametar T se obično menja pri svakoj iteraciji.

2.13.3. Binarno programiranje

U praksi postoje problemi u kojima se traži da sve promenljive budu binarne (uzimaju vrednosti 1 ili 0) i za koje su predloženi različiti algoritmi za njihovo rešavanje. Jedan od postojećih algoritma ima u osnovi istu strukturu kao metod grananja i ograničavanja. Kod ovog algoritma, odgovarajući relaksacioni problem (0-1) se dobija tako što se ignorišu sva ograničenja, osim onih koje navode da su promenljive binarne. Zatim se dolazi do delimičnog (parcijalnog rešenja) gde nisu određene vrednosti nekih promenljivih. Na kraju se vrši kompletiranje parcijalnog rešenja 0-1 programiranja tako što se vrši dodeljivanje vrednosti promenljivim sa nespecifiranim vrednostima u parcijalnom rešenju.

- **Programski paketi za dobijanje binarnog programiranja**

Programski paketi takođe imaju mogućnost rešavanja i binarnih problema linearnog programiranja.

Primer 2.13.2. Dat je matematički problem linearnog programiranja:

$$\max F(x) = 15x_1 + 13x_2 + 11x_3$$
$$8x_1 + 5x_2 + 3x_3 \leq 15$$
$$x_1, x_2, x_3 \geq 0$$

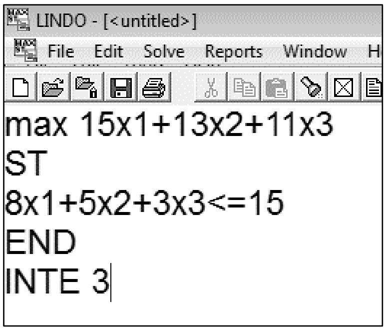
Problem rešiti pomoću programskih paketa LINDO, LINGO i QM for Windows. Pri tome, treba odrediti rešenje kod kojih su:

c) sve promenljive binarne,
d) x_2 i x_3 binarne promenljive, a x_1 ne mora biti binaran.

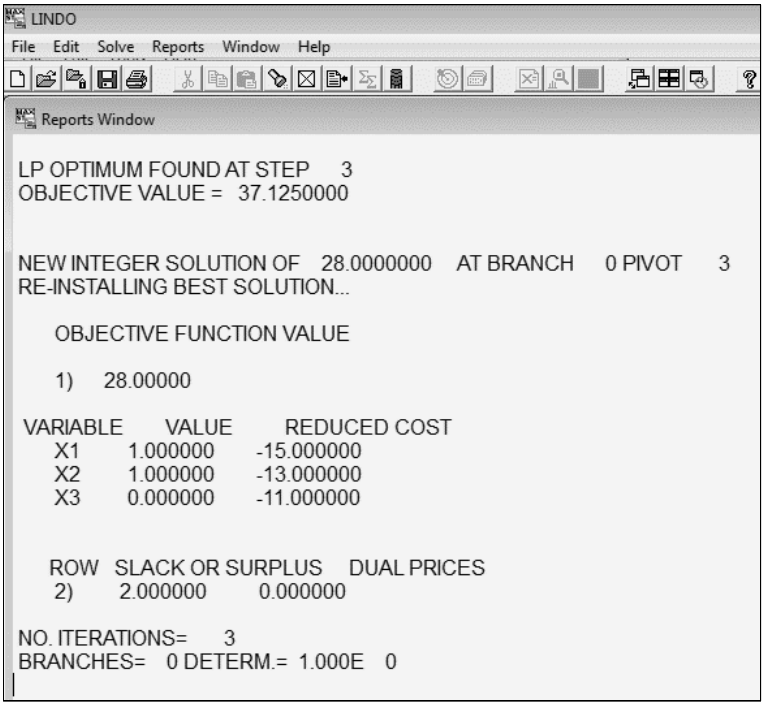
Rešenje.

- LINDO program

Lindo program koristi komandu INTE za dobijanje binarnog rešenja problema, koja se unosi ispod komande END. Ako se traži da sve promenljive budu binarne onda komanda ima oblik INTE *k*, gde je *k* broj promenljivih, U našem zadatku ima 3 promenljivih (x_1, x_2, x_3), tako da komanda ima sledeći oblik: INTE 3 (slika 2.13.13.). Dobijeno rešenje je prikazano na slici 2.13.14.



Slika 2.13.13. *Postavka zadatka kod binarnog problema (Lindo)*

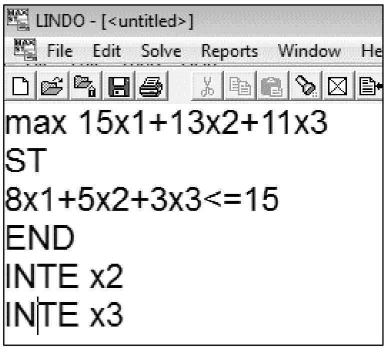


Slika 2.13.14. *Rešenje zadatka binarnog problema (Lindo)*

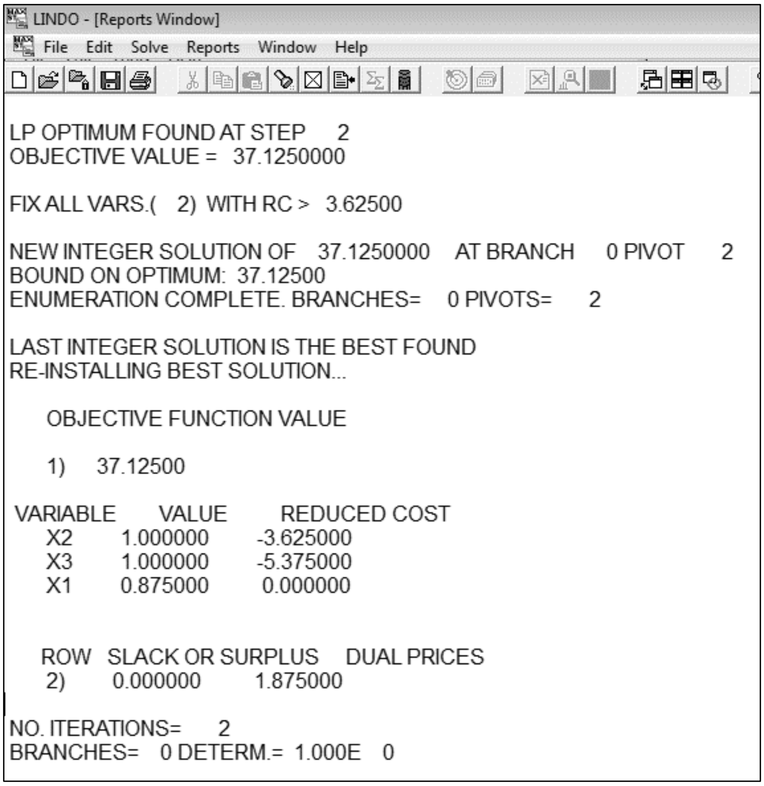
Kada se traži da rešenje bude mešovito onda se komandom INTE označava promenljive koje trebaju da uzmu celobrojne vrednosti (u našem zadatku bi bilo INTE x_2 ispod komande END, a ispod njega bi bila komanda INTE x_3) – slika 2.13.15. Dobijeno rešenje je prikazano na slici 2.13.16.

- LINGO program

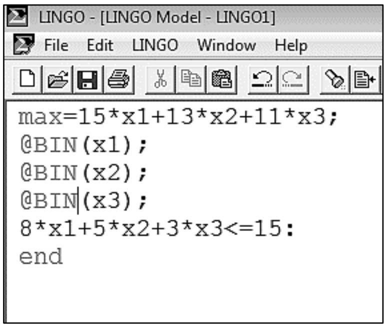
Lingo program koristi komandu BIN za dobijanje binarnog rešenja problema, koja se unosi ispod funkcije cilja. Ova komanda se daje za svaku promenljivu za koju se traži binarno rešenje. Komanda ima sledeći oblik: @BIN(x_j); (slike 2.13.17. i 2.13.19.). Dobijeno rešenje je prikazano na slikama 2.13.18. i 2.13.20.



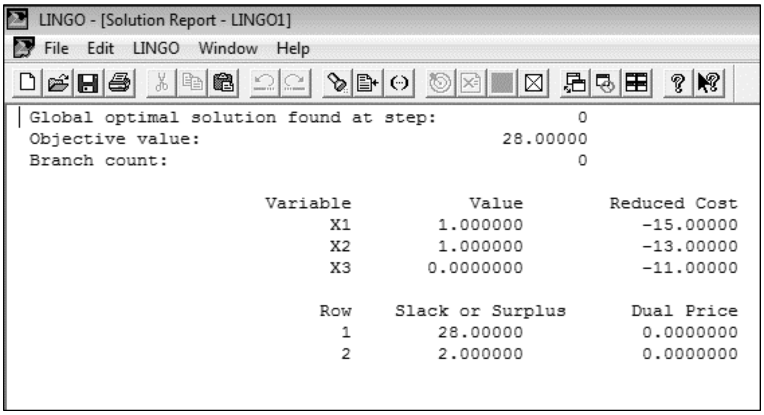
Slika 2.13.15. *Postavka zadatka kod mešovitog binarnog problema (Lindo)*



Slika 2.13.16. *Rešenje zadatka mešovitog binarnog problema (Lindo)*



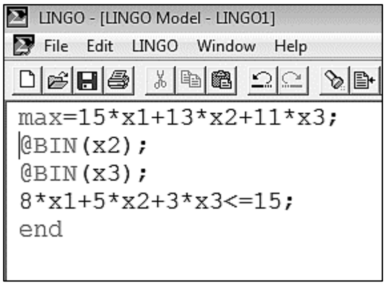
Slika 2.13.17. *Postavka zadatka kod binarnog problema (Lingo)*



LINGO - [Solution Report - LINGO1]
File Edit LINGO Window Help

```
Global optimal solution found at step:          0
Objective value:                         28.00000
Branch count:                                   0

                    Variable        Value      Reduced Cost
                          X1     1.000000         -15.00000
                          X2     1.000000         -13.00000
                          X3    0.0000000         -11.00000

                         Row  Slack or Surplus   Dual Price
                           1     28.00000         0.0000000
                           2     2.000000         0.0000000
```

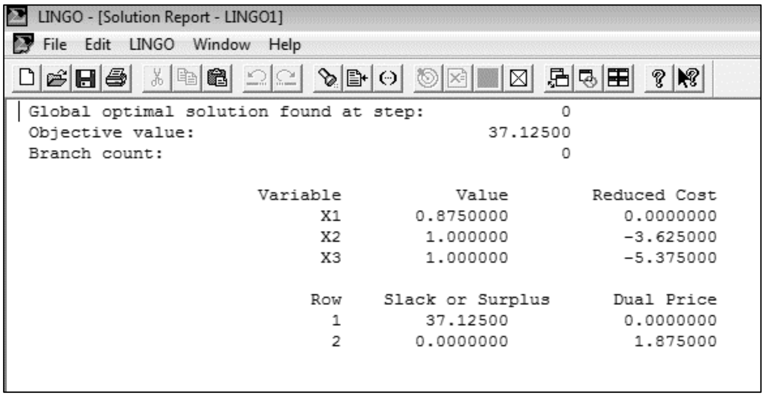
Slika 2.13.18. *Rešenje zadatka binarnog problema (Lingo)*



LINGO - [LINGO Model - LINGO1]
File Edit LINGO Window Help

```
max=15*x1+13*x2+11*x3;
@BIN(x2);
@BIN(x3);
8*x1+5*x2+3*x3<=15;
end
```

Slika 2.13.19. *Postavka zadatka kod mešovitog binarnog problema (Lingo)*



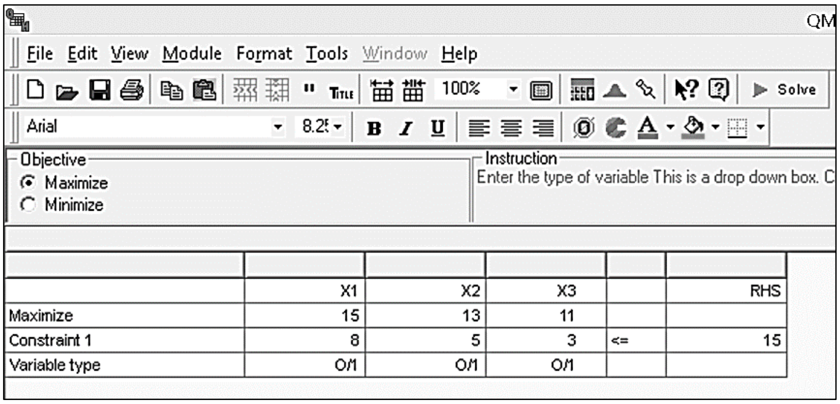
LINGO - [Solution Report - LINGO1]
File Edit LINGO Window Help

```
Global optimal solution found at step:          0
Objective value:                         37.12500
Branch count:                                   0

                    Variable        Value      Reduced Cost
                          X1    0.8750000         0.0000000
                          X2     1.000000         -3.625000
                          X3     1.000000         -5.375000

                         Row  Slack or Surplus   Dual Price
                           1     37.12500         0.0000000
                           2    0.0000000          1.875000
```

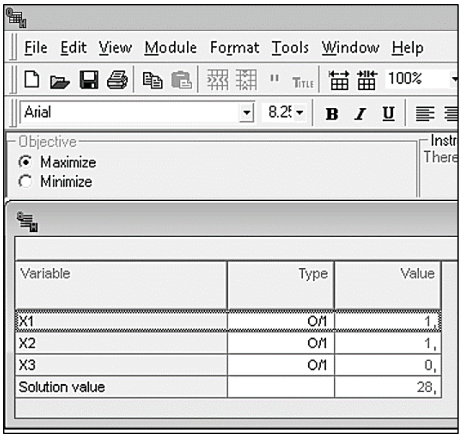
Slika 2.13.20. *Rešenje zadatka kod mešovitog binarnog problema (Lingo)*

- QM for Windows program

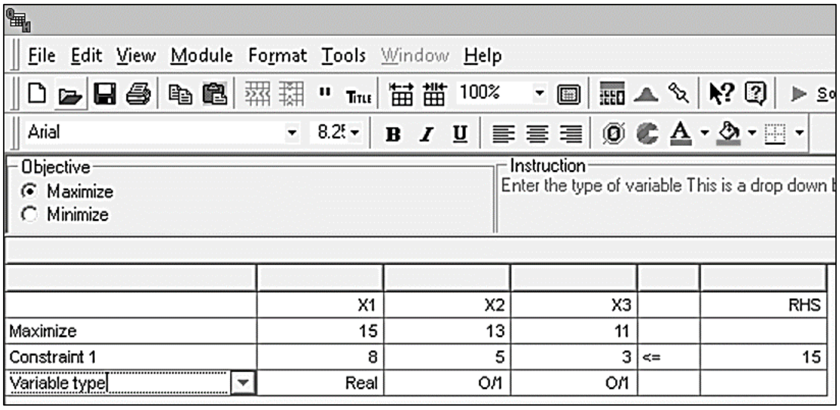
Kod programa QM for Windows u meniju “Module” (slika 2.10.5.) bira se opcija “Mixed Integer programming” za dobijanje binarnog rešenja problema. Nakon toga definišu se parametri problema i unose se koeficijenti matematičkog modela. Nakon ubacivanja koeficijenata, u zadnjem redu tabele bira se za svaku promenljivu opcija za dobijanje binarnog rešenja problema (0/1 programiranje), ili necelobrojnog (real). Na slici 2.13.21. prikazani su ulazni podaci problema , a na slici 2.13.22. prikazano je rešenje binarnog problema. Na slici 2.13.23. prikazani su ulazni podaci mešovitog binarnog problema, a na slici 2.13.24. prikazano je rešenje mešovitog binarnog problema.



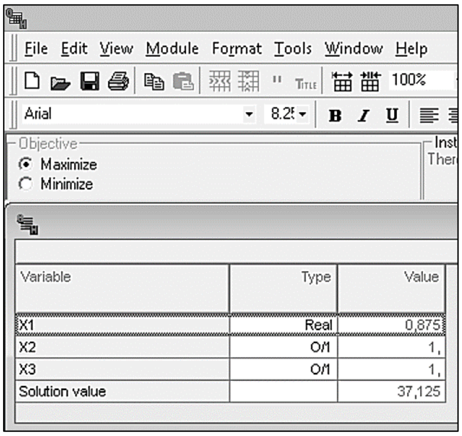
Slika 2.13.21. *Postavka zadatka kod binarnog problema (QM for Windows)*



Slika 2.13.22. *Rešenje zadatka kod binarnog problema (QM for Windows)*



Slika 2.13.23. *Postavka zadatka kod mešovitog binarnog problema (QM for Windows)*



Slika 2.13.24. *Rešenje zadatka kod mešovitog binarnog problema (QM for Windows)*

3. NELINEARNO PROGRAMIRANJE

Nelinearno programiranje je posebna metoda za rešavanje jedne posebne klase statičkih upravljačkih zadataka. Nelinearno programiranje ima za zadatak određivanja ekstremne (maksimalne ili minimalne) vrednosti funkcije cilja nad skupom ograničenja. Pri tome je se javlja nelinearnost kod funkcije cilja i/ili kod skupa ograničenja koji može biti definisan nelinearnim algebarskim jednačinama ili nejednačinama. Za rešavanje problema nelinearnog programiranja ne postoji univerzalna metoda, kao što je to slučaj kod linearnog programiranja (simpleks metod). Pri tome treba istaći da postoje i zadaci koji još uvek nisu rešeni. Do danas su razvijene mnogobrojne metode optimizacije pomoću kojih se mogu rešavati razni zadaci nelinearnog programiranja. Zbog složenosti zadataka nelinearnog programiranja, postojeće metode su specijalizovane za različite tipove zadataka koji se međusobno razlikuju po obliku matematičkog modela, tj. po obliku i dimenzijama funkcije cilja i skupa ograničenja.

Nelinearno programiranje obuhvata mnogo šire područje u odnosu na linearno programiranje. Zbog svoje strukture i svojih karakteristika, može se slobodno reći da je linearno programiranje specijalni slučaj nelinearnog programiranja.

3.1. OSNOVNI POJMOVI

- **Opšti oblik problema nelinearnog programiranja**

Opšti oblik problema nelinearnog programiranja ima sledeći oblik:

$$\begin{aligned} &min(max)\, f(x_1,\dots,x_n) \\ &g_i(x_1,\dots,x_n) \le 0, i = 1,\dots,m \end{aligned} \qquad (3.1.1.)$$

gde je

- f – funkcija cilja,
- g_i – funkcije ograničenja.

Vektorski zapis: $\mathrm{x} = (\mathrm{x}_1,\dots,\mathrm{x}_n)$

$$\begin{aligned} &min(max)\, f(x_1,\dots,x_n) \\ &g_i(x_1,\dots,x_n) \le 0, i = 1,\dots,m \end{aligned} \qquad (3.1.2.)$$

Treći (najkraći) zapis:

$$\begin{aligned} &min(max) f(x) \\ &x \in X \end{aligned} \tag{3.1.3.}$$

gde je $X = \{x \in R^n | g_i(x) \leq 0, i = 1,...,m\}$ dopustivi skup.

Ako je $X = R^n$ nelinearno programiranje se naziva problem bezuslovne optimizacije ili bezuslovnog ekstremuma.

$$\begin{aligned} &min(max) f(x) \\ &x \in R^n \end{aligned} \tag{3.1.4.}$$

Ako je $X \neq R^n$ nelinearno programiranje se naziva problem uslovne optimizacije ili uslovnog ekstremuma.

Napomena 3..1.1.: Problem maksimizacije se može svesti na problem minimizacije. Pri tome, rešava se pomoćni problem

$$\begin{aligned} &min(- f(x)) \\ &x \in X \end{aligned} \tag{3.1.5.}$$

Kada se dobije optimalno rešenje, optimalna vrednost funkcije cilja se množi sa –1 i to predstavlja rešenje problema maksimuma nelinearnog programiranja.

Napomena 3.1.2.: Dopustivi skup je bez umanjenja opštosti definisan preko ograničenja tipa $\leq$. Ograničenje $h(x) = 0$ može da se zameni sa dve nejednačine $h(x) \leq 0, h(x) \geq 0$, tj. sa nejednačinama $h(x) \leq 0, -h(x) \leq 0$, dok se ograničenje tipa $v(x) \geq u(x)$ može zapisati u obliku $u(x) - v(x) \leq 0$. Na primer, skup zadat sa

$$x_1^2 + x_2^2 = x_3 - 1, x_1 \geq x_2$$

može da se predstavi na sledeći način:

$$g_1(x) = x_1^2 + x_2^2 - x_3 + 1 \leq 0$$

$$g_2(x) = -x_1^2 - x_2^2 + x_3 - 1 \leq 0$$

$$g_3(x) = x_2 - x_1 \leq 0$$

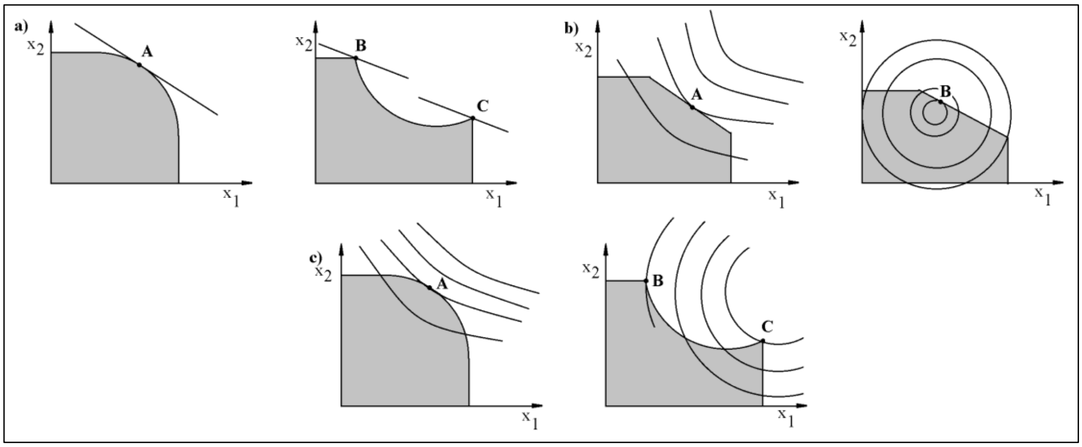
Napomena 3.1.3.: Ako u problemu učestvuju samo jednačine radi se o tzv. klasičnom problemu uslovnog ekstremuma:

$$\begin{aligned} &max(min) f(x_1,...,x_n) \\ &h_i(x_1,...,x_n) = 0, i = 1,...,m \end{aligned} \tag{3.1.6.}$$

Ovaj problem se rešava direktno, bez svođenja na ograničenja tipa $\leq$.

- **Grafički prikaz problema nelinearnog programiranja**

Grafički prikaz prikaz problema nelinearnog programiranja (kada postoje dve promenljive – x_1 i x_2) je dat na slici 3.1.1. Razmatrana su sva tri slučaja (linearna funkcija cilja i nelinearna ograničenja, nelinearna funkcija cilja i linearna ograničenja i nelinearna funkcija cilja i nelinearna ograničenja). Osenčena oblast predstavlja dopustiva rešenja problema, a tačke *A*, *B* i *C* ekstremne vrednosti funkcije cilja.



Slika 3.1.1. *Grafički prikaz problema nelinearnog programiranja. a) linerna funkcija cilja i nelinearna ograničenja; b) nelinerna funkcija cilja i linearna ograničenja; c) nelinerna funkcija cilja i nelinearna ograničenja.*

- **Dopustivo i optimalno rešenje nelinearnog programiranja**

Svako $x \in X$ se naziva *dopustivo rešenje*. Pojam optimalnosti je složeniji i obuhvata više pojmova:

– $x^* \in X$ je *globalni maksimum* problema nelinearnog programiranja ako je $f\left(x^*\right) \geq f(x)$ za sve $x \in X$.

– $x^* \in X$ je *strogi globalni maksimum* problema nelinearnog programiranja ako je $f\left(x^*\right) > f(x)$ za sve $x \in X, x \neq x^*$.

– $x^* \in X$ je *globalni minimum* problema nelinearnog programiranja ako je $f\left(x^*\right) \leq f(x)$ za sve $x \in X$.

– $x^* \in X$ je *strogi globalni minimum* problema nelinearnog programiranja ako je $f\left(x^*\right) < f(x)$ za sve $x \in X, x \neq x^*$.

Napomena 3.1.4.: Za globalni minimum i globalni maksimum se koriste i termin *globalni optimimum,* tj. *globalni ekstremum*.

Napomena 3.1.5.: Ako je f neprekidna funkcija a skup X zatvoren i ograničen postoji (bar jedan) globalni maksimum (minimum) $x^* \in X$.

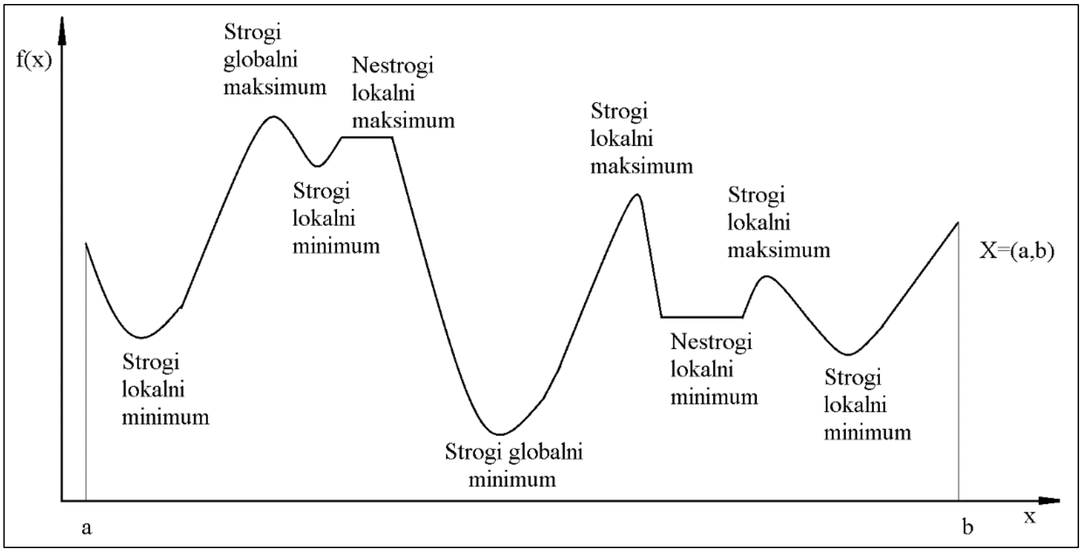
Tačka $x^* \in X$ je *lokalni maksimum* problema nelinearnog programiranja ako je najbolja u nekoj svojoj dopustivoj okolini ($\delta > 0$), pri čemu je $f(x^*) \geq f(x)$ za sve $x \in X$ takve da je $\|x - x^*\| < \delta$.

Tačka $x^* \in X$ je *lokalni minimum* problema nelinearnog programiranja ako je najbolja u nekoj svojoj dopustivoj okolini ($\delta > 0$), pri čemu je $f(x^*) \leq f(x)$ za sve $x \in X$ takve da je $\|x - x^*\| < \delta$.

Tačka $x^* \in X$ je *strogi lokalni maksimum* problema nelinearnog programiranja ako postoji $\delta > 0$ tako da je $f(x^*) > f(x)$ za sve $x \in X$ takve da je $\|x - x^*\| < \delta$ i $x \neq x^*$.

Tačka $x^* \in X$ je *strogi lokalni minimum* problema nelinearnog programiranja ako postoji $\delta > 0$ tako da je $f(x^*) < f(x)$ za sve $x \in X$ takve da je $\|x - x^*\| < \delta$ i $x \neq x^*$.

Na slici 3.1.2. dati su primeri navedenih pojmova u slučaju funkcije jedne promenljive.



Slika 3.1.2. *Vrste ekstremnih vrednosti problema nelinearnog programiranja*

- **Stanje teorije u oblasti nelinearnog programiranja**

1) Postoji karakterizacija lokalnih maksimuma/minimuma (neophodni i dovoljni uslovi za optimalnost).
2) Karakterizacija globalnih maksimuma/minimuma ne postoji, osim u specijalnim slučajevima, kao što je linearno programiranje, konveksno programiranje, itd.

3) *Metode nelinearnog programiranja* su postupci za efektivno nalaženje lokalnih maksimuma/minimuma.
4) *Metode globalne optimizacije* kombinuju metode nelinearnog programiranja sa raznim tehnikama pretraživanja dopustivog skupa u cilju nalaženja globalnih maksimuma/minimuma. Ako se radi o problemima malih dimenzija sa zatvorenim i ograničenim dopustivim skupom u principu se mogu naći svi lokalni maksimumi/minimumi i od njih odabrati najbolji. U praksi se obično radi sa problemima velikih dimenzija, tako da se problemi globalne optimizacije rešavaju uglavnom heurističkim metodama.

3.2. KLASIFIKACIJA REŠIVIH ZADATAKA NELINEARNOG PROGRAMIRANJA

Kao što je ranije navedeno, zadaci nelinearnog programiranja obuhvataju mnogo šire područje u odnosu na linearno programiranje, pri čemu tu postoje i klase zadataka koji nisu još rešeni, jer još uvek nisu razvijeni odgovarajući algoritmi za njihovo rešavanje. Iz tog razloga, u daljem tekstu su razmatrane one klase zadataka nelinearnog programiranja za koje postoje odgovarajući algoritmi za njihovo rešavanje i koji imaju veliku praktičnu primenu u praksi.

- **Nelinearno programiranje sa linearnim skupom ograničenja**

U ovoj klasi zadataka nelinearnog programiranja funkcija cilja je nelinearna funkcija, a ograničenja su predstavljena linearnim nejednačinama. Matematički model ima sledeći oblik:

$$\begin{aligned} & max(min) f(x_1,...,x_n) \\ & g_i : \sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i, (i = 1,...,m) \\ & x_j \geq 0, (j = 1,...,n) \end{aligned} \tag{3.2.1.}$$

- **Nelinearno programiranje sa separabilnom funkcijom cilja**

Kod ove klase zadataka funkcija cilja je definisana zbirom *n*-funkcija, koje su zavisne od samo jedne promenljive. Oblik separabilne funkcije cilja je:

$$F(x) = \sum_{j=1}^{n} f_j(x_j) \tag{3.2.2.}$$

- **Kvadratno programiranje**

Kod ove klase zadataka nelinearnog programiranja funkcija cilja je zadata u obliku kvadratne forme:

$$F(x)=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}\cdot x_i\cdot x_j \qquad (3.2.3.)$$

Specijalnu podklasu zadataka kvadratnog programiranja čine zadaci kod kojih je skup ograničenja linearan.

- **Celobrojno nelinearno programiranje**

Kod ove klase zadataka nelinearnog programiranja sve promenljive mogu uzimati samo celobrojne vrednosti. Specijalan slučaj ove klase zadataka predstavljaju oni zadaci kod kojih promenljive mogu uzimati samo dve vrednosti: nula i jedan – binarno programiranje ili 0-1 programiranje.

3.3. METODE REŠAVANJA ZADATAKA NELINEARNOG PROGRAMIRANJA

Sve metode rešavanja zadataka nelinearnog programiranja se mogu uopšteno podeliti na:

- analitičke metode,
- numeričke metode,
- grafičke metode, i
- eksperimentalne metode

Analitičke metode se primenjuju se za rešavanje problema kod kojih funkcija cilja ima takav oblik da je moguće primeniti poznata matematička pravila pri postupanju sa varijablama i funkcijama ograničenja. Nisu pogodne za rešavanje složenijih problema nelinearnog programiranja. Postupak analitičkih metoda se bazira na pronalaženju one vrednosti promenljivih, odnosno vektora X za koju je prvi izvod odnosno gradijent funkcije cilja $f(X)$ jednaka nuli. Ta tačka se naziva kritična tačka. Nakon toga proverava se predznak drugog izvoda funkcije cilja u toj tački i zavisno od predznaka zaključuje se da li se radi o minimumu ili maksimumu. Funkcije cilja mogu imati više tačaka u kojima je gradijent funkcije cilja jednak nuli tako da se optimumi dele na globalne i lokalne optimume. Da bi se odredilo koji je od optimuma globalni optimum odnosno stvarno najbolje rešenje, potrebno je izračunati vrednosti funkcije cilja u svim kritičnim tačkama i uporediti ih, nakon čega se određuje stvarno rešenje.

Numeričke metode se zasnivaju na nizu neprestanih poboljšanja aproksimacija optimalnog rešenja sledeći naredne korake koji su načelno isti za probleme sa i bez ograničenja.

Geometrijske metode su pogodne za rešavanje problema sa jednom promenljivom. Moguće je rešavati probleme i sa dve promenljive, ali to nije praktično zbog toga što je potrebno crtati u trodimenzionalnom prostoru pa se ne koristi često.

Eksperimentalne metode se baziraju na eksperimentima pomoću kojih se pokušava doći do ekstrema funkcija odnosno do optimuma. Kada se izvrši jedan eksperiment, dobijeni rezultati su ključni za određivanje gde treba locirati sledeći eksperiment koji će imati bolje rezultate funkcije cilja od prethodnog eksperimenta.

U okviru gore navedenih grupa metoda postoji veliki broj pojedinačnih metoda za rešavanje problema nelinearnog programiranja. U narednom tekstu akcenat je na najznačajnijim metodama, koje imaju najveći praktični značaj.

3.3.1. Bezuslovna optimizacija. Neophodni i dovoljni uslovi optimalnosti

Dat je problem

$$\begin{aligned} &min(max)\, f(x) \\ &x \in R^n \end{aligned}$$

Pretpostavlja se da je f dvaput neprekidno diferencijabilna na R^n.

Rešavanje zadatka počinje određivanjem stacionarnih tačaka koje predstavljaju moguća rešenja problema i to na osnovu sledećeg izraza:

$$\frac{\partial f}{\partial x_1} = 0, \ldots, \frac{\partial f}{\partial x_n} = 0 \qquad (3.3.1.)$$

Nakon toga se traži matrica drugih izvoda:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \qquad (3.3.2.)$$

Neka je x^* rešenje sistema (3.3.1.). Računa se $\nabla^2 f(x^*)$ i određuju se glavni minori $D_1, \ldots, D_n$.

Ako je $D_1>0$, $D_2>0, \ldots, D_n>0$ sledi da je x^* strogi lokalni minimum.

Ako je $D_1<0$, $D_2>0, \ldots, (-1)^n D_n>0$ sledi da je x^* strogi lokalni maksimum.

Teorema 3.3.1. (Neophodni uslovi za lokalni minimum). Ako je x^* lokalni minimum funkcije f, onda je $\frac{\partial f}{\partial x_1}(x^*) = 0, \ldots, \frac{\partial f}{\partial x_n}(x^*) = 0 \qquad (\nabla f(x^*) = 0)$.

Teorema 3.3.2. (Dovoljni uslovi za strogi lokalni minimum). Ako je $\nabla f(x^*) = 0$ i ako je matrica $\nabla^2 f(x^*) = 0$ *pozitivno definitna* (tj. $y^T \nabla^2 f(x^*) y > 0, \quad \forall y \neq 0$), onda je x^* je strogi lokalni minimum funkcije *f*.

Teorema 3.3.3. (Silvestrov kriterijum). $\nabla^2 f(x^*) = 0$ je pozitivno definitna matrica $\Leftrightarrow$ $D_1 > 0, D_2 > 0, \ldots, D_n > 0$.

Primer 3.3.1. Odrediti ekstremne vrednosti funkcije sa dve promenljive:

$$f(x) = x_1^3 + x_2^3 - x_1^2 - 2x_1x_2 - x_2^2$$

Rešenje. Prvo se određuju stacionarne tačke..

$$\left.\begin{aligned} \frac{\partial f}{\partial x_1} &= 3x_1^2 - 2x_1 - 2x_2 = 0 \\ \frac{\partial f}{\partial x_2} &= 3x_2^2 - 2x_2 - 2x_1 = 0 \end{aligned}\right\} \Rightarrow 3x_1^2 = 3x_2^2 \Rightarrow x_1 = x_2$$

$3x_1^2 - 4x_1 = 0$

$x_1(3x_1 - 4) = 0$

$x_1 = 0$ i $x_1 = 4/3$

Stacionarne tačke su $A(0, 0)$ i $B(4/3, 4/3)$.

Formira se matrica drugih izvoda.

$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1 - 2; \qquad \frac{\partial^2 f}{\partial x_1 \partial x_2} = -2; \qquad \frac{\partial^2 f}{\partial x_2 \partial x_1} = -2; \qquad \frac{\partial^2 f}{\partial x_2^2} = 6x_2 - 2$$

$$\nabla^2 f(x) = \begin{bmatrix} 6x_1 - 2 & -2 \\ -2 & 6x_2 - 2 \end{bmatrix}$$

Određuju se glavni minori za obe stacionarne tačke:

$A(0,0) \qquad \nabla^2 f(x_A) = \begin{bmatrix} -2 & -2 \\ -2 & -2 \end{bmatrix}$, $D_1 = -2$, $D_2 = 0$ – bez zaključka.

$B(4/3, 4/3) \qquad \nabla^2 f(x_B) = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}$, $D_1 = 6$, $D_2 = 32$ – strogi lokalni minimum.

Funkcija ima strogi lokalni monimum u tački *B*.

3.3.2. Klasični problem uslovnog ekstremuma

Dat je problem

$$min(max) f(x)$$
$$h_i(x) = 0, i = 1,\ldots,m$$

Pretpostavlja se da su f, h_i, $i = 1,\ldots,m$ diferencijabilne.

Onda je rang $J(x) = m$ na dopustivom skupu.

$$J(x)=\begin{bmatrix} \dfrac{\partial h_1}{\partial x_1} & \cdots & \dfrac{\partial h_1}{\partial x_n} \\ \vdots & & \\ \dfrac{\partial h_m}{\partial x_1} & \cdots & \dfrac{\partial h_m}{\partial x_n} \end{bmatrix}_{mxn} \qquad (3.3.3.)$$

3.3.2.1. Metoda eliminacije promenljivih

Ovde je ideja da se klasični problem optimizacije svede na problem bezuslovne optimizacije.

$$\begin{matrix} h_1(x_1,...,x_n)=0 & & x_1=\varphi_1(x_{m+1},...,x_n) \\ \vdots & \Rightarrow & \vdots \\ h_m(x_1,...,x_n)=0 & & x_m=\varphi_m(x_{m+1},...,x_n) \end{matrix} \qquad (3.3.4.)$$

Dobija se problem bezuslovne optimizacije:

$$min\, f(\varphi_1(x_{m+1},...,x_n),...,\varphi_m(x_{m+1},...,x_n))=F(x_{m+1},...,x_n) \qquad (3.3.5.)$$

Ako je $(x^*_{m+1},...,x^*_n)$ rešenje bezuslovne optimizacije, onda je $\left(\varphi_1\left(x^*_{m+1},...,x^*_n\right),...,\varphi_m\left(x^*_{m+1},...,x^*_n\right)\right)$ rešenje zadatka klasičnog problema optimizacije.

Primer 3.3.2. Metodom eliminacije rešiti zadatak:

$$minf(\mathrm{x}) = 4x_1^2 + 6x_2 + 5x_3^2$$
$$x_1 + x_2 + x_3 = 8$$

Rešenje: Ovde je $J = [1\ 1\ 1]$, rang $J = 1$. Eliminiše se jedna od promenljivih, na primer x_2. Sledi:

$$x_2 = 8 - x_1 - x_3$$

Tako da je:

$$f(x_1,x_3) = 4x_1^2 + 6(8 - x_1 - x_3) + 5x_3^2$$

$$\frac{\partial f}{\partial x_1}=8x_1-6=0\Rightarrow x_1=3/4; \qquad \frac{\partial f}{\partial x_3}=10x_3-6=0\Rightarrow x_2=3/5$$

Jedina stacionarna tačka je $x^* = (3/4, 3/5)$. Kako je

$$\frac{\partial^2 f}{\partial x_1^2}=8; \qquad \frac{\partial^2 f}{\partial x_1\partial x_3}=0; \qquad \frac{\partial^2 f}{\partial x_3\partial x_1}=0; \qquad \frac{\partial^2 f}{\partial x_3^2}=10$$

$$\nabla^2 f(x^*)=\begin{bmatrix} 8 & 0 \\ 0 & 10 \end{bmatrix} \quad D_1 = 8 > 0,\ D_2 = 80 > 0$$

sledi da je x^* strogi lokalni minimum, to jest da je tačka $x^* = (3/4, 133/20, 3/5)$ strogi lokalni minimum polaznog problema.

3.3.2.2. Metoda Lagranžovih množilaca

Za rešavanje zadataka kod ove metode se koristi Lagranžova funkcija koja ima sledeći oblik:

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) \qquad (3.3.6.)$$

gde se $\lambda = (\lambda_1,...,\lambda_m)$ naziva vektor Lagranžovih množilaca.

Traži se rešenje sistema:

$$\frac{\partial L}{\partial x_1} = 0,...,\frac{\partial L}{\partial x_n} = 0,\frac{\partial L}{\partial \lambda_1} = 0,...,\frac{\partial L}{\partial \lambda_m} = 0 \qquad (3.3.7.)$$

Rešenja izraza (3.3.7.) su stacionarne tačke Lagranžove funkcije. U daljem postupku se traži matrica:

$$H(x,\lambda) = \begin{bmatrix} 0 & \cdots & 0 & \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & & & \vdots & & \\ 0 & \cdots & 0 & \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \\ \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_1} & \frac{\partial^2 L}{\partial x_1^2} & \cdots & \frac{\partial^2 L}{\partial x_1 \partial x_n} \\ \vdots & & & \vdots & & \\ \frac{\partial h_1}{\partial x_n} & \cdots & \frac{\partial h_m}{\partial x_n} & \frac{\partial^2 L}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 L}{\partial x_n^2} \end{bmatrix}_{(m+n)x(m+n)} = \begin{bmatrix} 0 & J \\ J^T & \nabla_{xx}^2 L \end{bmatrix} \qquad (3.3.8.)$$

Neka je (x^*, λ^*) rešenje sistema i ako su $D_1,\ldots,D_{m+n}$ glavni minori matrice $H(x^*, \lambda^*)$, kao i $(-1)^m D_{2m+1} > 0,\ldots, (-1)^m D_{m+n} > 0$ onda je x^* strogi lokalni minimum polaznog problema. Ako je $(-1)^{m+1} D_{2m+1} > 0,\ldots, (-1)^n D_{m+n} > 0$ onda je x^* strogi lokalni maksimum polaznog problema.

Teorema 3.3.4. (Neophodni uslovi za lokalni minimum). Ako je x^* lokalni minimum problema, sledi $\exists \lambda^* = (\lambda_1^*,...,\lambda_m^*)$ tako da je $\nabla_x L(x^*,\lambda^*)) = 0, \nabla_\lambda L(x^*,\lambda^*) = 0$, tj. $\nabla L(x^*,\lambda^*)) = 0$.

Teorema 3.3.5. (Dovoljni uslovi za strogi lokalni minimum). Neka je (x^*, λ^*) stacionarna tačka funkcije L. Ako je matrica $\nabla_{xx}^2 L(x^*,\lambda^*)) = 0$ *pozitivno definitna na tangentnom prostoru* $y^T \nabla_{xx}^2 L(x^*,\lambda^*) y > 0 \quad \forall y \neq 0$ takvo da je $y^T \nabla h_i(x^*) = 0, i = 1,...,m$ tada je x^* strogi lokalni minimum klasičnog problema.

Teorema 3.3.6. (Dovoljni uslovi za pozitivnu definitnost na tangentnom prostoru). Ako je $(-1)^m D_{2m+1} > 0,\ldots, (-1)^m D_{m+n} > 0 \Rightarrow \nabla_{xx}^2 L(x^*,\lambda^*)$ je pozitivno definitna matrica na tangentnom prostoru.

Napomena 3.3.1.: Svi navedeni rezultati važe pod pretpostavkom da je rang $J=m$. Ako u nekim tačkama dopustivog skupa ovaj uslov ne važi, njih treba posebno ispitati jer su i one kandidati za ekstremum.

Primer 3.3.3. Metodom Lagranžovih množitelja rešiti zadatak:

$$f(x) = 2x_1^2 + 2x_2 + 2x_3^2$$
$$x_1 + x_2 + x_3 = 2$$

Rešenje: Lagranžova funkcija ima sledeći oblik:

$$L = 2x_1^2 + 2x_2 + 2x_3^2 + \lambda(x_1 + x_2 + x_3 - 2)$$

Izvodi po x_j i njihove vrednosti iznose:

$$\left.\begin{aligned} \frac{\partial L}{\partial x_1} &= 4x_1 + \lambda = 0 \\ \frac{\partial L}{\partial x_2} &= 4x_2 + \lambda = 0 \\ \frac{\partial L}{\partial x_3} &= 4x_3 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= x_1 + x_2 + x_3 - 2 \end{aligned}\right\} \Rightarrow x_1 = x_2 = x_3 = \frac{2}{3}, \lambda = -\frac{4}{3}, x^* = \left(\frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right), \lambda^* = -\frac{4}{3}$$

Drugi izvodi po x iznose:

$$\nabla_{xx}^2 L(x,\lambda) = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix} \qquad J = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Matrica H ima sledeći oblik:

$$H(x,\lambda) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ 1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix} = H(x^*, \lambda^*) \qquad m = 1, \qquad n = 3, \qquad 2m+1 = 3, \qquad m+n = 4m$$

Glavni minori matrice H iznose:

$$(-1)^m D_{2m+1} = (-1)D_3 = -\begin{vmatrix} 0 & 1 & 1 \\ 1 & 4 & 0 \\ 1 & 0 & 4 \end{vmatrix} = 8 > 0$$

$$(-1)^m D_{m+n} = -D_4 = -\begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ 1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 4 \end{vmatrix} = 48 > 0$$

Sledi da je x^* strogi lokalni minimum.

3.3.3. Opšti slučaj problema nelinearnog programiranja

Opšti slučaj se može izraziti na sledeći način:

$$\begin{aligned} &min(max) f(x) \\ &g_i(x) = 0, i = 1,...,m \end{aligned}$$

3.3.3.1. *Metoda izravnajućih funkcija*

Opšti slučaj problema nelinearnog programiranja se rešava tako što se dodavanjem nenegativnih *izravnavajućih funkcija* $x_{n+i}^2, i = 1,...,m$ on svodi na klasični problem uslovnog ekstremuma:

$$\begin{aligned} &min(max) f(x) \\ &g_i(x) + x_{n+i}^2 = 0, i = 1,...,m \end{aligned} \tag{3.3.9.}$$

Ovako modifikovani klasični problem se dalje može rešavati metodom eliminacije promenljivih ili metodom Lagranžovih množilaca.

Primer 3.3.4. Rešiti zadatak:

$$\begin{aligned} &min\ 5x_1^2 + 6x_2^2 \\ &2 - x_1 \le 0 \end{aligned}$$

Rešenje: Dodaje se izravnjajuća funkcija, pa problem dobija sledeći oblik:

$$\begin{aligned} &min\ 5x_1^2 + 6x_2^2 \\ &2 - x_1 + x_3^2 = 0 \end{aligned}$$

Sledi da je $x_1 = 2 + x_3^2$. Funkcija cilja dobija sledeći oblik:

$$min\ 5(2 + x_3^2)^2 + 6x_2^2 = F(x_2, x_3)$$

Određuju se izvodi po x:

$$\left.\begin{aligned} &\frac{\partial F}{\partial x_2} = 12x_2 = 0 \\ &\frac{\partial F}{\partial x_3} = 20x_3\left(2 + x_3^2\right) = 0 \end{aligned}\right\} \Rightarrow x_2 = x_3 = 0$$

Matrica drugog diferencijala iznosi:

$$\nabla^2 F(x_2, x_3) = \begin{bmatrix} 12 & 0 \\ 0 & 20\left(2 + x_3^2\right) + 20x_3 \cdot 2x_3 \end{bmatrix} \qquad \nabla^2 F(0,0) = \begin{bmatrix} 12 & 0 \\ 0 & 40 \end{bmatrix}$$

Glavni minori su: $D_1 = 12 > 0$, $D_2 = 480 > 0$. Sledi da je tačka (0,0) strogi lokalni minimum problema bezuslovne optimizacije. Takođe, tačka sa koordinatama (2,0,0) je strogi lokalni minimum klasičnog problema nelinearnog programiranja, a tačka (2,0) je strogi lokalni minimum opšteg problema nelinearnog programiranja.

3.3.3.2. Kun – Takerova teorema

Problem nelinearnog programiranja se može izraziti na sledeći način:

$$min(max) f(x)$$
$$g_i(x) \leq 0, i = 1,...,m$$

Pretpostavlja se da su funkcije f, $g_1,\ldots,g_m$ diferencijabilne.

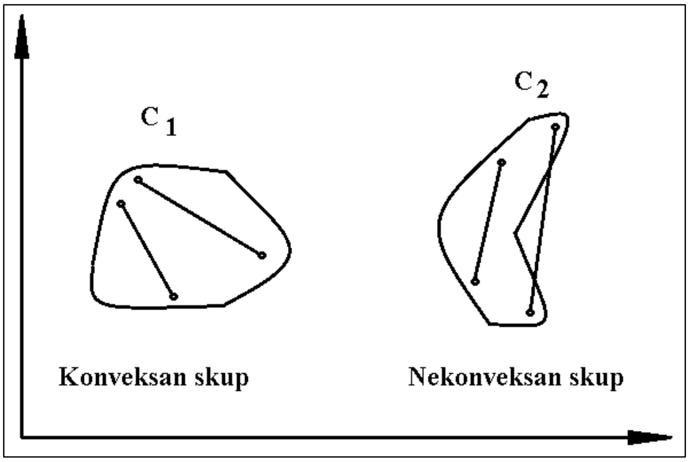
Ako su funkcije f, $g_1,\ldots,g_m$ konveksne problem nelinearnog programiranja se naziva *problem konveksnog programiranja.* Kod problema konveksnog programiranja postoji važno svojstvo da je svaki lokalni minimum istovremeno globalni minimum.

Kun-Takerova teorema daje neophodne uslove za lokalni minimum problema nelinearnog programiranja. Kada se radi o problemu konveksnog programiranja ti uslovi su i dovoljni. Takođe, Kun-Takerova teorema važi pod odredjenim uslovima regularnosti.

U daljem tekstu se najpre objašnjavaju ključni pojmovi vezani za konveksnost i regularnost.

- **Konveksnost**

Konveksan skup se može izraziti na sledeći način: $\lambda x_1 + (1-\lambda)x_2 \in C \quad \forall x_1, x_2 \in C \quad \forall \lambda \in [0.1]$. Grafički prikaz je dat na slici 3.3.1. Kod konveksnog skupa svake dve tačke datog skupa sadrži duž koja ih spaja i koja se nalazi u tom skupu.

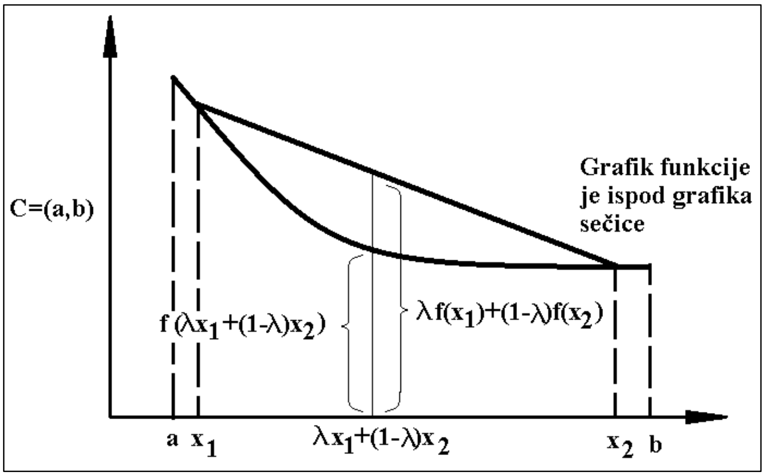


Slika 3.3.1. *Konveksan i nekonveksan skup*

Konveksna funkcija ima sledeći oblik:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \qquad \forall x_1, x_2 \in C \qquad \forall \lambda \in [0.1] \quad (3.3.10.)$$

Grafik funkcije se nalazi ispod sečice – slika 3.3.2.



Slika 3.3.2. *Konveksna funkcija*

Strogo konveksna funkcija ima sledeći oblik:

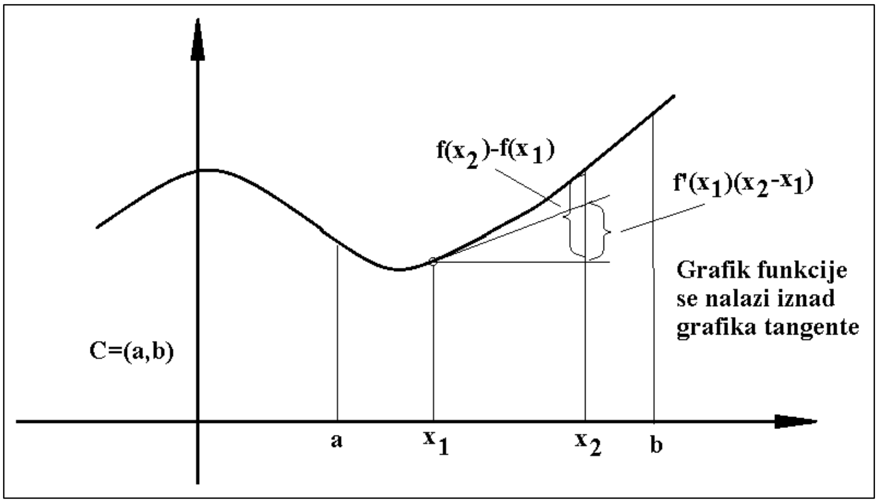
$$f(\lambda x_1+(1-\lambda)x_2)<\lambda f(x_1)+(1-\lambda)f(x_2), \forall x_1,x_2\in C \qquad x_1\neq x_2 \qquad \forall\lambda\in[0.1] \tag{3.3.11.}$$

Funkcija g je *konkavna* ako je njena negativna vrednost ($-g$) konveksna funkcija.

Ako je f diferencijabilna funkcija, onda se njen grafik se nalazi ispod sečice, odnosno grafik se nalazi iznad tangente.

Teorema 3.3.7. funkcija f je konveksna ako važi sledeća nejednakost (slika 3.3.3.):

$$\nabla f(x_1)^T(x_2-x_1)\leq f(x_2)-f(x_1) \qquad \forall x_1,x_2\in C \tag{3.3.12.}$$



Slika 3.3.3. *Konveksnost diferencijabilne funkcije na odseku (a, b)*

Ako je funkcija dvaput diferencijabilna onda je najlakše proveriti konveksnost funkcije preko matrice drugih izvoda na sledeći način:

1. Naći glavne minore matrice $\nabla^2 f(x)$. Ako su svi pozitivni na *C*, sledi da je funkcija *f* strogo konveksna na *C*.
2. Ako je su svi glavni minori nenegativni a bar jedan je jednak nuli treba naći sve minore simetrične u odnosu na glavnu dijagonalu. Ako su svi simetrični minori nenegativni na *C*, sledi da je funkcija *f* konveksna na *C*.

- **Regularnost**

Najpoznatija su sledeća dva uslova regularnosti:

R1 (Sleterov uslov). Važi ako su kod nelinearnog programiranja funkcije ograničenja $g_1,\ldots,g_m$ konveksne i postoji $\hat{x}$ tako da je $g_i(\hat{x}) < 0, i = 1,\ldots,m$.

R2. Važi u tački $\hat{x}$ ako su $\nabla g_i(\hat{x}), i \in I(\hat{x})$ linearno nezavisni, gde je $I(\hat{x}) = \{i | g_i(\hat{x}) = 0\}$ skup indeksa aktivnih ograničenja u $\hat{x}$.

Napomena 3.3.2.: Uslov R1 važi ili ne za problem u celini, a uslov R2 može važiti u nekim tačkama, a u nekim ne.

- **Kun – Takerova teorema – neophodni uslovi optimalnosti**

Neka je x^* lokalni minimum nelinearnog problema i neka važi R1 ili u x^* važi R2. Tada postoji $\lambda^* \in R^m$ tako da važi:

$$\begin{aligned}
&\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0 \\
&\lambda_i^* g_i(x^*) = 0, \quad i = 1,\ldots,m \\
&\lambda^* \geq 0 \\
&g_i(x^*) \leq 0, \quad i = 1,\ldots,m
\end{aligned} \tag{3.3.13.}$$

Ako važi R1 ili u svim dopustivim tačkama važi R2, onda se svi kandidati za lokalni minimum nalaze među rešenjima sistema, odnosno:

$$\begin{aligned}
&\nabla f(x\ ) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x\ ) = 0 \\
&\lambda_i g_i(x\ ) = 0, \quad i = 1,\ldots,m \\
&\lambda\ \geq 0 \\
&g_i(x\ ) \leq 0, \quad i = 1,\ldots,m
\end{aligned} \tag{3.3.14.}$$

Napomena 3.3.3.: λ^* se naziva vektor *Lagranžovih množilaca.*

Ukoliko su sva ograničenja kod problema nelinearnog programiranja linearna mogu se izostaviti uslovi regularnosti.

Teorema 3.3.8. (neophodni uslovi u slučaju linearnih ograničenja). Ako je kod problema nelinearnog programiranja $g_i(x) = a_i^T x - b_i$, i=1,...,m i ako je x^* lokalni minimum, tada postoji $\lambda^* \in R^m$ tako da važi (3.3.13.).

U konveksnom slučaju uslovi (3.3.13.) su istovremeno i dovoljni za optimalnost.

Teorema 3.3.9. (dovoljni uslovi u konveksnom slučaju). Neka su funkcije f, g_1,…, g_m konveksne. Ako u tački (x^*, λ^*) važe uslovi (3.3.13.) tada je x^* globalni minimum problema nelinearnog programiranja.

- **Kun – Takerova teorema – neophodni uslovi optimalnosti za problem zadat nejednačinama i jednačinama**

Ova varijanta Kun-Takerove teoreme važi u slučaju da je dopustivi skup problema nelinearnog programiranja zadat nejednačinama $g_i(x) \le 0$, i=1,...,m i jednačinama $h_i(x) = 0$, i=1,...,r.

Neka je x^* lokalni minimum problema nelinearnog programiranja i neka su vektori $\nabla g_i(x^*), i \in I(x^*), \nabla h_i(x^*), i = 1,...,r$ linearno nezavisni, tada postoje $\lambda^* \in R^m, \psi^* \in R^r$ tako da važi:

$$
\begin{aligned}
&\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^{r} \psi_i^r \nabla h_i(x^*) = 0 \\
&\lambda_i^* g_i(x^*) = 0, \quad i = 1,...,m \\
&\lambda^* \ge 0 \\
&g_i(x^*) \le 0, \quad i = 1,...,m, \quad h_i(x^*) = 0, \quad i = 1,...,r
\end{aligned}
\tag{3.3.15.}
$$

Napomena 3.3.4.: Lagranžovi množioci koji odgovaraju jednačinama nisu ograničeni po znaku. Uslov $\psi_i^* h_i(x^*) = 0$, i=1,...,r je trivijalno ispunjen.

3.3.3.3. Metode kaznenih funkcija

Polazi se od problema nelinearnog programiranja:

$$
\begin{gathered}
min(max) f(x) \\
g_i(x) \le 0, i = 1,...,m
\end{gathered}
$$

Metoda kaznenih funkcija se zasniva na uvođenju beskonačne kazne za napuštanje dopustivog skupa X, odnosno

$$q(x) = \begin{cases} 0 & x \in X \\ +\infty & x \notin X \end{cases} \tag{3.3.16.}$$

Problem se može rešiti tako što se menja problemom bezuslovne optimizacije:

$$\begin{aligned} &\min F(x) = f(x) + q(x) \\ &x \in R^n \end{aligned} \tag{3.3.17.}$$

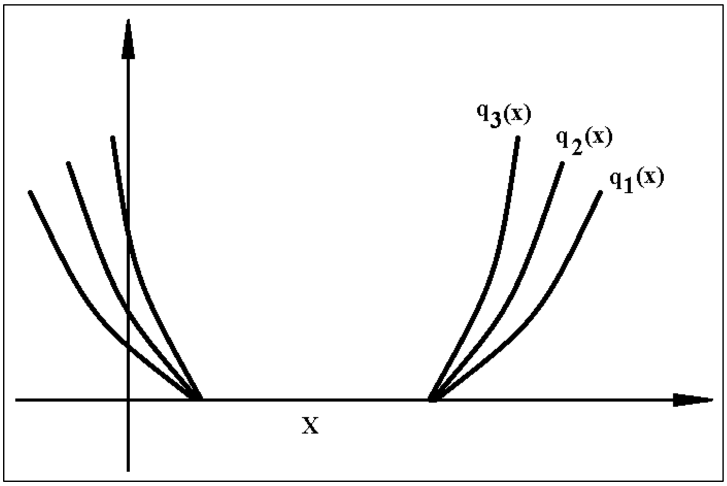
Pošto se ne može računati sa ∞, funkcija $q(x)$ se aproksimira nizom kaznenih funkcija. Postoje dve vrste kaznenih funkcija – spoljašnje (aproksimacija spolja) i unutrašnje (aproksimacija iznutra).

- **Metoda spoljašnjih kaznenih funkcija**

Spoljašne kaznene funkcije problema nelinearnog programiranja predstavljaju niz funkcija $q_k : R^n \to R$, k = 1, 2,... ako za svako k važi:

$$\begin{aligned} &q_k(x) = 0, \quad x \in X \\ &q_k(x) > 0, \quad x \notin X \\ &q_{k+1}(x) > q_k(x), \quad x \notin X \\ &q_k(x) \to \infty, \quad k \to \infty, \quad x \notin X \end{aligned} \tag{3.3.18.}$$

Na slici 3.3.4. data je geometrijska interpretacija niza spoljašnjih kaznenih funkcija $\{q_k(x)\}$.



Slika 3.3.4. *Niz spoljašnjih kaznenih funkcija*

Na ovaj način, problemu nelinearnog programiranja se pridružuje niz problema:

$$\begin{aligned} &\min F_k(x) = f(x) + q_k(x) \\ &x \in R^n \end{aligned} \tag{3.3.19.}$$

Ako je x^k rešenje problema (3.3.19.), pod odgovarajućim pretpostavkama se može pokazati da $x^k \to x^*$, $k \to \infty$, gde je x^* rešenje problema nelinearnog programiranja.

Napomena 3.3.5.: Ako za neko k važi $x^k \in X$ tada je x^k rešenje problema nelinearnog programiranja, tj. $x^* = x^k$.

Algoritam za rešavanje zadataka nelinearnog programiranja metodom spoljašnjih kaznenih funkcija je sledeći:

- ***Korak 0:*** Izabrati niz $\{q_k\}$,
- ***Korak 1:*** $k = 1$,
- ***Korak 2:*** Rešiti (3.3.19.) i označiti rešenje sa x^k,
- ***Korak 3:*** Ako $x^k \in X$ STOP ($x^* = x^k$), u suprotnom $k = k + 1$, ići na Korak 2.

Izbor niza spoljašnjih kaznenih funkcija vrši se najčešće na sledeći način:

$$q_k(x) = t_k \sum_{i=1}^{m} \max\{0, g_i(x)\}^2 \tag{3.3.20.}$$

gde je $\{t_k\}$ monotono rastući niz ($t_k \to \infty, k \to \infty$).

Ako u problemu nelinearnog programiranja učestvuju i jednačine $h_i(x) = 0$, $i = 1,...,r$ izbor niza spoljašnjih kaznenih funkcija vrši se na sledeći način:

$$q_k(x) = t_k \left(\sum_{i=1}^{m} \max\{0, g_i(x)\}^2 + \sum_{i=1}^{r} (h_i(x))^2 \right) \tag{3.3.21.}$$

Primer 3.3.5. Rešiti zadatak metodom spoljašnjih kaznenih funkcija:

$$\begin{aligned} &\min x^2 - 4x \\ &x \le 1 \end{aligned}$$

Rešenje: Neka je $q_k(x) = t_k \max\{0, x-1\}^2$ i neka $t_k \to \infty$, $k \to \infty$. Tada je

$$F_k(x) = x^2 - 4x + t_k \max\{0, x-1\}^2$$

Traži se x^k kao rešenje problema bezuslovne minimizacije funkcije $F_k(x)$. Problem se rešava analitički. Razlikuju se dva slučaja:

1. $\max\{0, x-1\} = 0$, tj. $x - 1 \le 0$. Sledi

$$F_k(x) = x^2 - 4x, \quad F_k'(x) = 2x - 4 \Rightarrow x = 2$$

Pošto je $x = 2 \ge 1$, odbacuje se ovo rešenje.

2. $max\{0, x-1\} = x-1$, tj. $x - 1 \geq 0$. Sledi

$$F_k(x) = x^2 - 4x + t_k(x-1)^2, \quad F_k'(x) = 2x - 4 + 2t_k(x-1) = 0$$

odakle je $x = \dfrac{2t_k + 4}{2t_k + 2}$.Kako ovo rešenje zadovoljava uslov $x \geq 1$, ono se usvaja. tj. $x^k = \dfrac{2t_k + 4}{2t_k + 2}$.

Pošto je $\lim\limits_{k \to \infty} t_k = \infty$ rešenje se dobija rešavanjem sledećeg izraza:

$$\lim_{k \to \infty} x_k = \lim_{k \to \infty} \frac{2 + \dfrac{6}{t_k}}{2 + \dfrac{2}{t_k}} = 1$$

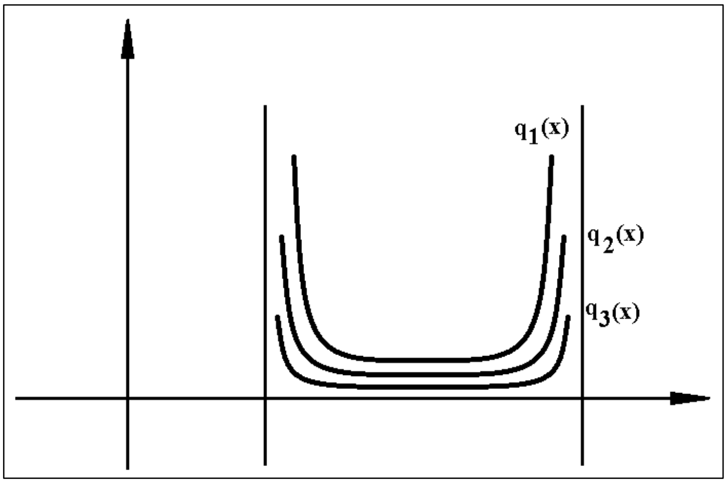
Pošto je $x = 1$, to predstavlja rešenje zadatka jer je ispunjen uslov ograničenja da je $x \leq 1$.

- **Metoda unutrašnjih kaznenih funkcija**

Unutrašnje kaznene funkcije problema nelinearnog programiranja predstavljaju niz funkcija $q_k(x) : \overset{o}{X} \to R$, $k = 1, 2,...$ pri čemu je $\overset{o}{X}$ unutrašnjost dopustivog skupa X (oko svake tačke u $\overset{o}{X}$ postoji sferna okolina koja je čitava sadržana u $\overset{o}{X}$), a ∂X njegova granica ($\partial X = X / \overset{o}{X}$). Za niz unutrašnjih kaznenih funkcija za problem nelinearnog programiranja važe sledeći izrazi:

$$\begin{aligned}
&|q_{k+1}(x)| < |q_k(x)|, \quad x \in \overset{o}{X} \\
&q_k(x) \to 0, \quad k \to \infty, \quad x \in \overset{o}{X} \\
&q_k(x_j) \to \infty, \quad j \to \infty, \quad za \quad svaki \quad niz \\
&\{x_j\} \subset \overset{o}{X} \quad takav \quad da \quad x_j \to \hat{x} \in \partial X, \quad j \to \infty
\end{aligned} \qquad (3.3.22.)$$

Na slici 3.3.5. data je geometrijska interpretacija niza unutrašnjih kaznenih funkcija $\{q_k\}$.



Slika 3.3.5. *Niz unutrašnjih kaznenih funkcija*

Problemu nelinearnog programiranja se pridružuje niz problema

$$\begin{aligned} &\min F_k(x) = f(x) + q_k(x) \\ &x \in \overset{o}{X} \end{aligned} \qquad (3.3.23.)$$

Neka je sa x^k označeno rešenje problema (3.3.23.). Pod odgovarajućim pretpostavkama se opet može pokazati da $x^k \to x^*$, $k \to \infty$, gde je x^* rešenje problema nelinearnog programiranja.

Napomena 3.3.6.: (3.3.23.) se može rešavati približnim metodama bezuslovne optimizacije jer se izlazak iz dopustivog skupa sprečava "barijerom" jer kaznena funkcija raste pri približavanju granici.

Algoritam za rešavanje zadataka nelinearnog programiranja metodom unutrašnjih kaznenih funkcija je sledeći:

- ***Korak 0:*** Izabrati niz $\{q_k(x)\}$,
- ***Korak 1:*** $k = 1$,
- ***Korak 2:*** Rešiti (3.3.23.) i označiti rešenje sa x^k,
- ***Korak 3:*** Ako $x^k \in X$ STOP ($x^* = x^k$), u suprotnom $k = k + 1$, ići na Korak 2.

Izbor niza unutrašnjih kaznenih funkcija vrši se najčešće na sledeći način:

$$q_k(x) = -\frac{1}{t_k}\sum_{i=1}^{m}\ln(-g_i(x)) \qquad (3.3.24.)$$

gde je $\{t_k\}$ monotono rastući niz ($t_k \to \infty, k \to \infty$).

Ako u definiciji problema nelinearnog programiranja učestvuju i jednačine metoda unutrašnjih kaznenih funkcija, metoda nije primenljiva osim u slučaju kada se radi o linearnim jednačinama (na primer, u slučaju linearnog programiranja).

Primer 3.3.6. Razmatra se zadatak 3.3.5. za rešavanje metodom unutrašnjih kaznenih funkcija:

$$\min x^2 - 4x$$
$$x \le 1$$

Rešenje: Neka je $q_k(x) = -\frac{1}{t_k}\ln(1-x)$. Tada je

$$F_k(x) = x^2 - 4x - \frac{1}{t_k}\ln(1-x), \qquad \overset{o}{X} = (-\infty, 1)$$

Problem se rešava analitički.

$$F_k'(x) = 2x - 4 + \frac{1}{t_k(1-x)} = 0$$

Kada se sredi prethodni izraz, dobija se kvadratna jednačina:

$$2t_k x^2 - 6t_k x + 4t_k - 1 = 0$$

odakle je

$$x_{1,2} = \frac{6t_k \pm \sqrt{4t_k^2 + 8t_k}}{4t_k} = \frac{6 \pm \sqrt{4 + 8/t_k}}{4}$$

S obzirom da je $8/t_k = 0$, ($t_k \to \infty$, $k \to \infty$), rešenje iznosi:

$$x_1 = 2$$
$$x_2 = 1$$

Pošto važi $x \le 1$, x_1 ne ispunjava uslov i on se odbacuje. Rešenje predstavlja x_2 ($x^k = x_2$).

3.3.4. Približne metode za nelinearno programiranje

3.3.4.1. Bezuslovna optimizacija

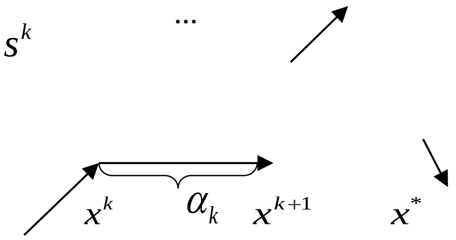
Problem nelinearnog programiranja je sledeći:

$$\min(\max) f(x)$$
$$x \in R^n$$

Metode bezuslovne optimizacije generišu niz $\{x^k\}$ po formuli:

$$x^{k+1} = x^k + \alpha_k s^k, \qquad k = 0,1,... \qquad (3.3.25.)$$

gde je $\{s^k\}$ niz pravaca, a $\{\alpha_k\}$ niz koraka:



Svaka tačka nagomilavanja x^* niza $\{x^k\}$ zadovoljava $\nabla f(x^*) = 0$. Ako je f konveksna funkcija, tada je x^* rešenje problema.

- **Metoda koordinantnog pretraživanja**

Kod ove metode pravci pretraživanja su koordinatni pravci: $\pm e_1, ..., \pm e_n$. Ideja je da se ide istim korakom duž koordinatnih pravaca dok ima napretka. Zatim se smanjuje korak.

Algoritam za rešavanje zadataka pomoću ove metode je sledeći:

- ***Korak 0:*** Izabrati $x^0 \in R^n, \alpha > 0, \varepsilon > o$. Staviti $k = 0$.
- ***Korak 1:*** Ispitati da li postoji $s^k \in \{\pm e_1, ..., \pm e_n\}$ tako da je $f(x^k + \alpha_k s^k) < f(x^k)$ Ako postoji ići na Korak 2. U suprotnom ići na Korak 3.
- ***Korak 2:*** Odrediti $x^{k+1} = x^k + \alpha_k s^k, \alpha_{k+1} = \alpha_k, k = k+1$ i vratiti se na Korak 1.
- ***Korak 3:*** Izračunati $\alpha_k = \alpha_k / 2$ i ići na Korak 1.

Kriterijum zaustavljanja (STOP) je kada je $\alpha_k < \varepsilon$.

Teorema 3.3.10. Ako je x^0 takvo da je skup $X = \{x \in R^n | f(x) \le f(x_0)\}$ ograničen i f je diferencijabilna na X^0 tada svaka tačka nagomilavanja x^* niza $\{x^k\}$ generisanog prethodnim algoritmom zadovoljava uslov $\nabla f(x^*) = 0$.

- **Gradijentna metoda (Metoda najbržeg spusta, Košijeva metoda)**

Kod ove metode se polazi od ideje da funkcija najbrže opada u pravcu antigradijenta $\frac{\partial f}{\partial s} = \nabla f^T s_0 = \|\nabla f\| \cos\varphi$. Minimalna vrednost $\frac{\partial f}{\partial s}$ se dobija za $\varphi = \pi \Rightarrow s^k = -\nabla f(x^k)$.

Algoritam (Košijev) za rešavanje zadataka pomoću ove metode je sledeći:

- ***Korak 0:*** Izabrati $x^0 \in R^n, \varepsilon > o, k = 0$.
- ***Korak 1:*** Izračunati $s^k = -\nabla f(x^k)$.

- ***Korak 2:*** Naći α_k kao rešenje jednodimenzionog problema $\min\limits_{\alpha>0} f\left(x^k + \alpha s^k\right)$.
- ***Korak 3:*** Izračunati $x^{k+1} = x^k + \alpha_k s^k, k = k+1$ i ići na Korak 1.

Kriterijum zaustavljanja (STOP) je kada je $\left\|\nabla f\left(x^k\right)\right\| < \varepsilon$.

Teorema 3.3.11. Neka je skup X^0 ograničen, funkcija f diferencijabilna na X^0. Tada svaka tačka nagomilavanja x^* niza $\left\{x^k\right\}$ generisanog Košijevim algoritmom zadovoljava $\nabla f\left(x^k\right) = 0$.

- **Njutnova metoda**

Kod ove metode se polazi od ideje da je funkcija

$$f(x) \approx f\left(x^k\right) + \nabla f\left(x^k\right)^T\left(x - x^k\right) + \frac{1}{2}\left(x - x^k\right)^T \nabla^2 f\left(x^k\right)\left(x - x^k\right) = Q_k(x) \quad (3.3.26.)$$

gde je x^{k+1} tačka minimuma funkcije $Q_k(x)$. Određuje se iz uslova:

$$\nabla Q_k(x) = \nabla f\left(x^k\right) + \nabla^2 f\left(x^k\right)\left(x - x^k\right) = 0 \Rightarrow$$

$$\nabla^2 f\left(x^k\right)\left(x - x^k\right) = -\nabla f\left(x^k\right) \Rightarrow x^{k+1} = x^k - \left(\nabla^2 f\left(x^k\right)\right)^{-1} \nabla f\left(x_k\right) \Rightarrow \quad (3.3.27.)$$

$$s^k = -\left(\nabla^2 f\left(x^k\right)\right)^{-1} \nabla f\left(x^k\right)$$

Algoritam (Njutnijev) za rešavanje zadataka pomoću ove metode je sledeći:

- ***Korak 0:*** Izabrati $x^0 \in R^n, \varepsilon > o, k = 0$.
- ***Korak 1:*** Izračunati $s^k = -\left(\nabla^2 f\left(x^k\right)\right)^{-1} \nabla f\left(x^k\right)$.
- ***Korak 2:*** Kao kod Košijevog algoritma.
- ***Korak 3:*** Kao kod Košijevog algoritma.

Kriterijum zaustavljanja (STOP) je kao kod Košijevog algoritma.

Teorema 3.3.12. Neka je skup X^0 ograničen, a funkcija f dvaput diferencijabilna na X^0 i $\nabla^2 f(x)$ pozitivno definitna matrica na X^0. Neka je niz $\left\{x^k\right\}$ generisan Njtnovim algoritmom. Tada $x^k \to x^*, k \to \infty$ (x^* je jedinstveno rešenje bezuslovne optimizacije).

Napomena 3.3.7.: Ako je $f(x) = q_0 + c^T x + x^T Q x$ (Q pozitivno definitna matrica) tada Njutnova metoda nalazi minimum u jednom koraku.

3.3.4.2. Uslovna optimizacija

Uslovna optimizacija obuhvata mnoštvo metoda, kao što su:

- ograničenja linearne jednačine (metoda zamene),
- ograničenja linearne nejednačine,
- funkcija cilja kvadratna, ograničenja linearna (kvadratno programiranje),
- ograničenja nelinearne jednačine i/ili nejednačine.

Univerzalnu primenu ima približna metoda kaznenih funkcija, tj. može se primeniti na većinu navedenih problema.

Algoritam približne metode kaznenih funkcija.je sledeći:

- ***Korak 0:*** Izabrati niz kaznenih parametara $\{t_k\}$ i niz spoljašnjih ili unutrašnjih kaznenih funkcija $\{q_k\}, x^0, \varepsilon > o, k = 1$. Ako je izabrana unutrašnja kaznena funkcija, tada mora da važi $x \in \overset{o}{X}$.
- ***Korak 1:*** Staviti $\varepsilon_k = 1/t_k$ (preciznost sa kojom se rešava *k*-ti problem).
- ***Korak 2:*** Polazeći od x^{k-1} kao početne tačke naći približno rešenje x^k problema $\min F_k(x) = f(x) + q_k(x)$ nekim od algoritama bezuslovne optimizacije sa kriterijumom zaustavljanja: STOP ako je $\left\|\nabla F_k\left(x^k\right)\right\| < \varepsilon_k$.
- ***Korak 3:*** Ako je $k = k + 1$ ići na Korak 1.

Kriterijum zaustavljanja: STOP ako je $\varepsilon_k < \varepsilon$.

Napomena 3.3.8.: Problemi u Koraku 2 se rešavaju sa rastućom tačnošću nekim od algoritama bezuslovne minimizacije, koji je sa gledišta ovog algoritma “crna kutija”. Rešenje prethodnog problema je početna tačka za sledeći problem.

3.3.5. Kvadratno programiranje

Problem je sledeći:

$$\begin{aligned} &\min(\max) f(x) = \frac{1}{2} x^T Q x + c^T x \\ &Ax \le b, x \ge 0 \end{aligned} \tag{3.3.28.}$$

Polazi se od sledećih pretpostavki, od kojih zavisi rešavanje zadataka:

1. Ovde se može pretpostaviti da je Q simetrična matrica, jer se svaka kvadratna forma može izraziti preko simetrične matrice.
2. Važe sledeće jednakosti: $\nabla f(x) = Qx + c, \quad \nabla^2 f(x) = Q$.
3. Ako je Q pozitivno definitna matrica (svi glavni minori su pozitivni) onda je funkcija $f(x)$ strogo konveksna funkcija, pa je izraz (3.3.28.) problem

konveksnog programiranja i Kun – Takerovi uslovi su potrebni i dovoljni za globalni minimum.

4. Ako je Q pozitivno semidefinitna matrica (svi simetrični minori su nenegativni) onda je funkcija $f(x)$ konveksna funkcija i važi isto kao pod 3.
5. Ako je Q negativno definitna (f je strogo konkavna) i dopustivi skup je ograničen, tada se svi globalni (i lokalni) minimumi postižu u ekstremnim tačkama dopustivog skupa (tzv. “kišobran”). S obzirom da su kandidati za globalne minimume ekstremne tačke, dovoljno je pretražiti konačan skup ekstremnih tačaka i naći one sa najmanjom vrednošću funkcije cilja. Kod primera malih dimenzija to je jednostavnije nego rešavanje sistema jednačina i nejednačina koji proističu iz Kun – Takerovih uslova. Slično se može postupiti u slučaju da je Q negativno semidefinitna.
6. Ako Q nije definitna kandidati za lokalne minimume se traže preko Kun – Takerovih uslova, koji su potrebni ali ne i dovoljni za optimalnost.

3.3.5.1. Primena Kun – Takerovih uslova

Ekvivalentan zapis izraza (3.3.28.) je:

$$\begin{aligned}
&\min(\max) f(x) = \frac{1}{2} x^T Q x + c^T x \\
&g_i(x) = a_i^T x - b_i \le 0, \qquad i = 1,\dots,m \\
&h_i(x) = -x \le 0 \qquad i = 1,\dots,n
\end{aligned} \tag{3.3.29.}$$

Kun-Takerovi uslovi za rešavanje zadataka kvadratnog programiranja su:

(i) $$\nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) + \sum_{i=1}^{n} \alpha_i \nabla h_i(x) = 0 \Leftrightarrow Qx + c + \sum_{i=1}^{m} \lambda_i a_i + \sum_{i=1}^{n} \alpha_i(-e_i) = 0$$

$$\Leftrightarrow Qx + c + A^T \lambda - \alpha = 0$$

(ii) $$\begin{matrix} \lambda_i g_i(x) = 0, i = 1,\dots,m \\ \alpha_i h_i(x) = 0, i = 1,\dots,n \end{matrix} \Leftrightarrow \begin{matrix} \lambda_i(a_i^T x - b_i) = 0, i = 1,\dots,m \\ \alpha_i x_i = 0, i = 1,\dots,n \end{matrix}$$

(iii) $\lambda \ge 0, \alpha \ge 0$

(iv) $$\begin{matrix} a_i^T x - b_i \le 0, i = 1,\dots,m \\ -x_i \le 0. i = 1,\dots,n \end{matrix} \Leftrightarrow \begin{matrix} a_i^T x - b_i + y_i = 0, i = 1,\dots,m \\ x \ge 0, y \ge 0 \end{matrix} \Leftrightarrow \begin{matrix} Ax + y = b \\ x \ge 0, y \ge 0 \end{matrix}$$

Drugi uslov (ii) se može prikazati na sledeći način:

$\lambda_i(a_i^T x - b_i) = \lambda_i(-y) = 0 \Leftrightarrow \lambda^T y = 0$, jer je $\lambda_i \ge 0, y_i \ge 0, i = 1,\dots,m$

$\alpha_i x_i = 0 \Leftrightarrow \alpha^T x = 0$, jer je $\alpha_i \ge 0, x_i \ge 0, i = 1,\dots,n$

Kun-Takerovi uslovi u izmenjenom redosledu imaju sledeći oblik:

$$\begin{aligned}
&Ax + y = b \\
&-Qx - A^T\lambda + \alpha = c \\
&\lambda^T y = 0, \alpha^T x = 0 \\
&\lambda \geq 0, \alpha \geq 0, x \geq 0, y \geq 0
\end{aligned} \qquad (3.3.30.)$$

Nove oznake koje se unose u Kun – Takerovim uslovima su:

$$M = \begin{bmatrix} O & -A \\ A^T & Q \end{bmatrix}, \qquad q = \begin{bmatrix} b \\ c \end{bmatrix}, \qquad w = \begin{bmatrix} y \\ \alpha \end{bmatrix}, \qquad z = \begin{bmatrix} \lambda \\ x \end{bmatrix}$$

Kun-Takerovi uslovi u novim oznakama:

$$\overset{w}{\begin{bmatrix} y \\ \alpha \end{bmatrix}} - \overset{M}{\begin{bmatrix} O & -A \\ A^T & Q \end{bmatrix}} \overset{z}{\begin{bmatrix} \lambda \\ x \end{bmatrix}} = \overset{q}{\begin{bmatrix} b \\ c \end{bmatrix}}$$

$$\begin{bmatrix} y^T & \alpha^T \end{bmatrix} \begin{bmatrix} \lambda \\ x \end{bmatrix} = 0 \quad tj. \quad y^T\lambda + \alpha^T x = 0 \qquad (3.3.31.)$$

$$y \geq 0, \alpha \geq 0, \lambda \geq 0, x \geq 0$$

Nakon izvršenih transformacija dobijen je tzv. linearni problem komplementarnosti:

$$\begin{aligned}
&w - Mz = q \\
&w^T z = 0 \\
&w \geq 0, z \geq 0
\end{aligned} \qquad (3.3.32.)$$

Rešenja (3.3.32.) daju kandidate za lokalni minimum zadatka kvadratnog programiranja. Ako je Q pozitivno definitna matrica, rešenje (3.3.32.) daje globalno rešenje zadatka kvadratnog programiranja.

3.3.5.2. Rešavanje problema komplementarnosti

Kod ovog problema polazi se od pretpostavke daje Q pozitivno definitna matrica.

Ideja je sledeća: ako je $q \geq 0$ može se uzeti da je $w = q$, $z = 0$. Sledi da je rešenje zadatka kvadratnog programiranja $x = 0, (z = (\lambda.x) = (0,0)) \Rightarrow x = 0$. U suprotnom, uvodi se veštačka promenljiva $z_0 \geq 0$ i uvodi se u izraz $W - MZ - z_0 e = q$.

Odgovarajuća matrica je:

$$\begin{array}{cccc} W & Z & z_0 & \\ \end{array}$$
$$\begin{bmatrix} I & |-M| & |-e| & q \end{bmatrix}$$

Dalji postupak rešavanja se vrši u dve faze.

I faza: Pivotiranjem u koloni z_0 postiže se $q \geq 0$ i z_0 ulazi u bazu, a jedan w_i izlazi.

II faza: Uzastopno se pivotira dok se z_0 ponovo ne izbaci iz baze.

Pravilo izbora pivota: čuva se veza $w_i z_i = 0, i = 1,...,n$ (tzv. *komplementarnost*) i pozitivnost desne strane (kao kod simpleks algoritma).

Pravilo za očuvanje komplementarnosti:

a) ako iz baze izadje w_i u bazu ulazi z_i.
b) ako iz baze izadje z_i u bazu ulazi w_i.

3.3.6. Geometrijski metod nelinearnog programiranja

Kao i kod linearnog programiranja, geometrijski metod se može iskoristiti kod problema koji sadrže n = 2, a najviše n = 3 promenljive. Geometrijski metod, iako ne često korišćen, koristi se zato što olakšava pristup opštoj algebarskoj metodi. Najčešće se rešavaju zadaci koji spadaju u problem nelinearnog programiranja sa linearnim ograničenjima, koji imaju sledeći opšti oblik:

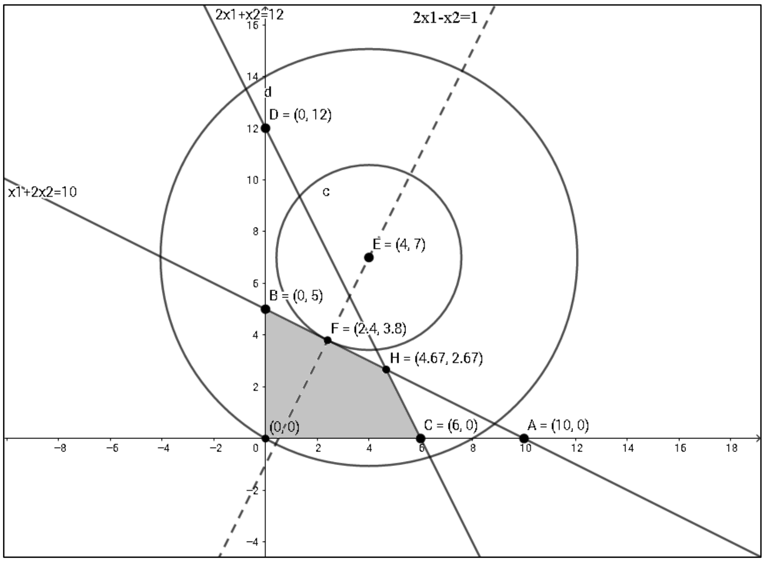
$$max(min) f(x_1,...,x_n)$$
$$g_i : \sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i, (i = 1,...,m)$$
$$x_j \geq 0, (j = 1,...,n)$$

Na sledećem primeru je prikazan postupak rešavanja zadatka nelinearnog programiranja primenom geometrijskog metoda.

Primer 3.3.7. Naći ekstremne vrednosti nelinearne funkcije cilja sa linearnim ograničenjima:

$$F(x) = (x_1 - 4)^2 + (x_2 - 7)^2$$
$$\text{p.o.} \quad x_1 + 2x_2 \leq 10$$
$$2x_1 + x_2 \leq 12$$
$$x_1, x_2 \geq 0$$

Rešenje: Ovaj problem ima samo dve promenljive, pa se optimalno rešenje može utvrditi grafičkom metodom. Problem se rešava tako se sve nejednačine iz sistema ograničenja predstave grafički u pravouglom koordinatnom sistemu po istovetnom postupku kao kod linearnog programiranja. Grafička prezentacija problema nelinearnog programiranja je data na slici 3.3.6.



Slika 3.3.6. *Grafička prezentacija zadatka nelinearnog programiranja*

Kao što se vidi, funkcija cilja predstavlja kružnicu sa centrom u tački (4,7). Minimum će biti u tački u kojoj funkcija cilja dodiruje oblast ograničenja, a maksimum će biti u jednom od temena oblasti ograničenja. Za određivanje koordinate tačke F, u kojoj funkcija cilja ima minimum, koristi se uslov normalnosti pravih $P_1(x_1 + 2x_2 \leq 10)$ i prave P_3 koja prolazi kroz centar kružnice. Koeficijent nagiba prave P_1 se dobija iz transformisanog izraza ove prave: $x_2 = -\frac{1}{2}x_1 + 5$, odakle sledi da je $k_1 = -\frac{1}{2}$. Uslov normalnosti pravih je sledeći: $k_1 = -1/k_2$. Koeficijent pravca prave P_3 na pravu P_1 je $k_2 = 2$. Pošto prava P_3 prolazi kroz centar kružnice, to će se iskoristiti da bi se dobila jednačine ove prave P_3 (uzima se da je $x_1 = 4$, a $x_2 = 7$), tj.: P_3: $x_2 = 2x_1 + n$. Sledi da je $n = x_2 - 2x_1$, odnosno kada se zamene vrednosti $n = 7 - 2 \cdot 4 = -1$. Kada se sve sredi dobija se konačan oblik jednačine P_3: $2x_1 - x_2 = 1$.

Sada treba odrediti koordinate tačke F (tačka minimuma) rešavanjem sistema dve jednačine prave (P_1 i P3) sa dve nepoznate. Rešenje sistema je $x_1 = 2{,}4$, $x_2 = 3{,}8$, odnosno $F(2{,}4;3{,}8)$.

Minimalna vrednost funkcije cilja iznosi:

$$\min F(x) = (2{,}4 - 4)^2 + (3{,}8 - 7)^2 = 12{,}8$$

Maksimum funkcije cilja će biti u nekom od temena oblasti mogućih rešenja $B(0{,}5)$, $H(4{,}67;2{,}67)$, $C(6{,}0)$ i $O(0{,}0)$, u kojima je vrednost funkcije cilja sledeća:

$F(B) = (0 - 4)^2 + (5 - 7)^2 = 20$
$F(H) = (4{,}67 - 4)^2 + (2{,}67 - 7)^2 = 19{,}2$
$F(C) = (6 - 4)^2 + (0 - 7)^2 = 53$
$F(O) = (0 - 4)^2 + (0 - 7)^2 = 65$

Funkcija cilja ima maksimum u tački O(0,0) i ima vrednost $\max F(0{,}0)=65$.

3.3.7. Celobrojno programiranje

Kod mnogih zadataka nelinearnog programiranja postoji zahtev da jedna, više ili sve promenljive budu celobrojne, odnosno da uzmu vrednosti koje su celi brojevi. Prema prirodi zadataka, mogu se odvojiti tri tipa celobrojnog programiranja:

- čisto celobrojno programiranje, gde su sve promenljive celobrojne,
- mešovito celobrojno programiranje, gde su samo neke promenljive celobrojne, i
- binarno programiranje (0-1 programiranje), gde su sve ili neke promenljive binarne (uzimaju vrednosti 1 ili 0).

Opšti model celobrojnog nelinearnog programiranja ima sledeci oblik:

$$\begin{aligned} &\max(\min) f(x_1,...,x_n) \\ &Ax \le B \\ &x \in R^{n_1} x Z^{n_2} \end{aligned} \tag{3.3.33.}$$

Kod binarnog celobrojnog nelinearnog programiranja razmatra se sledeci matematički model

$$\begin{aligned} &\max(\min) f(x_1,...,x_n) \\ &Ax = B \\ &x \in \{0,1\} \end{aligned} \tag{3.3.34.}$$

Tradicionalno, celobrojno nelinearno programiranje se razmatra u kontekstu oblasti globalne optimizacije. Pri tome, fokus je na numeričkim algoritmima za rešavanje nelinearnih kontinualnih problema optimizacije gde se celovitost ograničenja razmatra naknadno, koristeći pri tome metodu grananja i ograničavanja nad celobrojnim promenljivim.

Danas postoje više algoritma za dobijanje celobrojnog rešenja zadataka nelinearnog programiranja. U daljem tekstu se navode algoritmi koji imaju najširu primenu i to:

- konveksna celobrojna maksimizacija,
- konveksna celobrojna minimizacija,
- polinomijalna optimizacija,
- globalna optimizacija,
- ostale metode (pseudo-Boolean optimizacija, kvadratni zadaci kao i opšte 0-1 polinomialno programiranje).

U tabeli 3.3.1. su prikazani algoritmi za dobijanje celobrojnog rešenja zadataka nelinearnog programiranja u zavisnosti od vrste funkcije cilja i ograničenja.

***Tabela 3.3.1.** Računska kompleksnost i algoritmi za celobrojnu nelinearnu optimizaciju*

<table>
<tr><th rowspan="2">Funkcija cilja</th><th colspan="3">Ograničenja</th></tr>
<tr><th>Linearna</th><th>Konveksna polinomijalna</th><th>Proizvoljno polinomijalna</th></tr>
<tr><td>Linearna</td><td>Polinomijalno vremenska u fiksnim dimenzijama:
– Lenstrov algoritam
– Generilozavana bazna redukcija
– Kratke racionalno generisane funkcije</td><td rowspan="2">Polinomijalno vremenska u fiksnim dimenzijama:
– Lenstrov algoritam</td><td>Neizračunljivo: Hilbertov 10. problem</td></tr>
<tr><td>Konveks maks</td><td>Polinomijalno vremenska u fiksnim dimenzijama</td><td>Neizračunljivo</td></tr>
<tr><td>Konveks min</td><td colspan="2">Polinomijalno vremenska u fiksnim dimenzijama: Lenstrov tip algoritma</td><td>Neizračunljivo</td></tr>
<tr><td>Proizvoljno polinomijalna</td><td colspan="2">Neprimereno: čak i za kvadratne forme i za fiksne dimenzije
Fiksne dimenzije i linearna ograničenja: Kratke racionalno generisane funkcije</td><td>Neizračunljivo</td></tr>
</table>

Za rešavanje zadataka binarnog (0-1) programiranja razvijene su posebne metode, jer se tu, zbog konačnog broja mogućih alternativa, ne mogu koristiti kontinualne metode. Neke od tih metoda se zasnivaju na kombinatorici, pa se zato ove metode zovu – kombinatorno programiranje. Postoji više takvih metoda. Najstarija od njih se bazira na ispitivanju svih mogućih varijanti dopustivog plana i odabiranju one varijante koja predstavlja optimalno rešenje. Obzirom da je broj promenljivih koje se pojavljuju u realnim matematičkim modelima često veoma veliki, ova metoda postaje u praksi neupotrebljiva. Iz tog razloga, druge metode za rešavanje zadataka binarnog programiranja odbacuju određene varijante dopustivog plana, za koje se pouzdano zna da ne mogu predstavljati optimalni plan i fokusiraju se na ispitivanju preostalih varijanti i odabiru one varijante koja predstavlja optimalno rešenje.

3.4. PROGRAMSKI PAKET LINGO ZA REŠAVANJE PROBLEMA NELINEARNOG PROGRAMIRANJA

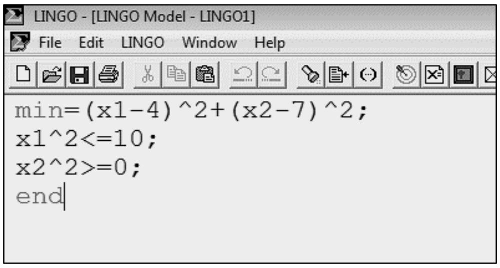
Praktičan način rešavanja problema nelinearnog programiranja predstavlja korišćenje razvijenih programskih paketa, između kojih se izdvaja LINGO. Rešavanje zadataka uz pomoć ovog programskog paketa je pokazano kroz sledeće primere u slučaju nelinerane funkcije cilja i nelinearnih ograničenja (primer 3.3.8.) i nelinerane funkcije cilja i linearnih ograničenja (primer 3.3.9.). U primeru 3.3.10. je rešen zadatak binarnog programiranja (0-1 programiranje).

Primer 3.3.8. Naći minimum nelinearne funkcije cilja sa nelinearnim ograničenjima:

$$
\begin{aligned}
& \min F(x) = (x_1 - 4)^2 + (x_2 - 7)^2 \\
\text{p.o.} \quad & x_1^2 \le 10 \\
& x_2^2 \ge 0
\end{aligned}
$$

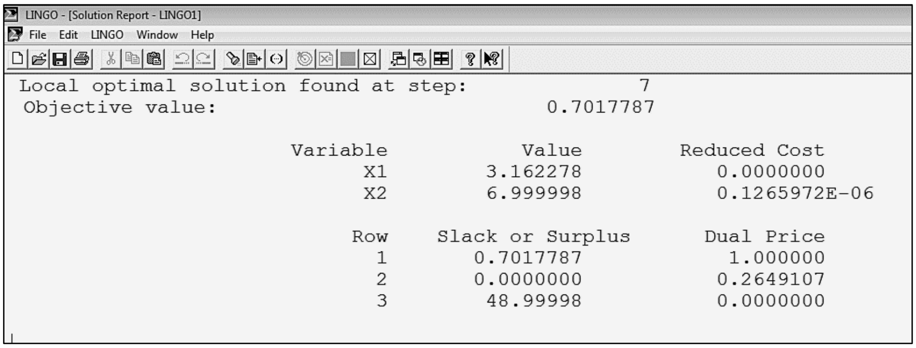
Rešenje: Izgled ekrana sa tekstom zadatka i postavljenim matematičkim modelom je dat na slici 3.3.7., sa koje se mogu videti i osnovna pravila sintakse programa LINGO.

-



Slika 3.3.7. *Izgled ekrana sa postavljenim matematičkim modelom*

Izgled ekrana sa rezultatima je dat na slici 3.3.8.



Slika 3.3.8. *Izgled ekrana sa rezultatima*

Vidi se da je optimalni rezultat u datim uslovima:

$$
- X^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 3{,}162278 \\ 6{,}999998 \end{bmatrix}
$$

Pri ovome, minimum funkcije cilja iznosi min$F(X^*) = 0{,}7017787$.

Primer 3.3.9. Fabrika proizvodi 3 tipa proizvoda, tip I, tip II i tip III. Od strane kupaca je vec naručeno 45 komada proizvoda tipa I, 40 komada tipa II i 10 komada tipa III. Potrebna ulaganja za proizvodnju jednog tipa proizvoda iznose: za tip I - 1750 EUR-a, za tip II - 2050 EUR-a i za tip III - 2750 EUR-a. Menadžment fabrike je zaključio da je opravdano proizvesti i veći broj proizvoda, da ce se za njih pronaći kupci. Raspoloživi fond za ulaganje iznosi do 195 000 EUR-a. Dobit koji se ostvaruje prodajom jednog tipa proizvoda nije linearna, već zavisi i od broja proizvedenih proizvoda (zbog odnosa fiksnih i promenljivih troškova), i iznosi:

Za tip I: $7x_1^2 - x_1$

Za tip II: $9x_2^2 + 4x_2$

Za tip III: $x_3^2 + 7x_3$

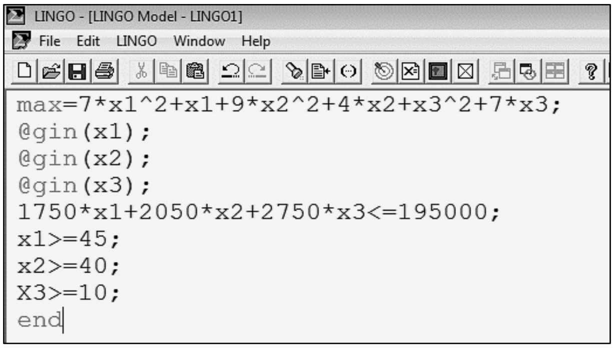
Potrebno je odrediti takav proizvodni program, koji će u uslovima zadatih ograničenja obezbediti maksimalnu dobit.

Rešenje: Matematički model problema:

$$\max F(x) = 7x_1^2 - x_1 + 9x_2^2 + 4x_2 + x_3^2 + 7x_3$$

$$\begin{array}{llllll}
\text{p.o.} & 1750x_1 + 2050x_2 + 2750x_3 & \le & 195000 \\
 & x_1 & \ge & 45 \\
 & x_2 & \ge & 40 \\
 & x_3 & \ge & 10
\end{array}$$

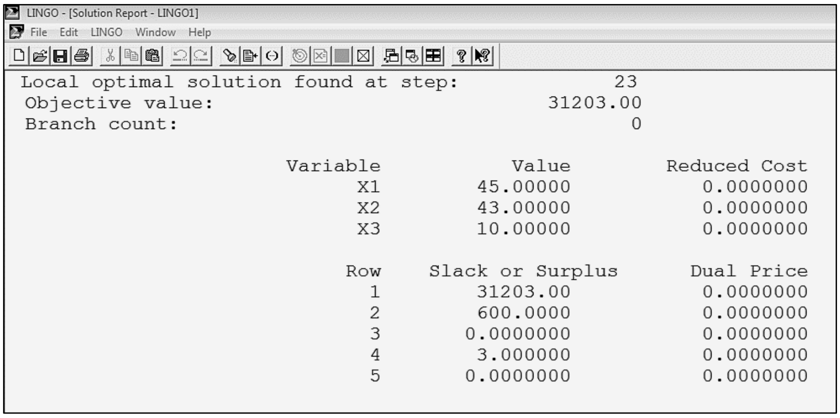
Dodatno ograničenje je da se traži celobrojno rešenje, pošto se radi o proizvodima. Izgled ekrana sa postavljenim matematičkim modelom je prikazan na slici 3.3.9.




```
max=7*x1^2+x1+9*x2^2+4*x2+x3^2+7*x3;
@gin(x1);
@gin(x2);
@gin(x3);
1750*x1+2050*x2+2750*x3<=195000;
x1>=45;
x2>=40;
X3>=10;
end
```

Slika 3.3.9. *Izgled ekrana sa postavljenim matematičkim modelom*

Izgled ekrana sa rezultatima je dat na slici 3.3.10.




```
Local optimal solution found at step:          23
Objective value:                          31203.00
Branch count:                                    0

                   Variable           Value        Reduced Cost
                         X1        45.00000           0.0000000
                         X2        43.00000           0.0000000
                         X3        10.00000           0.0000000

                        Row    Slack or Surplus      Dual Price
                          1        31203.00           0.0000000
                          2        600.0000           0.0000000
                          3       0.0000000           0.0000000
                          4        3.000000           0.0000000
                          5       0.0000000           0.0000000
```

Slika 3.3.10. *Izgled ekrana sa rezultatima*

Vidi se da je optimalni rezultat u datim uslovima:

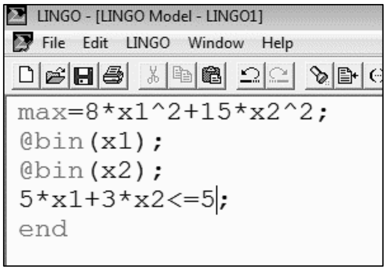
$$X^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \end{bmatrix} = \begin{bmatrix} 45 \\ 43 \\ 10 \end{bmatrix}$$

Pri ovome, maksimum funkcije cilja iznosi $\max F(X^*) = 31203$.

Primer 3.3.10. Naći maximum nelinearne funkcije cilja sa linearnim ograničenjem, pri čemu promenljive mogu uzeti samo vrednosti $\{0,1\}$:

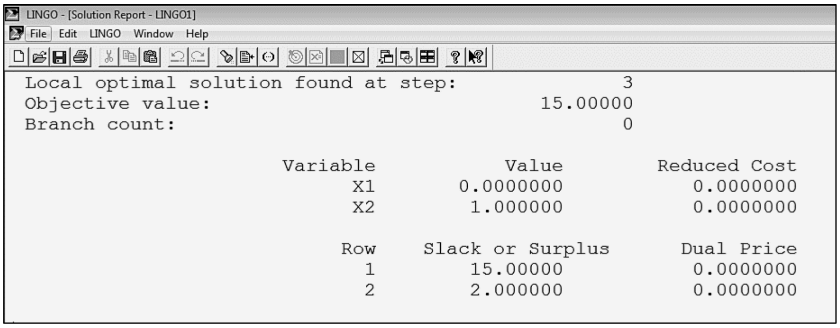
$$\max F(x) = 8x_1^2 + 15x_2^2$$
$$\text{p.o.} \quad 5x_1 + 3x_2 \leq 5$$

Rešenje: Izgled ekrana sa tekstom zadatka i postavljenim matematičkim modelom je dat na slici 3.3.11.



Slika 3.3.11. *Izgled ekrana sa postavljenim matematičkim modelom*

Izgled ekrana sa rezultatima je dat na slici 3.3.12.



Slika 3.3.12. *Izgled ekrana sa rezultatima*

Vidi se da je optimalni rezultat u datim uslovima:

$$X^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Pri ovome, maximum funkcije cilja iznosi $\max F(X^*) = 15$.

4. DINAMIČKO PROGRAMIRANJE

Dinamičko programiranje predstavlja specijalan slučaj nelinearnog programiranja. Dinamičko programiranje je poseban matematički aparat, koji omogućuje optimalno planiranje tzv. višeetapnih procesa upravljanja, kao što su procesi u kojima se donosi niz odluka i to postepeno u vremenu. U praksi se javljaju mnogi zadaci upravljanja procesima (u tehnici, ekonomiji, vojsci, nauci, itd.) koji se mogu predstaviti u vidu višeetapnih procesa, na koje se može primeniti metoda dinamičkog programiranja.

Rešenje zadatka predstavlja dobijanje optimalnog plana upravljanja, odnosno ono upravljanje koje daje najbolje rešenje kojim se postiže maksimalni cilj operacije. Shodno tome, u cilju primene metode dinamičkog programiranja neophodno je da se za svaki razmatrani proces formira odgovarajući matematički model, sa precizno definisanom funkcijom cilja, koji treba maksimizirati ili minimizirati, kao i ograničenjima koja se javljaju u toku procesa. Kod ovako postavljenog zadatka treba naći funkcionalne relacije primenom principa optimalnosti

Određivanje optimalnog plana upravljanja se zasniva na principu optimalnosti koji se, prema osnivaču ove discipline, naziva Belmanov princip optimalnosti. Belmanov princip optimalnosti se može definisati kao osobinu da optimalna strategija, bez obzira na početno stanje i prethodno dobijeno rešenje, u svakoj narednoj etapi treba da odredi optimalnu strategiju u odnosu na prethodno dobijeno stanje, kao rezultat prethodnog rešenja. Princip optimalnosti se može prikazati matematičkim putem funkcionalnim jednačinama ili rekurzivnim relacijama pomoću kojih se izražavaju veze između funkcija cilja posmatrane etape i funkcije susedne etape.

4.1. OSNOVNI POJMOVI DINAMIČKOG PROGRAMIRANJA

Pri definisanju osnovnih pojmova dinamičkog programiranja polazi se od termina sistema. Sistem predstavlja svaki fizički (tehnički, ekonomski, privredni...) sistem koji se može prikazati kao vektor stanja:

$$r(t) = [r_1(t) \quad r_2(t) \quad \cdots \quad r_N(t)] \tag{4.1.1.}$$

Osobine sistema su određene komponentama vektora $r(t)$. Broj N naziva se dimenzijom sistema. Ako se sa p_0 označi početno, a sa p_1, p_2,...p_n,… stanja sistema u uzastopnim vremenskim trenucima, onda važi sledeća relacija:

$$p_{n+1} = W(p_n) \qquad n = 0,1,2,... \tag{4.1.2.}$$

koja određuje skup vektora (p_0, p_1, p_2,…) kao reprezentaciju ponašanja sistema u diskretnim trenucima vremena n = 0, 1, 2,… Na osnovu toga se može zaključiti da takav skup vektora definiše jednu specijalnu vrstu procesa koja se naziva višeetapni proces. Ova vrsta procesa je definisana početnim stanjem sistema $p = p_0$ i trasformacijom $W(p)$, što se može predstaviti na sledeći način:

$$[p, W(p)] \tag{4.1.3.}$$

Shodno tome, proces se može opisati kao ponašanje sistema u toku vremena.

Kada se govori o funkcijama procesa, skalarne funkcije $G(p_0, p_1, p_2, \ldots, p_n)$ koje zavise od procesa mogu se, kod višeetapnih procesa, iskazati na sledeći način:

$$\sum_{i=0}^{n} G(p_i) \tag{4.1.4.}$$

4.2. VRSTE PROCESA

Procesi se mogu podeliti na sledeći način:

- diskretni i neprekidni procesi,
- stacionarni i nestacionarni procesi,
- determinirani i stohastički procesi.

Diskretni procesi su višeetapni procesi u kojima se stanje sistema menja samo u diskretnim trenucima vremena. ***Neprekidni procesi*** su procesi u kojima se stanje sistema neprekidno menja u vremenu. Izučavanje neprekidnih procesa je mnogo složenije u odnosu na diskretne procese. Za njihovo izučavanje se mogu koristiti rezultati dobijeni za diskretne procese, pustivši pri tome da dužina intervala vremena između dva posmatranja teži nuli. Takođe, kod njih se mora dokazati postojanje i jedinstvenost rešenja. U praksi se iz datih razloga najčešće izučavaju diskretni višeetapni procesi.

Stacionarni procesi su višeetapni procesi gde je oblik transformacije $W(p)$ takav da ne zavisi od vremena. ***Nestacionarni procesi*** su su procesi gde je oblik transformacije $W(p)$ takav da zavisi od vremena. Tu važi sledeća relacija

$$p_n + 1 = W_n(p_n) \tag{4.2.1.}$$

gde indeks n označava da transformacija W zavisi od vremena.

Determinirani procesi su procesi kod kojih je stanje sistema p_m+1 jedinstveno određeno na osnovu stanja sistema p_m, putem transformacije W, odnosno

$$p_m + 1 = W(p_m) \tag{4.2.2.}$$

Kod ***stohastičkih procesa*** transformacija W nije potpuno poznata, tako da je to stohastička transformacija koja daje slučajni vektor stanja p_1, čija je raspodela verovatnoća odredjena sa p. Niz takvih vektora $[p, p_1, p_2,\ldots]$ definiše diskretni višeetapni proces stohastičkog tipa. Shodno tome su i odgovarajuće funkcije slučajne veličine. Kod određivanja numeričkih rezultata, uzimaju se matematička očekivanja funkcija ovih slučajnih promenljivih.

4.3. OPŠTE KARAKTERISTIKE DINAMIČKOG PROGRAMIRANJA I NJEGOVA PRIMENA

Opšte karakterisitke modela dinamičkog programiranja su:

- globalni model (problem) se raščlanjuje na faze (etape) i u svakoj fazi se vrši optimizacija (maksimizacija dobiti, minimizacija troškova i dr.),
- etapa (faza) se definiše kao uređen skup stanja,
- prilikom optimizacije procesa upravljanja (ili dejstva) transformacija svakog stanja tekuće etape (faze) je povezana sa sledećom etapom,
- u analizi trenutnog stanja, najbolje upravljanje u narednoj etapi je nezavisno od upravljanja (akcija) koje je realizovano u prethodnoj etapi, i
- optimalno upravljanje počinje sa prvom ili poslednjom fazom, u zavisnosti od prirode modela.

U cilju optimalnog upravljanja višeetapnim procesima potrebno je uraditi matematičku karakterizaciju ovog zadatka, odnosno izabrati neki numerički kriterijum F kojim se može dovoljno dobro opisati stanje u procesu i time izmeriti i samu efikasnost upravljanja. Kod višeetepenog procesa sa n faza se u svakoj fazi primenjuje neko upravljanje U_i ($i = 1,\ldots, n$). Upravljanje celim procesom predstavlja niz upravljanja po fazama, što se može prikazati na sledeći način

$$U = (U_1, U_2, \ldots, U_n) \qquad (4.3.1.)$$

koje se zove strategija. Svakoj izabranoj strategiji odgovara određena vrednost kriterijuma F, odnosno

$$F(U) = F(U_1, U_2, \ldots, U_n) \qquad (4.3.2.)$$

Osnovni cilj kod optimalnog upravljanja jeste odrediti takvu strategiju U^* za koju kriterijum F postiže svoju optimalnu vrednost, tj

$$F(U^*) = (opt)F \qquad (4.3.3.)$$

Ako se sa f_i označi rezultat koji se postiže u i-toj fazi upravljanja U_i, onda je ukupan efekat f_u optimizacije kriterijuma F jednak

$$f_u = f_1 + f_2 + \cdots + f_n \qquad (4.3.4.)$$

Određivanje optimalne vrednosti kriterijuma F se vrši tako da upravljanje u svakoj fazi obezbeđuje optimalno nastavljanje procesa. Drugim rečima, u i-toj fazi izbor upravljanja se vrši s obzirom na mogućnosti izbora u preostalim fazama $j=i+1,\ldots,n$. Na kraju se upravljanja U_k biraju tako da se postigne optimalno upravljanje procesom ne samo u toj fazi već u svim fazama $j = i$, i + 1,…, n zajedno.

4.3.1. Belmanov princip optimalnosti

Osnovnu ideju o primeni metoda dinamičkog programiranja, koja je u prethodnom tekstu objašnjena, postavio je Ričard Belman (Richard Bellman) 1957. godine. Matematički model Belmanovog principa optimalnosti za slučaj aditivne funkcije kriterijuma polazi od sledećeg: neka su date funkcije $f_1(x_1)$, $f_2(x_2),\ldots, f_n(x_n)$ i pretostavka je da treba naći vrednosti $x_1, x_2, \ldots, x_n$ za koje funkcija

$$F(x_1, x_2, \ldots, x_n) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \qquad (4.3.5.)$$

postiže maksimalnu vrednost pri ograničenjima

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \le b_n \qquad (4.3.6.)$$

gde je $a_j \geq 0$, $x_j \geq 0$ i $b_n \geq 0$. Za proizvoljno $k = 1,\ldots, n$ se uvodi niz funkcija $\{F_k(b_k)\}$ i to:

$$\begin{aligned} F_1(b_1) &= \max_{a_1x_1 \le b_1}\{f_1(x_1)\} = \max_{x_1 \le b_1/a_1}\{f_1(x_1)\}, \\ F_2(b_2) &= \max_{a_1x_1 + a_2x_2 \le b_2}\{f_1(x_1) + f_2(x_2)\} \\ &= \max_{a_2x_2 \le b_2}\left\{\max_{a_1x_1 \le b_2 - a_2x_2}\left[f_1(x_1) + f_2(x_2)\right]\right\} \\ &= \max_{a_2x_2 \le b_2}\left\{f_2(x_2) + \max_{a_1x_1 \le b_2 - a_2x_2}\left[f_1(x_1)\right]\right\} \\ &= \max_{a_2x_2 \le b_2}\{f_2(x_2) + F_1(b_2 - a_2x_2)\}. \end{aligned} \qquad (4.3.7.)$$

U opštem slučaju, važi sledeći izraz:

$$F_k(b_k) = \max_{x_k \le b_k/a_k}\{f_k(x_k) + F_{k-1}(b_k - a_kx_k)\}, \qquad k = 2,\ldots,n \qquad (4.3.8.)$$

pri čemu je $F_0(b_0) = 0$.

Koristeći Belmanove jednačine (4.3.7.) i (4.3.8.) polazni zadatak sa n promenljivih se razlaže na n jednostavnijih problema sa po jednom promenljivom i na taj način se rešavanje može realizovati po fazama. Funkcija F_1 se određuje neposredno, dok se funkcije $F_2,\ldots, F_n$ određuju pomoću rekurentne formule (4.3.8.), pri čemu $F_n(b_n)$ predstavlja traženu optimalnu vrednost.

Praktična primena modela dinamičkog programiranja je u ovom udžbeniku prikazana na sledećim primerima:

– jednodimenzionalna raspodela resursa,
– višedimenzionalna raspodela resursa,
– određivanje najkraćeg puta u mreži,
– raspodela poslova na mašine,
– optimalna politika zamene opreme.

4.3.2. Jednodimenzionalna raspodela resursa

Jednodimenzionalna raspodela resursa predstavlja zadatak koji razmatra kako raspodeliti jednorodni resurs na n mesta trošenja (različiti proizvodni procesi prerade (utroška) razmatranog resursa, različite linije, odnosno mašine, radna mesta, itd.) pri čemu je količina datog resursa ograničena. Svako mesto trošenja ima različitu efikasnost, koja može da se ogleda kroz dobit ili troškove koji se ostvaruju utroškom određene količine resursa na određenom mestu trošenja. Kapacitet svakog mesta trošenja je takođe ograničen.

Ako funkcija cilja ima zadatak da maksimizira dobit, onda ona ima sledeći oblik:

$$\max F(X) = \sum_{j=1}^{n} f_j(x_j) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \tag{4.3.9.}$$

gde je

– x_j – količina resursa dodeljena j-tom procesu (mašini, liniji,...),
– $f_j(x_j)$ – funkcija dobiti koja se ostvari kada se količina resursa x_j dodeli j-tom procesu.

Pošto je količina resursa ograničena, sledi da je

$$x_1 + x_2 + \cdots + x_n \le S \tag{4.3.10.}$$

gde je S ukupna količina raspoloživih resursa. U slučaju da su kapaciteti procesa ograničeni, javljaju se i sledeća ograničenja:

$$0 \le x_j \le Q_j$$

(4.3.11.)

gde je Q_j kapacitet procesa (mesta utroška resursa).

Ako se ovaj problem intrepretira pomoću Belmanovog principa optimalnosti onda se on može definisati na sledeći način: ako se određena količina ograničenog resursa S, koja će se označiti sa x_n dodeli n-tom procesu, onda količinu koja preostaje (S–x_n) treba raspodeliti na ostale procese (mesta utroška) tako da ostvarena dobit bude maksimalna.

Koristeći Belmanove jednačine (4.3.7.) i (4.3.8.), maksimalna dobit, koja se ostvaruje raspodelom ograničene količine resursa S na n procesa se može prikazati na sledeći način:

$$F_n(S) = \max_{0 \le x_n \le S} \{ f_n(x_n) + F_{n-1}(S - x_n) \} \tag{4.3.12.}$$

Dobit na osnovu dodele količine resursa x_1 prvom procesu (prvom mestu utroška resursa) iznosi:

$$F_1(S) = \max_{0 \le x_1 \le Q_1} \{ f_1(x_1) \} \tag{4.3.13.}$$

uz $f_0(S) = 0$ što sledi iz same definicije funkcije.

Ako se sa x_i^* označe one vrednosti resursa pri kojima se u i-tom procesu postiže maksimalna vrednost dobiti, onda sledi:

$$F_n(S) = \max_{0 \le x_n^* \le S} \left\{ f_n\left(x_n^*\right) + F_{n-1}\left(S - x_n^*\right) \right\} \tag{4.3.14.}$$

$$F_{n-1}\left(S - x_n^*\right) = \max_{0 \le x_{n-1} \le S - x_n^*} \left\{ f_{n-1}\left(x_{n-1}\right) + F_{n-2}\left[\left(S - x_n^*\right) - x_{n-1}\right] \right\} \tag{4.3.15.}$$

Na isti način se računaju i ostale vrednosti $x_{n-1}^*, ..., x_2^*, x_1^*$, imajući u vidu da su to sve delovi ograničenog resursa S koji se dodeljuju odgovarajućim procesima. Za vrednost x_1^* maksimalna dobit u prvom procesu je:

$$F_1\left(S - x_n^* - x_{n-1}^* - \cdots - x_2^*\right) = \max_{0 \le x_1 \le S - x_n^* - \cdots - x_2^*} \left\{ f_1\left(x_1\right) \right\} = f_1\left(S - x_n^* - x_{n-1}^* - \cdots - x_2^*\right) \tag{4.3.16.}$$

Primer 4.3.1. Za potrebe rada proizvodnog sistema, potrebno je table iverice (jednorodni resurs) raspodeliti na tri moguće proizvodne linije, pri čemu je kapacitet proizvodnih linija limitiran na količinu iverice (resursa) $0 \le x_i \le 3$ za i =1 – 3. Količina iverice (broj tabli) je takođe u razmatranom periodu ograničena i iznosi S = 6 jedinica. Ostvarena dobit na linijama zavisi od količine (broja) preradjene iverice, i to:

- za liniju 1: $f_1 = 4x_1^2$
- za liniju 2: $f_2 = 7x_2^2$
- za liniju 3: $f_3 = 3x_3^2$

Ograničenu količinu tabli iverice treba tako distribuirati proizvodnim linijama da ostvarena dobit od njene prerade, u razmatranom periodu bude maksimalna.

Rešenje: Funkcije cilja i ograničenja iznose:

$$\max F(x) = 4x_1^2 + 7x_2^2 + 3x_3^2$$
$$\text{p.o.} \quad x_1 + x_2 + x_3 \le 6$$
$$0 \le x_i \le 3$$

Dobit u funkciji raspoloživog resursa je data u tabeli 4.3.1.

Tabela 4.3.1. *Dobit u zavisnosti od raspoloživog resursa*

S	**x_1**	**$F_1(S)$**	**X_2**	**$F_2(S)$**	**X_3**	**$F_3(S)$**
0	0	0	0	0	0	0
1	1	4	1	7	0	7
2	2	16	2	28	0	28
3	3	36	3	63	0	63
4	3	36	3	67	0	67
5	3	36	3	63	0	63
6	3	36	3	63	0	63

Kolona $F_1(S)$ u tabeli je izračunata pomoću izraza:

$$F_1(S) = \max_{0 \le x_1 \le 3} \left\{ 4x_1^2 \right\}$$

Ako se razmatraju i prvi i drugi proces, vrednosti u koloni $F_2(S)$ se dobijaju pomoću izraza:

$$F_2(S) = \max_{0 \le x_2 \le 3} \left\{ 7x_2^2 + F_1(S - x_2) \right\}$$

Sledi:

– $S = 0$

$$F_2(0) = \max_{x_2 = 0} \left\{ 7 \cdot 0^2 + F_1(0-0) = 0 \right\} = 0$$

– $S = 1$

$$F_2(1) = \max_{\substack{x_2=0 \\ x_2=1}} \left\{ \begin{matrix} 7 \cdot 0^2 + F_1(1-0) = 4 \\ 7 \cdot 1^2 + F_1(1-1) = 7 \end{matrix} \right\} = 7$$

– $S = 2$

$$F_2(2) = \max_{\substack{x_2=0 \\ x_2=1 \\ x_2=2}} \left\{ \begin{matrix} 7 \cdot 0^2 + F_1(2-0) = 16 \\ 7 \cdot 1^2 + F_1(2-1) = 11 \\ 7 \cdot 2^2 + F_1(2-2) = 28 \end{matrix} \right\} = 28$$

– $S = 3$

$$F_2(3) = \max_{\substack{x_2=0 \\ x_2=1 \\ x_2=2 \\ x_2=3}} \left\{ \begin{matrix} 7 \cdot 0^2 + F_1(3-0) = 36 \\ 7 \cdot 1^2 + F_1(3-1) = 23 \\ 7 \cdot 2^2 + F_1(3-2) = 32 \\ 7 \cdot 3^2 + F_1(3-3) = 63 \end{matrix} \right\} = 63$$

– $S = 4$

$$F_2(4) = \max_{\substack{x_2=0 \\ x_2=1 \\ x_2=2 \\ x_2=3}} \left\{ \begin{matrix} 7 \cdot 0^2 + F_1(4-0) = 36 \\ 7 \cdot 1^2 + F_1(4-1) = 43 \\ 7 \cdot 2^2 + F_1(4-2) = 44 \\ 7 \cdot 3^2 + F_1(4-3) = 67 \end{matrix} \right\} = 67$$

– $S = 5$

$$F_2(5) = \max_{\substack{x_2=0\\x_2=1\\x_2=2\\x_2=3}} \begin{Bmatrix} 7\cdot 0^2 + F_1(5-0) = 36 \\ 7\cdot 1^2 + F_1(5-1) = 43 \\ 7\cdot 2^2 + F_1(5-2) = 64 \\ 7\cdot 3^2 + F_1(5-3) = 79 \end{Bmatrix} = 79$$

– $S = 6$

$$F_2(6) = \max_{\substack{x_2=0\\x_2=1\\x_2=2\\x_2=3}} \begin{Bmatrix} 7\cdot 0^2 + F_1(6-0) = 36 \\ 7\cdot 1^2 + F_1(6-1) = 43 \\ 7\cdot 2^2 + F_1(6-2) = 64 \\ 7\cdot 3^2 + F_1(6-3) = 99 \end{Bmatrix} = 99$$

Ako se na kraju razmatraju sva tri procesa, vrednosti u koloni $F_3(S)$ se dobijaju pomoću izraza:

$$F_3(S) = \max_{0\le x_3\le 3}\left\{3x_2^2 + F_2(S-x_3)\right\}$$

Sledi:

– $S = 0$

$$F_3(0) = \max_{x_3=0}\left\{3\cdot 0^2 + F_2(0-0) = 0\right\} = 0$$

– $S = 1$

$$F_3(1) = \max_{\substack{x_3=0\\x_3=1}} \begin{Bmatrix} 3\cdot 0^2 + F_2(1-0) = 7 \\ 3\cdot 1^2 + F_2(1-1) = 3 \end{Bmatrix} = 7$$

– $S = 2$

$$F_3(2) = \max_{\substack{x_3=0\\x_3=1\\x_3=2}} \begin{Bmatrix} 3\cdot 0^2 + F_2(2-0) = 28 \\ 3\cdot 1^2 + F_2(2-1) = 10 \\ 3\cdot 2^2 + F_2(2-2) = 12 \end{Bmatrix} = 28$$

– $S = 3$

$$F_3(3) = \max_{\substack{x_3=0\\x_3=1\\x_3=2\\x_3=3}} \begin{Bmatrix} 3\cdot 0^2 + F_2(3-0) = 63 \\ 3\cdot 1^2 + F_2(3-1) = 31 \\ 3\cdot 2^2 + F_2(3-2) = 19 \\ 3\cdot 3^2 + F_2(3-3) = 27 \end{Bmatrix} = 63$$

– $S = 4$

$$F_3(4) = \max_{\substack{x_3=0\\x_3=1\\x_3=2\\x_3=3}} \left\{ \begin{matrix} 3\cdot 0^2 + F_2(4-0) = 67 \\ 3\cdot 1^2 + F_2(4-1) = 66 \\ 3\cdot 2^2 + F_2(4-2) = 40 \\ 3\cdot 3^2 + F_2(4-3) = 34 \end{matrix} \right\} = 67$$

– $S = 5$

$$F_3(5) = \max_{\substack{x_3=0\\x_3=1\\x_3=2\\x_3=3}} \left\{ \begin{matrix} 3\cdot 0^2 + F_2(5-0) = 79 \\ 3\cdot 1^2 + F_2(5-1) = 70 \\ 3\cdot 2^2 + F_2(5-2) = 75 \\ 3\cdot 3^2 + F_2(5-3) = 55 \end{matrix} \right\} = 79$$

– $S = 6$

$$F_3(6) = \max_{\substack{x_3=0\\x_3=1\\x_3=2\\x_3=3}} \left\{ \begin{matrix} 3\cdot 0^2 + F_2(6-0) = 99 \\ 3\cdot 1^2 + F_2(6-1) = 82 \\ 3\cdot 2^2 + F_2(6-2) = 79 \\ 3\cdot 3^2 + F_2(6-3) = 90 \end{matrix} \right\} = 99$$

Maksimalna vrednost koja je dobijena na osnovu treće funkcije je i rešenje problema, tako da je:

$$\max F(x) = 99 \text{ n.j.}$$

Optimalna vrednost za treću promenljivu je $x_3 = 0$ ($F_3(6) = 99$). Vrednosti ostale dve promenljive se određuju is tabele 4.3.1. (poslednji red), tako da je $x_2 = 3$, dok iz ograničenja sledi da je $x_1 = 3$.

Zadatak jednodimenzionalne raspodele resursa se može postaviti i da se minimiziraju određene vrednosti procesa pri trošenju ograničenog resursa. Ako funkcija cilja ima zadatak da minimizira troškove, onda ona ima sledeći oblik:

$$\min F(X) = \sum_{j=1}^{n} f_j(x_j) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \qquad (4.3.17.)$$

gde je

– x_j – količina resursa dodeljena j-tom procesu (mašini, liniji,...),
– $f_j(x_j)$ – funkcija troškova koji nastaju kada se količina resursa x_j dodeli j-tom procesu.

Pošto je količina resursa ograničena, sledi da je

$$x_1 + x_2 + \cdots + x_n \leq S \qquad (4.3.18.)$$

gde je S ukupna količina raspoloživih resursa.

U slučaju da su kapaciteti procesa ograničeni, javljaju se i sledeća ograničenja:

$$0 \le x_j \le Q_j \tag{4.3.19.}$$

gde je Q_j kapacitet procesa (mesta utroška resursa).

Ako se i ovaj problem intrepretira pomoću Belmanovog principa optimalnosti onda se on može definisati na sledeći način: ako se određena količina ograničenog resursa S, koja će se označiti sa x_n dodeli n-tom procesu, onda količinu koja preostaje (S–x_n) treba raspodeliti na ostale procese (mesta utroška) tako da troškovi budu minimalni.

Minimalni troškovi, koji se ostvaruju raspodelom ograničene količine resursa S na n procesa se može prikazati na sledeći način:

$$F_n(S) = \min_{0 \le x_n \le S} \{ f_n(x_n) + F_{n-1}(S - x_n) \} \tag{4.3.20.}$$

Troškovi na osnovu dodele količine resursa x_1 prvom procesu (prvom mestu utroška resursa) iznose:

$$F_1(S) = \min_{0 \le x_1 \le Q_1} \{ f_1(x_1) \} \tag{4.3.21.}$$

uz $f_0(S) = 0$ što sledi iz same definicije funkcije.

Ako se sa x_i^* označe one vrednosti resursa pri kojima se u i-tom procesu postižu minimalne vrednosti troškova, onda sledi:

$$F_n(S) = \min_{0 \le x_n^* \le S} \left\{ f_n\left(x_n^*\right) + F_{n-1}\left(S - x_n^*\right) \right\} \tag{4.3.22.}$$

$$F_{n-1}\left(S - x_n^*\right) = \min_{0 \le x_{n-1} \le S - x_n^*} x \left\{ f_{n-1}(x_{n-1}) + F_{n-2}\left[\left(S - x_n^*\right) - x_{n-1}\right] \right\} \tag{4.3.23.}$$

Na isti način se računaju i ostale vrednosti $x_{n-1}^*, \ldots, x_2^*, x_1^*$. Za vrednost x_1^* minimalni troškovi u prvom procesu iznose:

$$F_1\left(S - x_n^* - x_{n-1}^* - \cdots - x_2^*\right) = \min_{0 \le x_1 \le S - x_n^* - \cdots - x_2^*} \left\{ f_1(x_1) \right\} = f_1\left(S - x_n^* - x_{n-1}^* - \cdots - x_2^*\right) \tag{4.3.24.}$$

4.3.3. Višedimenzionalna raspodela resursa

Višedimenzionalna raspodela resursa predstavlja zadatak koji razmatra kako raspodeliti resurse S_1 i S_2 na n mesta trošenja, pri čemu je količina datih resursa ograničena. Svako mesto trošenja ima različitu efikasnost koja se ostvaruju utroškom određene količine resursa na određenom mestu trošenja. Kapacitet svakog mesta trošenja je takođe ograničen.

Shodno tome, funkcija cilja ima sledeći oblik:

$$\max/\min F(X,Y) = \sum_{j=1}^{n} f_j(x_j, y_j) \tag{4.3.17.}$$

gde je

- x_j – deo resursa S_1 dodeljena j-tom procesu (mašini, liniji,...),
- y_j – deo resursa S_2 dodeljena j-tom procesu (mašini, liniji,...),
- $f_j(x_j, y_j)$ – optimalni rezultat koji se ostvari kada se količina resursa x_j i y_j dodeli j-tom procesu.

Pošto je količina resursa ograničena, sledi da je

$$x_j \le S_1 \tag{4.3.18.}$$

$$y_j \le S_2 \tag{4.3.19.}$$

$$x_{j,\min} \le x_j \le x_{j,\max} \tag{4.3.20.}$$

$$y_{j,\min} \le y_j \le y_{j,\max} \tag{4.3.21.}$$

gde su $x_{j,min}/x_{j,max}$ dozvoljena minimalna/maksimalna vrednost x_j i $y_{j,min}/y_{j,max}$ dozvoljena minimalna/maksimalna vrednost y_j.

Koristeći Belmanove jednačine (4.3.7.) i (4.3.8.), optimalni rezultat koji se ostvaruje raspodelom ograničene količine resursa S_1 i S_2 na n procesa se može prikazati na sledeći način:

$$F_{k,j}^{*} = \max_{x_0,\, y_{j,\min} \le y_j \le y_{j,\max}}/\min \left\{ f_j + F_{k-1,(j-y_j)}^{*} \right\} \tag{4.3.22.}$$

gde je

- k – oznaka koraka,
- x_0 – početna tačka.

U prvom koraku ($k = 1$) se prvom mestu trošenja dodeljuju delovi resursa x_0 i y_j za koje funkcija (4.3.22.) ima optimalnu (maksimalnu/minimalnu) vrednost – $F_{k,j}^{*}$. Preostali deo resursa S_2 se dalje raspodeljuje na n – 1 mesta trošenja, uzimajući u obzir rezultate raspodela u prethodnim koracima. Nakon toga, na i-to mesto trošenja se raspodeljuje deo resursa $x_{0,i}$ i y_0 za koji funkcija (4.3.22.) ima optimalnu (maksimalnu/minimalnu) vrednost. Preostali deo resursa S_1 se dalje raspodjeljuje na n – 1 mesta trošenja, uzimajući u obzir rezultate raspodela u prethodnim koracima.

$$F_{k,j}^{*} = \max_{x_{j,\min} \le x_j \le x_{j,\max},\, y_0}/\min \left\{ f_j + F_{k-1,(j-x_j)}^{*} \right\} \tag{4.3.23.}$$

gde je y_0 tačka približavanja.

Na kraju se dobija:

$$F = \sum_{j=1}^{n} f^*\left(x_j, y_j\right) = \sum_{j=1}^{n} f\left(x_j^*, y_j^*\right) \tag{4.3.24.}$$

4.3.4. Određivanje najkraćeg puta u mreži

Određivanje najkraćeg puta je deo dinamičkog programiranja koji se bavi komunikacijskim sistemima i mrežama, pri čemu je jedan od ključnih zadataka kako pronaći najkraći put u mreži od početne do krajnje tačke. Istraživanjima u ovoj oblasti su se bavili mnogi autori i ona su dovela do pronalazaka nekoliko različitih rešenja, odnosno algoritama za rešavanje problema najkraćeg puta u mreži. Napoznatiji algoritmi za rešavanje problema najkraćeg puta u mreži su Dijkstrinov, Belman-Fordov, Flojd-Varšalov algoritam i drugi. Izbor odgovarajućeg algoritma kojim se pokušava rešiti problem najkraćeg puta u mreži vrši se zavisno od zadatih parametara mreže i samog vremena izvođenja algoritma.

Grafovi ili mreže mogu imati i pozitivne i negativne težinske vrednosti grana. Ako graf $G=(V,E)$ ne sadrži negativne cikluse ili petlje dostupne iz izvorišnog čvora, tada za $v \in V$ važi da je težinska vrednost najkraćeg puta $\delta(s,v)$ dobro definisana, te može da poprimi negativnu vrednost. Sa druge strane, ako postoji negativni ciklus koji počinje u čvoru s, tada težinske vrednosti najkraćeg puta nisu dobro definisane, tako da se tu uvek može pronaći kraći put sa manjom težinskom vrednošću i tu se ulazi u negativni ciklus. Tada se težinska vrednost najkraćeg puta definiše kao: $\delta(s,v) = -\infty$. Neki algoritmi rade samo sa nenegativnim težinskim vrednostima grana (Dijkstrinov algoritam), dok drugi algoritmi rade sa negativnim težinskim vrednostima grana sve dok u mreži ne postoji negativni ciklus (Belman-Fordov algoritam).

4.3.5. Raspodela poslova na mašine

Problem raspodele različitih poslova na određen broj mašina koje mogu da realizuju te različite poslove predstavlja problem dinamičkog programiranja koji ima zadatak da maksimizira dobit dobijanjem optimalnog rešenja (optimalne raspodele poslova na mašine). Problem se može prikazati na sledeći način:

- neka u kompaniji postoji n jednorodnih mašina koje mogu da obavljaju poslove 1 i 2,
- ako x mašina u toku prvog razmatranog perioda realizuje operaciju 1, to kompaniji donosi dobit $g(x)$,

– na kraju prvog perioda jedan deo mašina će biti amortizovan, odnosno neće biti za dalju upotrebu, tako da se u narednom periodu (drugom) može računati sa $a(x)$ mašina,
– ako y mašina u toku prvog razmatranog perioda realizuje operaciju 2, to će kompaniji doneti dobit $h(y)$,
– na kraju prvog perioda jedan deo mašina će takođe biti amortizovan, odnosno neće biti za dalju upotrebu, tako da se u narednom periodu (drugom) može računati sa $b(y)$ mašina,

Prvi period se može definisati pomoću sledećih jednačina:

– za operaciju 1 se određuje broj mašina x_1, dok za obavljanje operacije 2 preostaje sledeći broj mašina:

$$y_1 = n_1 - x_1 \tag{4.3.25.}$$

gde je $n_1 = n$ – ukupan broj raspoloživih mašina na početku.

– ukupna dobit u prvom periodu iznosi:

$$g(x_1) + h(y_1) \tag{4.3.26.}$$

– broj upotrebljivih mašina na kraju prvog perioda (upotrebljivih za rad u drugom periodu) iznosi:

$$n_2 = a(x_1) + b(y_1) \tag{4.3.27.}$$

Drugi period se može definisati pomoću narednih jednačina:

– broj upotrebljivih mašina iznosi:

$$n_2 = a(x_1) + b(y_1)$$

– za operaciju 2 se određuje broj mašina x_2 (od raspoloživih n_2 mašina), dok za obavljanje operacije 2 preostaje sledeći broj mašina:

$$y_2 = n_2 - x_2 \tag{4.3.28.}$$

– ukupna dobit u drugom periodu iznosi:

$$g(x_2) + h(y_2) \tag{4.3.29.}$$

– broj upotrebljivih mašina na kraju prvog perioda (upotrebljivih za rad u trećem periodu) iznosi:

$$n_3 = a(x_2) + b(y_2) \tag{4.3.30.}$$

Na identičan način se mogu napisati izrazi do N-tog perioda.

Zadatak optimizacije je da se odredi broj mašina koje će obavljati operaciju 1 i operaciju 2 u svakom periodu, a da je pri tome ostvarena dobit posle N perioda maksimalna.

Matematički model zadatka raspodele poslova na mašine se može prikazati na sledeći način:

- Funkcija cilja:

$$\max F(X,Y)=\sum_{i=1}^{N} g(x_i)+h(y_i) \tag{4.3.31.}$$

- Ograničenja (proističu od broja raspoloživih mašina u svakom periodu):

$$\begin{aligned} & x_i + y_i = n_i \\ & n_{i+1} = a(x_i)+b(y_i) \qquad i=1,2,\ldots,N-1 \\ & 0 \le x_i \le n_i \qquad i=1,2,\ldots,N \end{aligned} \tag{4.3.32.}$$

Ako su funkcija cilja i ograničenja linearna, problem se svodi na problem linearnog programiranja. U drugom slučaju, kada se javlja nelinearnost problem se rešava metodom dinamičkog programiranja.

- Funkcionalne relacije:

Ako se za $f_N(n)$ označi maksimalna dobit u N perioda, gde je n broj raspoloživih mašina na početku prvog perioda, tada se rekurentne jednačine ovog zadatka dinamičkog programiranja mogu napisati u sledećem obliku:

- do kraja planskog perioda predstoji samo još jedan period (ili se razmatra problem koji ima samo jedan period):

$$f_1(n)=\max_{0\le x\le n}\{g(x)+h(y)\}=\max_{0\le x\le n}\{g(x)+h(n-x)\} \tag{4.3.33.}$$

- do kraja planskog perioda predstoji još k perioda, pri čemu je $k > 1$:

$$f_k(n)=\max_{0\le x\le n}\{g(x)+h(n-x)+ f_{k-1}[a(x)+b(n-x)]\} \tag{4.3.34.}$$

Primer 4.3.2. Preduzeće raspolaže sa 100 visokoproduktivnih strugova. U naredne četiri godine (N = 4) preduzeće treba da izrađuje vratila tipa A i tipa B, koji se mogu izrađivati na strugovima. U zavisnosti od raspodele strugova, u i-tom periodu se može ostvariti dobit:

$$g(x_i) + h(y_i) = 0{,}6x_i + 0{,}4y_i$$

Amortizacija zavisi od vrste vratila koji se izrađuje na strugovima i iznosi 60 % za strugove koje izrađuju vratila tipa A i 30 % za strugove koje izrađuju vratila tipa B, tako da broj raspoloživih strugova za naredni period iznosi:

$$n_{i+1} = 0{,}4x_i + 0{,}7y_i$$

Odrediti broj strugova koji će izrađivati vratila tipa A i vratila tipa B u svakom periodu, a da pri tome ostvarena dobit posle N = 4 perioda bude maksimalna.

Rešenje: Matematički model problema je:

Funkcija cilja:

$$\max F(X,Y)=\sum_{i=1}^{4}(0{,}6x_i + 0{,}4y_i)$$

Ograničenja:

$$x_i + y_i = n_i$$
$$n_{i+1} = 0{,}4x_i + 0{,}7y_i$$
$$0 \le x_i \le n_i, \qquad n_1 = n$$

Prvo se razmatra samo četvrti period (period u kome je do kraja planskog perioda ostao samo jedan, taj četvrti). Maksimalna dobit u ovom periodu iznosi:

$$f_1(n) = \max_{0 \le x_4 \le n_4} \left(0{,}6x_4 + 0{,}4y_4\right)$$

Pošto iz ograničenja sledi da je:

$$x_4 + y_4 = n_4$$

onda je

$$y_4 = n_4 - x_4$$

pa kad se zameni y_4 u izrazu za $f_1(n)$ dobija se:

$$f_1(n) = \max_{0 \le x_4 \le n_4} \left(0{,}6x_4 + 0{,}4(n_4 - x_4)\right)$$

Kada se sve sredi, dobija se konačni izraz za $f_1(n)$, odnosno:

$$f_1(n) = \max_{0 \le x_4 \le n_4} \left(0{,}2x_4 + 0{,}4n_4\right)$$

Sada se razmatraju dva slučaja i to kada je $x_4 = 0$ i kada je $x_4 = n_4$. Usvaja se bolji rezultat.

Za $x_4 = 0$ sledi:

$$f_1(n) = 0{,}4n_4$$

Za $x_4 = n_4$ sledi:

$$f_1(n) = 0{,}6n_4$$

Usvaja se veći rezultat, odnosno za četvrti period treba da bude $x_4 = n_4$, pri čemu je vrednost dobiti u ovom periodu $f_1(n) = 0{,}6n_4$.

U drugom koraku (k=2) posmatra se maksimalna dobit za četvrti i treći period, pa rekurentna jednačina za ove periode ima sledeći oblik:

$$f_2(n) = \max_{0 \le x_3 \le n_3} \left\{0{,}6x_3 + 0{,}4y_3 + f_1\left(0{,}4x_3 + 0{,}7y_3\right)\right\}$$

Pošto iz ograničenja sledi da je:

$$x_3 + y_3 = n_3$$

onda je

$$y_3 = n_3 - x_3$$

pri čemu je $f_1 = 0{,}6$. Kada se ove vrednosti ubave u izraz za $f_2(n)$ dobija se:

$$f_2(n) = \max_{0 \le x_3 \le n_3} \left\{0{,}6x_3 + 0{,}4(n_3 - x_3) + 0{,}6\left[0{,}4x_3 + 0{,}7(n_3 - x_3)\right]\right\}$$

Kada se sve sredi, dobija se konačni izraz za $f_2(n)$, odnosno:

$$f_2(n) = \max_{0 \le x_3 \le n_3} \left\{0{,}02x_3 + 0{,}82n_3\right\}$$

Sada se takođe razmatraju dva slučaja i to kada je x_3 = 0 i kada je $x_3 = n_3$. Usvaja se bolji rezultat (veća vrednost dobiti).

Za x_3 = 0 sledi:

$$f_2(n) = 0{,}82 n_3$$

Za $x_3 = n_3$ sledi:

$$f_2(n) = 0{,}84 n_3$$

Usvaja se veći rezultat, odnosno za treći i četvrti period treba da bude $x_3 = n_3$, pri čemu je vrednost dobiti u ovim periodima $f_2(n) = 0{,}84 n_3$.

U trećem koraku (k=3) posmatra se maksimalna dobit za četvrti, treći i drugi period, pa rekurentna jednačina za ove periode ima sledeći oblik:

$$f_3(n) = \max_{0 \le x_2 \le n_2} \{0{,}6 x_2 + 0{,}4 y_2 + f_2(0{,}4 x_2 + 0{,}7 y_2)\}$$

Pošto iz ograničenja sledi da je:

$$x_2 + y_2 = n_2$$

onda je

$$y_2 = n_2 - x_2$$

pri čemu je f_2 = 0,84. Kada se ove vrednosti ubave u izraz za $f_3(n)$ dobija se:

$$f_3(n) = \max_{0 \le x_2 \le n_2} \{0{,}6 x_2 + 0{,}4(n_2 - x_2) + 0{,}84[0{,}4 x_2 + 0{,}7(n_2 - x_2)]\}$$

Kada se sve sredi, dobija se konačni izraz za $f_3(n)$, odnosno:

$$f_3(n) = \max_{0 \le x_2 \le n_2} \{-0{,}052 x_2 + 0{,}988 n_2\}$$

Dalje, razmatraju se slučajevi kada je x_2 = 0 i kada je $x_2 = n_2$. Usvaja se bolji rezultat (veća vrednost dobiti).

Za x_2 = 0 sledi:

$$f_3(n) = 0{,}988 n_2$$

Za $x_2 = n_2$ sledi:

$$f_3(n) = 0{,}936 n_2$$

Usvaja se veći rezultat, odnosno za drugi, treći i četvrti period treba da bude x_2 = 0, pri čemu je vrednost dobiti u ovim periodima $f_3(n) = 0{,}988 n_2$.

Ako se uzmu u obzir sva četiri perioda, rekurentna jednačina za sve periode ima sledeći oblik:

$$f_4(n) = \max_{0 \le x_1 \le n_1} \{0{,}6 x_1 + 0{,}4 y_1 + f_3(0{,}4 x_1 + 0{,}7 y_1)\}$$

Pošto iz ograničenja sledi da je:

$$x_1 + y_1 = n_1$$

onda je

$$y_1 = n_1 - x_1$$

pri čemu je f_3 = 0,988. Kada se ove vrednosti ubave u izraz za $f_4(n)$ dobija se:

$$f_4(n) = \max_{0 \le x_1 \le n_1} \{0{,}6x_1 + 0{,}4(n_1 - x_1) + 0{,}988[0{,}4x_1 + 0{,}7(n_1 - x_1)]\}$$

Kada se sve sredi, dobija se konačni izraz za $f_4(n)$, odnosno:

$$f_4(n) = \max_{0 \le x_1 \le n_1} \{-0{,}0964x_1 + 1{,}0916n_1\}$$

Dalje, razmatraju se slučajevi kada je $x_1 = 0$ i kada je $x_1 = n_1$. Usvaja se bolji rezultat (veća vrednost dobiti).

Za $x_1 = 0$ sledi:

$$f_4(n) = 1{,}0916n_1$$

Za $x_1 = n_1$ sledi:

$$f_4(n) = 0{,}9952n_1$$

Usvaja se veći rezultat, odnosno za sve periode treba da bude $x_1 = 0$, pri čemu je vrednost dobiti u ovim periodima $f_4(n) = 1{,}0916n_1$.

Rešenje problema je:

$$\max F(X,Y) = f_4(100) = 1{,}0916 \cdot 100 = 109{,}16$$

Maksimalna dobit iznosi 109,16 n.j.

Broj strugova koji će izrađivati vratila tipa A i vratila tipa B u svakom periodu iznosi:

- Prvi period:

  $x_1 = 0$ sledi $y_1 = 100$

  U prvom periodu svi strugovi će izraditi vratila tipa B.

  Na kraju prvog perioda zbog amortizacije preostaće sledeći broj strugova koji prelazi u drugi period:

$$n_2 = 0{,}4x_1 + 0{,}7y_1 = 0{,}4\cdot 0 + 0{,}7\cdot 100 = 70$$

- Drugi period:

  $n_2 = 70$

  $x_2 = 0$ sledi $y_2 = 70$

  I u drugom periodu svi strugovi će izraditi vratila tipa B.

  Na kraju drugog perioda zbog amortizacije preostaće sledeći broj strugova koji prelazi u treći period:

$$n_3 = 0{,}4x_2 + 0{,}7y_2 = 0{,}4\cdot 0 + 0{,}7\cdot 70 = 49$$

- Treći period:

  $n_3 = 49$

  $x_3 = 49$ sledi $y_3 = 0$

  U trećem periodu svi strugovi će izraditi vratila tipa A.

  Na kraju trećeg perioda zbog amortizacije preostaće sledeći broj strugova koji prelazi u četvrti period:

$$n_4 = 0{,}4x_3 + 0{,}7y_3 = 0{,}4\cdot 49 + 0{,}7\cdot 0 = 19{,}6 \approx 20$$

- Četvrti period:

  $n_4 = 20$

  $x_4 = 20$ sledi $y_4 = 0$

I u četvrtom periodu svi strugovi će izraditi vratila tipa A.

Na kraju četvrtog perioda zbog amortizacije preostaće sledeći broj strugova za eventualni dalji rad (peti period):

$$n_5 = 0{,}4x_4 + 0{,}7y_4 = 0{,}4\cdot 20+0{,}7\cdot 0 = 16$$

4.3.6. Optimalna politika zamene opreme

Optimalna politika zamene opreme ima zadatak da maksimizira dobit tokom razmatranog perioda rada od N godina pravovremenim zamenama opreme.

Optimizacija kod zamene opreme se bazira na određenim parametrima kao što su produktivnost opreme, troškovi korišćenja opreme i preostala vrednost opreme.

Neka je:

- $D(t)$ – dobit koja se ostvaruje upotrebom opreme stare t godina,
- $C(t)$ – godišnji troškovi održavanja opreme,
- $V(t)$ – preostala vrednost opreme stare t godina
- C – nabavna cena nove opreme.

Neka funkcija cilja $f_N(t)$ predstavlja dobit u razmatranom periodu od N godina koja se ostvaruje eksploatacijom opreme stare t godina. Optimizacija se oslanja na određivanju funkcionalnih relacija, koje definišu veze među karakterističnim veličinama u toku svaka dva susedna perioda. Pri tome, svi periodi na koje se ukupni period N rasčlanjuje se računaju od kraja. Na početku bilo kog k-tog perioda organizacija raspolaže sa opremom starom t godina i moguća su dva alternativna rešenja:

- zadržati staru opremu (staru t godina) i ostvariti dobit:

$$D(t) - C(t) + f_{k-1}(t+1) \tag{4.3.35.}$$

- nabaviti novu opremu (t = 0) i ostvariti dobit:

$$V(t) - C + D(0) - C(0) + f_{k-1}(1) \tag{4.3.36.}$$

Optimalna dobit od eksploatacije opreme u preostalih k perioda $f_k(t)$ je:

$$f_k(t) = \max\begin{cases} D(t) - C(t) + f_{k-1}(t+1) \\ V(t) - C + D(0) - C(0) + f_{k-1}(1) \end{cases} \tag{4.3.37.}$$

Maksimalna dobit koja se može ostvariti u slučaju da je preostao samo jedan period (k = 1) korišćenjem opreme stare t godina (t = 1, 2,..., N) iznosi:

$$f_1(t) = \max\begin{cases} D(t) - C(t) \\ V(t) - C + D(0) - C(0) \end{cases} \tag{4.3.38.}$$

5. OPTIMALNO REZERVIRANJE

Optimizacija mnogih procesa koji se pojavljuju u praksi zahteva projektovanje kompleksnih tehničkih, organizacionih i drugih sistema. Od sistema projektovanih na takav način se zahteva visok stepen pouzdanosti. Pouzdanost se definiše kao verovatnoća, na određenom nivou poverenja, da će sistem uspešno, bez otkaza, obaviti funkciju za koju je namenjen, unutar definisanih granica performansi, u toku određenog vremena trajanja zadataka, kada se koristi na propisani način i u svrhu za koju je namenjen, pod specificiranim nivoima opterećenja, uzimajući u obzir i prethodno vreme korišćenja sistema. Drugim rečima, pouzdanost predstavlja sposobnost sistema da uspešno obavlja određenu funkciju, pod određenim uslovima i u određenom vremenskom intervalu. Vidi se da je pouzdanost verovatnoća, što znači broj između 0 i 1 ili 0% i 100%. Problem pouzdanosti se ogleda u tome što se sa povećanjem složenosti i broja komponenti sistema, pouzdanost rada istog opada. Na primer, tehnički sistemi predstavljaju skupove elemenata i relacija između njih i njihovih karakteristika, povezanih međusobno u celinu, pri čemu se njihova kompleksnost ogleda u sledećem:

- struktura sastavljena od funkcionalnih celina (sistem, podsistem, sklopovi, podsklopovi i elementi),
- međusobna povezanost funkcionalnih celina, i
- visok stepen složenosti struktura sistema.

Osim pouzdanosti, za ocenu rada sistema su značajni i drugi pokazatelji, kao što su:

- nivo poverenja (verovatnoća da je neki parametar unutar određenih granica ili je iznad donje granice i najčešće se predstavlja u vidu intervala, uz poverenje da će stvarna vrednost biti u tom intervalu),
- rad bez otkaza (stanje kada su sve performanse sistema u skladu sa specifikacijama),
- otkaz (stanje nesposobnosti sistema da izvršava zahtevane funkcije ili stanje kada performanse sistema nisu u skladu sa specifikacijama),
- prethodno vreme korišćenja sistema (parametar koji ima uticaj na izračunavanje pouzdanosti izvršenja određenog zadatka),
- vreme trajanja zadatka (veličina koja je obrnuto proporcionalna nivou pouzdanosti, tj. ako se želi veoma visoka pouzdanost onda vreme trajanja zadatka treba da traje što kraće), i dr.

Zadatak optimizacije kod optimalnog rezerviranja se sastoje u iznalaženju ekstremne vrednosti određenog željenog parametra sistema (pozdanost, otkazi, i dr.), pri određenim ograničenjima. Najćešća ograničenja se odnose na troškove ili na neke druge parametre kao što su brzina, težina, površina, zapremina, itd.).

Ovako formulisani zadatak kod optimalnog rezerviranja predstavlja primarni zadatak optimizacije.

Na osnovu ovako formulisanog primarnog zadatka se može postaviti i dualni zadataka (problem). Kod ovog tipa zadatka se zahteva dostizanje vrednosti nekog od pokazatelja kvaliteta funkcionisanja sistema (takođe pozdanost, otkazi, i dr.), pri najracionalnijoj raspodeli resursa za realizaciju datog sistema (najčešći uslovi su minimalni troškovi).

Najčešći zadaci kod optimalnog rezerviranja se odnose na probleme optimalne organizacije održavanja, optimalnog režima eksploatacije, određivanju optimalnog nivoa rezervnih delova, itd.

Povećanje pouzdanosti sistema se zasniva na povećanju pouzdanosti pojedinih elemenata, odnosno podsistema čime se povećava pouzdanost rada sistema u celini. Ako to tehnološki nije izvodljivo, onda se povećanje pouzdanosti sistema može ostvariti dupliranjem, tripliranjem ili višestrukim povećanjem količina pojedinih elemenata sistema (takozvana "vruća rezerva" ili stand-by elementi) ili određivanjem optimalne količine rezervnih elemenata svih tipova ("hladna rezerva", magacinska rezerva) koji ulaze u sastav podsistema i sistema kao celine.

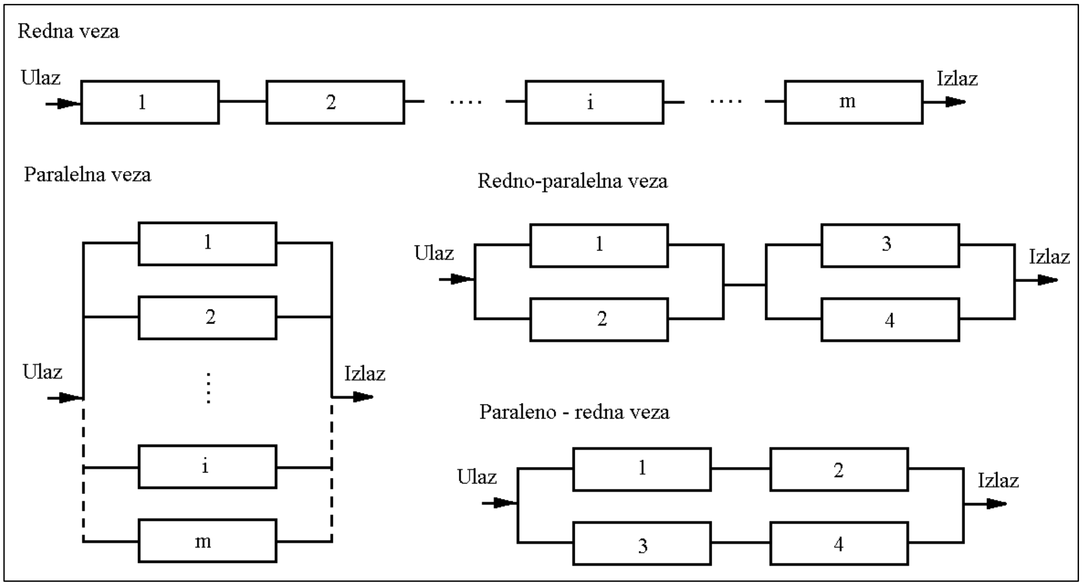
5.1. OSNOVNI POJMOVI I OZNAKE

Sa aspekta optimalnog rezerviranja, pod pojmom "sistem" se podrazumeva objekat koji je sastavljen od *n*-nezavisnih (u smislu pouzdanosti) podsistema, kod koga otkaz bilo kog od *n*-podsistema prouzrokuje otkaz sistema u celini. Shodno tome, podsistem se može definisati kao takav deo sistema koji, osim što njegov otkaz prouzrokuje otkaz celog sistema, ima nezavisno od ostalih podsistema svoje rezervne elemente.

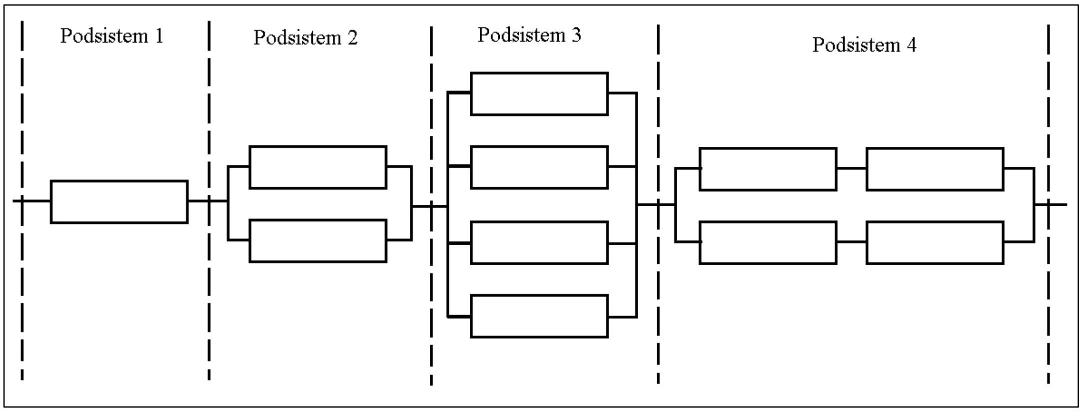
Podsistem je sastavljen od jednog ili više elemenata po pravilu istog tipa. Elementi podsistema mogu međusobno biti povezani na četiri načina: redno, paralelno, redno – paralelno i paralelno – redno (slika 5.1.1.).

Shodno tome, na slici 5.1.2. prikazan je složeni sistem koji se sastoji od više podsistema, sa raznim vezama između njihovih elemenata.

Podsistem je sastavljen od grupe elemenata istog tipa nezavisno od toga gde su oni ugrađeni u sistem. Svaki element podsistema ima određenu pouzdanost funkcionisanja. Shodno tome, svaki podsistem je određen nekom karakteristikom pouzdanosti koja zavisi od uloge podsistema u sklopu celog sistema. Brojna vrednost pokazatelja pouzdanosti podsistema zavisi od broja rezervnih elemenata datog podsistema. Kod složenih sistema i podsistema, gde se javljaju kombinacije rednih i paralelnih veza, pouzdanost takvih sistema i podsistema dobija se razlaganjem na prostije celine koje imaju ili samo rednu ili samo paralelnu konfiguraciju.



Slika 5.1.1. *Tipovi veza između elemenata podsistema*



Slika 5.1.2. *Složeni sistem i podsistemi*

Osnovni pojmovi i oznake koje se koriste kod rešavanja zadataka optimalnog rezerviranja, gde pouzdanost podsistema zavisi od broja rezervnih elemenata datog podsistema su:

- broj rezervnih elemenata *i*-tog podsistema – x_i;
- vektor čije su komponente broj rezervnih elemenata svih podsistema – X = (x_1, x_2,$\cdots$, x_n);
- vektor koji određuje sastav sistema – R(X) = R(x_1, x_2,$\cdots$, x_n);
- pokazatelj pouzdanosti jednog elementa *i*-tog podsistema (verovatnoća bezotkazanog rada u funkciji vremena ili koeficijent gotovosti) – p_i;

- potrebna vrednost pokazatelja pouzdanosti sistema kod rešavanja primarnog (ishodnog) zadatka – P_0;
- pokazatelj pozdanosti i-tog podsistema koji ima x_i rezervnih elemenata – $P_i(x_i)$;
- pokazatelj pouzdanosti sistema u celini, kada i-ti podsistem ima x_i rezervnih elemenata – $P(x_i)$. Ovaj pokazatelj pouzdanosti sistema u celini predstavlja određenu funkciju, koja zavisi od vrednosti pokazatelja pouzdanosti podsistema, odnosno broja rezervnih elemenata u svakom od podsistema, pa tu važi sledeća relacija:

  $$P(x_1, x_2,\cdots, x_n) = P\{P_1(x_1), P_2(x_2),\cdots, P_n(x_n)\} \tag{5.1.1.}$$

  U najvećem broju slučajeva pouzdanost sistema P(X) može da se odredi za sve sisteme, kod kojih otkaz bilo kog podsistema dovodi do otkaza celog sistema. Na osnovu toga, može da se primeni sledeća relacija:

  $$P(X) = P_1(x_1) \cdot P_2(x_2) \cdots P_n(x_n) = \prod_{I=1}^{n} P_i(x_i) \tag{5.1.2.}$$

  Prilikom razmatranja podsistema jednog složenog sistema, podrazumeva se da je zavisnost P_i od x_i poznata, tako da se sve potrebne vrednosti $P_i(x_i)$ mogu izračunati.
- vektor pokazatelj sastava i pouzdanosti – R(P; X);
- pokazatelj nepouzdanosti (verovatnoća otkaza) jednog elementa i-tog podsistema – $q_i = 1 - p_i$;
- vrednost pokazatelja nepouzdanosti (verovatnoća otkaza) sistema kod rešavanja primarnog (ishodnog) zadatka optimalnog rezerviranja – Q_0;
- pokazatelj nepouzdanosti (verovatnoća otkaza) i-tog podsistema koji sadrži x_i rezervnih elemenata – $Q_i(x_i)$;
- pokazatelj nepouzdanosti (verovatnoća otkaza) celog sistema, kada i-ti podsistem ima x_i rezervnih elemenata – Q(X);
- cena jednog elementa i-tog tipa – c_i;
- zadano ograničenje u odnosu na ukupne troškove sistema, pri rešavanju dualnog zadatka optimalnog rezerviranja – C_0;
- troškovi i-tog podsistema kada je on sastavljen od x_i rezervnih elemenata – $C_i(x_i)$;
- troškovi celog sistema, pri čemu i-ti podsistem ima x_i rezervnih elemenata – C(X). Računaju se po sledećem obrascu:

  $$C(X) = C_1(x_1) + C_2(x_2) + \cdots + C_n(x_n) = \sum_{i=1}^{n} C_i(x_i) \tag{5.1.3.}$$

- vektor pokazatelj pouzdanosti i troškova – R(P; C);
- težina jednog elementa i-tog tipa – g_i;
- ograničenje težine celog sistem – G_0;
- težina i-tog podsistema kada je on sastavljen od x_i rezervnih elemenata – $G_i(x_i)$;

- težina celog sistema, pri čemu i-ti podsistem ima x_i rezervnih elemenata – G(X);
- zapremina jednog elementa i-tog tipa – v_i;
- ograničenje zapremine celog sistem – V_0;
- zapremina i-tog podsistema kada je on sastavljen od x_i rezervnih elemenata – $V_i(x_i)$;
- zapremina celog sistema, pri čemu i-ti podsistem ima x_i rezervnih elemenata – V(X);

5.2. ANALIZA POUZDANOSTI SLOŽENIH SISTEMA

Kao što je rečeno, pri analizi pouzdanosti nekog kompleksnog sistema, neophodno je izvršiti razlaganje tog sistema na funkcionalne celine – podsisteme, uređaje, blokove, elemente itd. Shodno tome, analiza pouzdanosti sistema se bazira na poznavanju pouzdanosti sastavnih delova i njihovog uticaja na rad sistema.

5.2.1. Pouzdanost kod redne veze elemenata sistema

Pri analizi pouzdanosti sistema veoma često se javlja redna veza elemenata sistema. Redna veza elemenata je prikazana na slici 5.1.1.

Da bi ovakav sistem ispravno radio, neophodno je da svi elementi redne veze ispravno rade.

Ako sa x_i obeležimo događaj koji označava ispravan rad i-tog elementa, tada će pouzdanost, odnosno verovatnoća ispravnog rada i-tog elementa biti $P(x_i)$. Pouzdanost celog sistema ili dela sistema gde postoji redna veza elemenata, ako elementi pri funkcionisanju ne utiču jedni na druge, može da se odredi pomoću jednačine (5.1.2.).

U slučaju kada su elementi identični, pri čemu pouzdanost svakog elementa iznosi p, pouzdanost sistema iznosi:

$$P = p^n \qquad (5.2.1.)$$

gde je
n – broj redno povezanih elemenata sistema.

5.2.2. Pouzdanost kod paralelne veze elemenata sistema

Paralelna veza se može definisati kao sistem od n elemenata za čiji je ispravan rad dovojlno da bar jedan element ispravno radi – slika 5.1.1.

Paralelna veza se javlja u dva slučaja: kao rezultat strukture samog sistema, i kao posledica konstrukcijskog rešenja kada se radi povećanja pouzdanosti ugrađuju rezervni elementi koji se uključuju kada radni element otkaže.

Računanje verovatnoće kod ove vrste veze elemenata je komplikovano, pa se zato računanje vrši preko nepouzdanosti Q. Naime, sistem će otkazati samo ako svi elementi otkažu, pa u slučaju da su otkazi elemenata nezavisni, pouzdanost sistema iznosi:

$$P = 1 - Q(x_1) \cdot Q(x_2) \cdots Q(x_n) = 1 - \prod_{i=1}^{n} Q(x_i) \qquad (5.2.2.)$$

Pošto je

$$Q(x_i) = 1 - P(x_i) \qquad (5.2.3.)$$

izraz (5.2.2.) može da se napiše u sledećem obliku:

$$P = 1 - \big(1 - P(x_1)\big) \cdot \big(1 - P(x_2)\big) \cdots \big(1 - P(x_n)\big) = \prod_{i=1}^{n} \big(1 - P(x_i)\big) \qquad (5.2.4.)$$

U slučaju kada su svi elementi identični, pri čemu je pouzdanost svakog od njih p, izraz (5.2.4.) dobija sledeći oblik:

$$P = 1 - (1 - p)^n \qquad (5.2.5.)$$

gde je
n – broj paralelno povezanih elemenata sistema.

5.2.3. Pouzdanost kod redno-paralelne veze elemenata sistema

Kada se javljaju kombinacije redne i paralelne veze elemenata sistema, pouzdanost takvog sistema dobija se razlaganjem na prostije celine koje imaju ili samo rednu ili samo paralelnu konfiguraciju i primenjuju se odgovarajući postupci proračuna koji su dati u prethodnim tačkama ovog udžbenika (tačke 5.2.1. i 5.2.2.).

Na slici 5.1.1. prikazana je redno-paralelna veza elemenata sistema. Ova struktura je sačinjena od n redno vezanih grupa koje se sastoje od jednakog broja m paralelno vezanih elemenata. Shodno dosadašnjim razmatranjima, pouzdanost *j-te* grupe iznosi:

$$P_j = 1 - Q(x_{1j}) \cdot Q(x_{2j}) \cdots Q(x_{nj}) = 1 - \prod_{i=1}^{m} Q(x_{ij}) \qquad (5.2.6.)$$

Ukupna pouzdanost cele redno-paralelne veze će biti:

$$P = P_1 \cdot P_2 \cdots P_n = \prod_{j=1}^{n} P_j \qquad (5.2.7.)$$

Ako su svi elementi identični i pouzdanost svakog od njih jednaka *p*, jednačina (5.2.7.) dobija oblik:

$$P = \prod_{j=1}^{n} [1-(1-p)^m] = [1-(1-p)^m]^n \qquad (5.2.8.)$$

5.2.4. Pouzdanost kod paralelno-redne veze elemenata sistema

U ovom slučaju se razmatra *m* paralelnih grana sa po *n* elemenata u svakoj grani. Pretpostavka je da su svi otkazi međusobno nezavisni. Da bi se odredila pouzdanost cele konfiguracije najpre je potrebno odrediti pouzdanost *i*-te grane.

Ona se određuje pomoću sledećeg uzraza:

$$P_i = P(x_{i1}) \cdot P(x_{i2}) \cdots P(x_{in}) = \prod_{j=1}^{n} P(x_{ij}) \qquad (5.2.9.)$$

Pošto je P_i pouzdanost *i*-te grane i pošto postoji paralelna veza od *m* ovakvih grana, dobija se izraz za pouzdanost paralelno-redne veze:

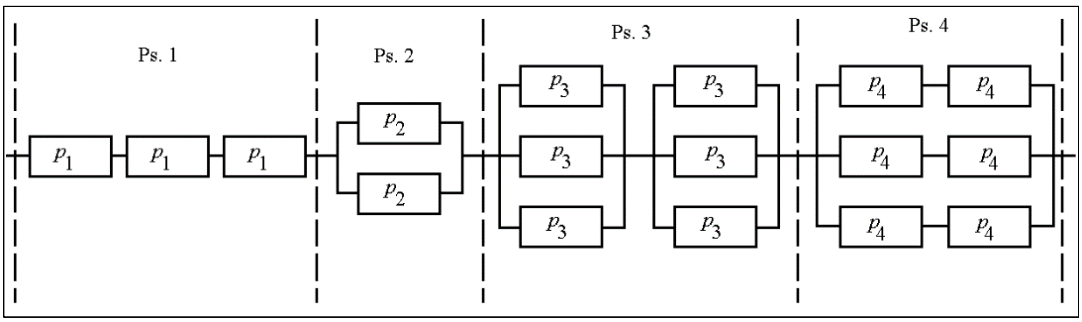
$$P = 1-(1-P_1)\cdot(1-P_2)\cdots(1-P_m) = 1-\prod_{i=1}^{m}(1-P_i) \qquad (5.2.10.)$$

Kada su svi elementi identični i kada je pouzdanost svakog od njih *p*, pouzdanost sistema može se izračunati pomoću sledećeg izraza:

$$P = 1-\prod_{i=1}^{m}(1-p^n) = 1-(1-p^n)^m \qquad (5.2.11.)$$

Primer 5.2.1. Napisati izraz za pouzdanost sistema prikazanog na slici 5.2.1. i izračunati njegovu vrednost, ako su poznate pouzdanosti njegovih elemenata i to:

$p_1 = 0{,}95$
$p_2 = 0{,}90$
$p_3 = 0{,}85$
$p_4 = 0{,}80$



Slika 5.2.1. *Sistem sa podsistemima*

Rešenje. Pouzdanost sistema u celini se računa pomoću obrasca (5.1.2.). Za konkretan slučaj on iznosi

$$P = P_1 \cdot P_2 \cdot P_3 \cdot P_4$$

gde je
P_1, P_2, P_3 i P_4 – pouzdanost podsistema Ps. 1 – Ps. 4.

Pouzdanost podsistema Ps. 1 (P_1) se računa pomoću obrasca (5.2.1.), pošto se tu radi o rednoj vezi identičnih elemenata čija je pouzdanost ista, odnosno

$$P_1 = p_1^n$$

gde je
$n = 3$ – broj redno povezanih elemenata podsistema Ps. 1.

Sledi

$$P_1 = p_1^3$$

Pouzdanost podsistema Ps. 2 (P_2) se računa pomoću obrasca (5.2.5.), pošto se tu radi o paralelnoj vezi identičnih elemenata čija je pouzdanost ista, odnosno

$$P_2 = 1 - (1 - p_2)^n$$

gde je
$n = 2$ – broj paralelno povezanih elemenata podsistema Ps. 2.

Sledi

$$P_2 = 1 - (1 - p_2)^2$$

odnosno

$$P_2 = 2 \cdot p_2 - p_2^2$$

Pouzdanost podsistema Ps. 3 (P_3) se računa pomoću obrasca (5.2.8.), pošto se tu radi o redno-paralelnoj vezi identičnih elemenata čija je pouzdanost ista, odnosno

$$P_3 = [1 - (1 - p_3)^m]^n$$

gde je
$n = 2$ – broj redno vezanih grupa elemenata podsistema Ps. 3, i
$m = 3$ – broj paralelno vezanih elemenata podsistema Ps. 3.

Sledi

$$P_3 = [1 - (1 - p_3)^3]^2$$

odnosno

$$P_3 = 9 \cdot p_3^2 - 18 \cdot p_3^3 + 15 \cdot p_3^4 - 6 \cdot p_3^5 + p_3^6$$

Pouzdanost podsistema Ps. 4 (P_4) se računa pomoću obrasca (5.2.11.), pošto se tu radi o paralelno-rednoj vezi identičnih elemenata čija je pouzdanost ista, odnosno

$$P_4 = 1 - (1 - p_4^n)^m$$

gde je
$n = 2$ – broj redno vezanih elemenata u svakoj grani podsistema Ps. 4, i
$m = 3$ – broj paralelnih grana podsistema Ps. 4.

Sledi

$$P_4 = 1 - (1 - p_4^2)^3$$

odnosno

$$P_4 = 3 \cdot p_4^2 - 3 \cdot p_4^4 + p_4^6$$

Shodno tome, pouzdanost sistema u celini iznosi (obrazac 5.1.2.):

$$P = p_1^3 \cdot [1 - (1 - p_2)^2] \cdot [1 - (1 - p_3)^3]^2 \cdot [1 - (1 - p_4^2)^3]$$

odnosno

$$P = p_1^3 \cdot (2 \cdot p_2 - p_2^2) \cdot (9 \cdot p_3^2 - 18 \cdot p_3^3 + 15 \cdot p_3^4 - 6 \cdot p_3^5 + p_3^6) \cdot (3 \cdot p_4^2 - 3 \cdot p_4^4 + p_4^6)$$

Kada se zamene vrednosti, dobija se

$$P = 0{,}95^3 \cdot [1 - (1 - 0{,}90)^2] \cdot [1 - (1 - 0{,}85)^3]^2 \cdot [1 - (1 - 0{,}80^2)^3]$$

$$P = 0{,}8037467$$

5.3. POSTAVKA ZADATAKA OPTIMALNOG REZERVIRANJA

Problem optimalnog rezerviranja ima odgovarajuću primenu u praksi. Shodno tome, određeni zadaci optimalnog rezerviranja se mogu tipizirati i svesti na oblik takozvanih standardnih zadataka. Da bi se postavka datih standardnih zadataka mogla razumeti, u daljem tekstu su oni objašnjeni pomoću jednostavnijih primera kod kojih se pouzdanost sistema, odnosno podsistema može prikazati pomoću sledećeg obrasca:

$$P_i(x_i) = 1 - q_i^{x_i+1} \qquad (5.3.1.)$$

gde je

x_i – broj rezervnih elemenata sistema (podsistema),

q_i – nepouzdanost, odnosno verovatnoća otkaza *i*-tog podsistema,

$i = 1, 2, ..., n$ – broj podsistema složenog sistema.

Pošto između pouzdanosti i nepouzdanosti sistema (podsistema) postoji relacija $q_i = 1 - p_i$, izraz (5.3.1.) može da se napiše u sledećem obliku:

$$P_i(x_i) = 1 - (1 - p_i)^{x_i+1} \qquad (5.3.2.)$$

gde je

p_i – pouzdanost *i*-tog podsistema.

Zadatak se svodi na određivanju potrebnog broja rezervnih elemenata (uz jedan radni element) sistema, odnosno podsistema (x_i) koji obezbeđuju potrebni nivo pouzdanosti rada sistema u celini.

5.3.1. Matematički model zadataka optimalnog rezerviranja

U slučaju jednog ograničavajućeg faktora, matematički model zadataka optimalnog rezerviranja se definiše na sledeći način: potrebno je odrediti takav broj rezervnih elemenata za svaki podsistem, odnosno vektor $X = (x_1, x_2, \cdots, x_n)$ koji će obezbediti određenu vrednost verovatnoće bezotkaznog rada sistema, ali uz minimalne troškove njegove realizacije.

Funkcija cilja ima sledeći oblik:

$$\min C(X) = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} c_i \cdot x_i \qquad (5.3.3.)$$

odnosno

$$\min C(X) = c_1 \cdot x_1 + c_2 \cdot x_2 + \cdots + c_n \cdot x_n \qquad (5.3.4.)$$

Ograničenje se definiše pomoću sledećeg izraza:

$$P(X) = \prod_{i=1}^{n} p_i(x_i) \quad \geq P_0 \qquad (5.3.5.)$$

odnosno

$$P(X) = p_1(x_1) \cdot p_2(x_2) \cdot \cdots \cdot + p_n(x_n) \geq P_0 \qquad (5.3.6.)$$

Takođe, u slučaju jednog ograničavajućeg faktora, matematički model zadataka optimalnog rezerviranja može da se definiše i na sledeći (obrnuti) način: potrebno je odrediti takav broj rezervnih elemenata za svaki podsistem, odnosno vektor $X = (x_1, x_2, \cdots, x_n)$ koji će obezbediti maksimalnu vrednost verovatnoće bezotkaznog rada sistema, pri čemu su troškovi njegove realizacije ograničeni (ne prelaze unapred određenu vrednost).

Funkcija cilja ima sledeći oblik:

$$\max P(X) = \prod_{i=1}^{n} p_i(x_i) \qquad (5.3.7.)$$

odnosno

$$\max P(X) = p_1(x_1) \cdot p_2(x_2) \cdot \cdots \cdot + p_n(x_n) \qquad (5.3.8.)$$

Ograničenje je sledeće:

$$C(X) = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} c_i \cdot x_i \leq C_0 \qquad (5.3.9.)$$

odnosno

$$C(X) = c_1 \cdot x_1 + c_2 \cdot x_2 + \cdots + c_n \cdot x_n \leq C_0 \qquad (5.3.10.)$$

U slučaju većeg broja ograničavajućih faktora, matematički model zadataka optimalnog rezerviranja može da se definiše na sledeći način: potrebno je odrediti takav broj rezervnih elemenata za svaki podsistem, odnosno vektor $X = (x_1, x_2, \cdots,$

x_n) koji će obezbediti maksimalnu vrednost verovatnoće bezotkaznog rada sistema, pri čemu su resursi od kojih zavisi njegova realizacija ograničeni (ne prelaze unapred određenu vrednost).

Funkcija cilja ima sledeći oblik:

$$\max P(X) = \prod_{i=1}^{n} p_i(x_i) \qquad (5.3.11.)$$

odnosno

$$\max P(X) = p_1(x_1) \cdot p_2(x_2) \cdots\cdots + p_n(x_n) \qquad (5.3.12.)$$

Ograničenja su sledeća:

$$C(X) = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} c_i \cdot x_i \leq C_0 \text{ (ograničeni troškovi)}$$

$$G(X) = \sum_{i=1}^{n} G_i(x_i) = \sum_{i=1}^{n} g_i \cdot x_i \leq G_0 \text{ (ograničena masa)}$$

$$S(X) = \sum_{i=1}^{n} S_i(x_i) = \sum_{i=1}^{n} s_i \cdot x_i \leq C_0 \text{ (ograničena površina)} \qquad (5.3.13.)$$

$$V(X) = \sum_{i=1}^{n} V_i(x_i) = \sum_{i=1}^{n} V_i \cdot x_i \leq C_0 \text{ (ograničena zapremina)}$$

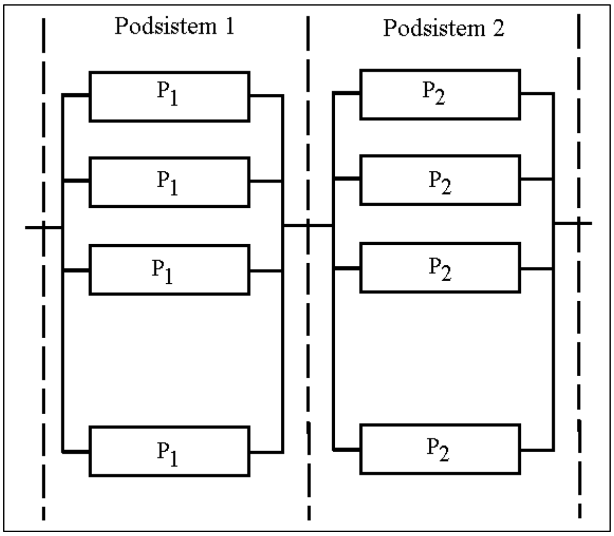
$$\vdots$$

5.3.2. Metode rešavanja zadataka optimalnog rezerviranja

Zadaci optimalnog rezerviranja se mogu rešiti primenom više metoda, kao što su:

- metoda linearnog programiranja (naročito kod postojanja jednog ograničavajućeg faktora),
- metoda Lagranžovih množitelja,
- gradijentna metoda, i
- metoda dinamičkog programiranja.

Primer 5.3.1. Modeliranje sistema sa zahtevom za minimalnim troškovima njegove realizacije i sa traženom pouzdanošću: razmatrani sistem se sastoji od dva podsistema (slika 5.3.1.). Elementi imaju sledeću pouzdanog rada p_1 = 0,8 i p_2 = 0,6. Potrebno je odrediti takav broj rezervnih elemenata za svaki podsistem, odnosno vektor $X = (x_1, x_2)$ koji će obezbediti ukupnu pouzdanost rada sistema u celini od $P_0 \geq 0{,}99$, uz minimalne troškove njegove realizacije, pri čemu cene pojedinih elemenata iznose c_1 = 2 n.j. i c_2 = 5 n.j.



Slika 5.3.1. *Sistem sa dva podsistema*

Rešenje. Matematički model datog problema može da se definiše pomoću obrazaca (5.3.3.) – (5.3.6.), odnosno za konkretan slučaj se može napisati:

Funkcija cilja

$$\min C(X) = c_1 \cdot x_1 + c_2 \cdot x_2 \qquad (5.3.14.)$$

Ograničenje

$$P(X) = p_1(x_1) \cdot p_2(x_2) \geq P_0 \qquad (5.3.15.)$$

U daljem, fokus je na ograničenju koji treba modifikovati korišćenjem obrasca (5.3.2.), pri čemu obrazac (5.3.15.) dobija sledeći oblik:

$$P(X) = [1-(1-p_1)^{x_1+1}] \cdot [1-(1-p_2)^{x_2+1}] \geq P_0 \qquad (5.3.16.)$$

Verovatnoće bezotkaznog rada svakog od podsistema, zavisno od njihovog broja rezervnih elemenata (x_i), mogu biti određene pomoću odgovarajuće tabele (tabela 5.3.1.), pri čemu se za proračun koristi obrazac (5.3.2.).

Tabela 5.3.1. *Verrovatnoće bezotkaznog rada svakog podsistema, zavisno od njihovog broja rezervnih elemenata*

x_i	$p_1(x_1)$	$p_2(x_2)$
0	0,8	0,6
1	0,96	0,84
2	0,992	0,936
3	0,9984	0,9744
4	0,99968	0,98976
5	0,999936	0,995904
6	0,9999872	0,9983616
7	0,99999744	0,99934464

Iz tabele 5.3.1. treba odrediti koliko prvom, a koliko drugom podsistemu treba dodati rezervnih elemenata da bi pouzdanost rada celog sistema bila veća od 0,99. Pošto pouzdanost sistema mora biti veća od 0,99, pouzdanost i prvog i drugog podsistema datog sistema mora takođe biti veća od 0,99, pošto njihov proizvod (izraz (5.3.15.)) treba da bude veći od zadate vrednosti (0,99).

Postupak određivanja potrebnog broja rezervnih elemenata može biti urađen na dva načina, pri čemu će se dobiti dva rešenja. Optimalno rešenje je ono koje zadovoljava funkciju cilja, odnosno ono kod koje su troškovi niži.

Prvo rešenje se dobija kada se iz tabele 5.3.1. izabere broj rezervnih elemenata prvog podsistena (x_1) za koje je pouzdanost datog podsistema veća od 0,99 ($p_1(x_1) \geq 0{,}99$). Iz tabele 5.3.1. se vidi da je to slučaj za $x_1 = 2$, jer je onda $p_1(2) = 0{,}992 \geq 0{,}99$. Broj rezervnih elemenata drugog podsistema (x_2) se dobija tako što se prvo računa najmanja pouzdanost datog podsistema $p_2(x_2)$ pomoću obrasca (5.3.15.). Sledi

$$p_2(x_2) \geq \frac{P_0}{p_1(x_1)} \tag{5.3.17.}$$

odnosno

$$p_2(x_2) \geq \frac{0{,}99}{0{,}992} \geq 0{,}99798387$$

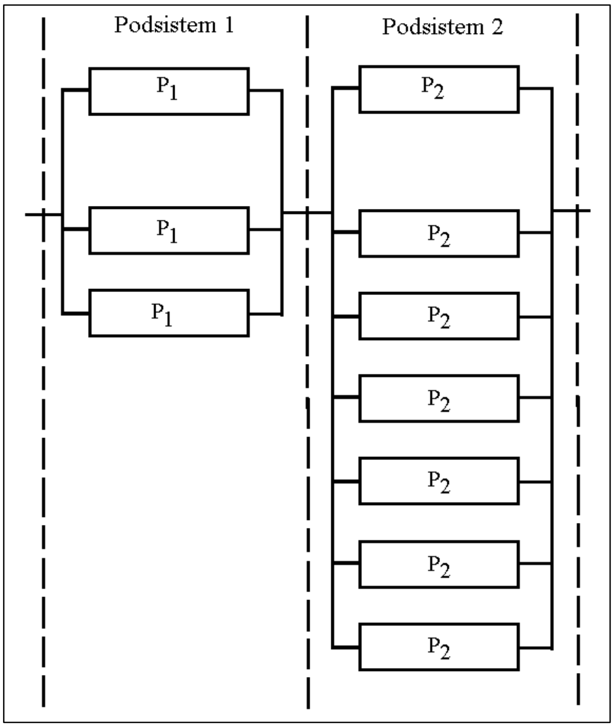
Nakon ovog proračuna, u tabeli 5.3.1. se u zadnjoj koloni traži prva veća vrednost pouzdanosti drugog podsistema koji ovde iznosi $p_2(x_2) = 0{,}9983616$. Broj potrebnih rezervnih elemenata drugog podsistema iznosi $x_2 = 6$ (tabela 5.3.1.).

Shodno ovome, prvo rešenje je

$X = (x_1, x_2) \Rightarrow X = (2, 6)$ – broj rezervnih elemenata prvog podsistema je 2, a drugog 6 (uz po 1 radni – slika 5.3.2.),

$P(X) = p_1(x_1) \cdot p_2(x_2) \geq P_0 \Rightarrow P(X) = 0{,}992 \cdot 0{,}9983616 = 0{,}9903747 \geq 0{,}99$ – pouzdanost sistema je veća od zahtevane,

$C(X) = c_1 \cdot x_1 + c_2 \cdot x_2 \Rightarrow C(X) = 2 \cdot 2 + 5 \cdot 6 = 34 \text{ n.j.}$ – troškovi realizacije sistema su 34 n.j.



Slika 5.3.2. *Prvo rešenje pouzdanosti sistema*

Kako je rečeno, drugo rešenje se dobija kada se iz tabele 5.3.1. izabere broj rezervnih elemenata drugog podsistena (x_2) za koje je pouzdanost datog podsistema veća od 0,99 ($p_2(x_2) \geq 0{,}99$). Iz tabele 5.3.1. se vidi da je to slučaj za $x_2 = 5$, jer je tada $p_2(x_2) = 0{,}995904 \geq 0{,}99$. Broj rezervnih elemenata prvog podsistema (x_1) se dobija iz obrasca (5.3.15.). Sledi

$$p_1(x_1) \geq \frac{P_0}{p_2(x_2)} \qquad (5.3.18.)$$

odnosno

$$p_1(x_1) \geq \frac{0{,}99}{0{,}995904} \geq 0{,}9940717177593$$

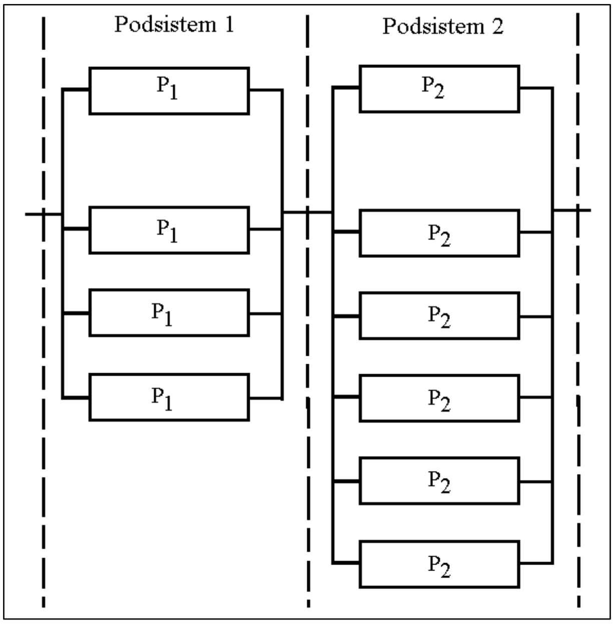
Nakon ovog proračuna, u tabeli 5.3.1. se u drugoj koloni traži prva veća vrednost pouzdanosti prvog podsistema koji ovde iznosi $p_1(x_1) = 0{,}9984$. Broj potrebnih rezervnih elemenata prvog podsistema iznosi $x_1 = 3$ (tabela 5.3.1.).

Shodno ovome, drugo rešenje je

$X = (x_1, x_2) \Rightarrow X = (3, 5)$ – broj rezervnih elemenata prvog podsistema je 3, a drugog 5 (uz po 1 radni – slika 5.3.3.),

$P(X) = p_1(x_1) \cdot p_2(x_2) \geq P_0 \Rightarrow P(X) = 0{,}9984 \cdot 0{,}995904 = 0{,}9943105536 \geq 0{,}99$ – pouzdanost sistema je veća od zahtevane,

$C(X) = c_1 \cdot x_1 + c_2 \cdot x_2 \Rightarrow C(X) = 2 \cdot 3 + 5 \cdot 5 = 31 \text{ n. j.}$ – troškovi realizacije sistema su 31 n.j.



Slika 5.3.3. *Drugo rešenje pouzdanosti sistema*

Na osnovu proračuna usvaja se drugo rešenje kao optimalno jer je cena njegove realizacije niža u odnosu na prvo rešenje, što je i osnovni zahtev funkcije cilja zadatka. Takođe, iako oba rešenje daju pouzdanost veću od 0,99 (zahtev ograničenja), drugo rešenje nudi i veću pouzdanost, što je i njegova dodatna prednost.

LITERATURA

A. Art Lew, H. Mauch, *Dynamic programming – a computational tool*, Springer, Berlin, 2007.

A. Charnes, *Optimality and degeneracy in linear programming*, Econometrica, 20(2), (1952).

A. Dax, *Linear programming via least squares*, Linear algebra and its applications, 111 (1988), 313-324.

A. Jovanović, *Operaciona istraživanja* (autorizovana predavanja), Tehnički fakultet u Boru, Bor, 2005.

A. R. Conn, *Linear Programming via a nondifferentiable penalty function*, SIAM Journal on numerical analysis, 13(1) (1976), 145-154.

A. Štajer, *Rešavanje problema minimizacije sa ograničenjima primenom kaznenih funkcija* (Master rad), Univerzitet u Novom Sadu, Prirodno-matematički fakultet, Departman za matematiku i informatiku, Novi Sad, 2014.

A. Wren, *Computers in transport planning and operation*, Ian Allan, London, 1971.

B. D. Bounday, *Basic linear programming*, Edvard Arnold, Baltimore, 1984.

B. L. Golden, A. A. Asad, *Vehicle routing: methods and studies*, North Holland, Amsterdam, 1988.

B. Korte, J. Vygen, *Combinatorial optimization*, Springer-Verlag, Berlin, Heidelberg, 2002.

B. Render, R. M. Stair, M. Hanna, *Quantitative analysis for management 9e*, Pearson Prentice Hall, Pearson Education, New Jersy, 2006.

D. Cvetković, M. Čangalović, Đ. Dragošija, V. Kovačević-Vujčić, S. Simić, J. Vuleta, Kombinatorna optimizacija, Društvo operacionih istraživača, Beograd, 1996.

D. Letić, *Operaciona istraživanja*, Univerzitet u Novom Sadu, Novi Sad, 2003.

D. Lipovac, D. Letić, Metode o*peracionih istraživanja*, Univerzitet u Novom Sadu, Zrenjanin, 1999.

D. Milčić, *Pouzdanost mašinskih sistema*, Mašinski fakultet, Niš, 2005.

D. M. Ryan, C. Hjorring, F. Glover, *Extension of the petal method for vehicle routing*, Journal of the operational research societe, 44 (1993), 289-296.

D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Athena, 2004.

D. Vigo, *A heuristic algorithm for the asimmetric capacitated vehicle routing problem*, European journal of the operational research, 89 (1996), 108-126.

Ed. T. Terlaky, *Interior point methods of mathematical programming*, Kluwer Academic Publication, Dodrecht, Boston, London, 1996.

E. M. L. Beale, *A method of solving linear programming problems when some but not all of the variables must take integer values*, Statistical techniques research group, Technical Report No. 19, Princeton, 1958.

E. M. L. Beale, *Branch-and-bound methods for mathematical programming system*, Annals of discrete mathematics, 5 (1979), 201-219.

F. L. Hitchcock, *The distribution of a product from several sources to numerous localitites*, Journal of mathematics and physics, 20 (1941), 224-230.

G. B. Dantzig, *Programming of interdependent activities, mathematical model*, Econometrica, 17 (1949), 200-211.

G. B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, 1963.

G. L. Nemhauser, L. A. Wolsey, *Integer and combinatorial optimization*, John Wiley&Sons, New York, 1988.

H. P. Taha, *Integer programming: theory, applications, and computations*, Academic Press, Orlando, 1975.

H. P. Williams, *Model building in mathematical programming*, 3rd. ed. Wiley, New York, 1990.

I. Beker, G. Ivanović, D. Stanivuković, *Pouzdanost tehničkih sistema*, Fakultet tehničkih nauka, Novi Sad, 2010.

I. M. Jovanović, P. S. Stanimirović, *Transportni zadatak na putnoj mreži*, Majska konferencija o strategijskom menadžmentu (2007), 253-262.

J. B. Bramel, D. Simchi-Levi, *A location based heuristic for general routing problems*, Operations research, 43 (1995), 649-660.

J. Czyzyk, S. Mehrotra, S. J. Wright, *PCx User Guide*, Optimization Thechnology Center, Technical Report 96/01 (1996) 1-21.

J. Egervary, *Matrixok combinatorius tulajdonsagairol*, Math.Fiz. Lopak, 1931.

J. Gondzio, HOPDM (Version 2.12), *A fast LP solver based on a primal-dual interior point method*, European journal of operations research, 85 (1995), 221-225.

J. J. Forrest, D. Goldfarb, *Steepest-edge simplex algorithms for linear programming*, Mathematical programming, 57 (1992), 341-374.

J. Petrić, *Operaciona istraživanja, knjiga prva*, Savremena Administracija, Beograd, 1974.

J. Petrić, Z. Kojić, L. Šarenac, *Operaciona istraživanja, zbirka rešenih zadataka I i II*, Privredno finansijski vodič, Beograd, 1979.

J. P. Ignizio, *Linear programming in single-multiple-objective systems*, Englewood Cliffs: Prentice Hall, 1982.

J. Renaud, F. F. Boctor, G. Laporte, *An improved petal heuristic for the vehicle routing problem*, Journal of the operational research societe, 47 (1996), 329-336.

J. W. Baumol, A.S., Blinder, *Microeconomics, principles and policy,* The Dryden Press, Fort Worth, 1994.

L. A. Wolsey, *Integer programming*, John Wiley&Sons, New York, 1998.

L. D. Pyle, *The generalized inverse in linear programming. Basic structure*, SIAM Journal on applied mathematics, 22 (1972), 335-355.

L. D. Pyle, R. E. Cline, *The genalized inverse in linear programming – interior gradient projection methods*, SIAM Journal on applied mathematics, 24 (1973), 511-534.

L. D. Pyle, T*he weighted genalized inverse in nonlinear programming - active set selection using a variable-metric generalization of the simplex algorithm,* Lecture notes in economics and mathematical system, Austin, Texas, September 13-15, 1977.

L. G. Khachian, *A Polinomial algorithm in linear programming*, Doklady Akademii Nauk SSSR, 244(5) (1979), 1093-1096.

L. V. Kantorovich, *Matematicheskie metodi v organizacii i planirovanii proizvodstva*, Izd. LGU, 1939.

M. A. Bhatti, *Practical optimization with MATHEMATICA applications*, Springer Verlag Telos, Berlin, 2000.

M. Bazaraa, C. M. Shetty, *Nonlinear programming - theory and algorithms*, John Wiley&Sons, New York, 1979.

M. Dorigo, G. Di Caro, L. M. Gambardella, *Ant algorithms for discrete optimization*, Artificial life, 5(2) (1999), 137-172.

M. Kojima, N. Megiddo, T. Noma, A. Yoshise, *A unified approach to interior-point algorithms for linear complementarity problems*, Lecture notes in computer science 538, Springer-Verlag, Berlin, 1991.

M. Martić, D. Makajić-Nikolić, G. Popović, B. Panić, *Operaciona istraživanja 1, zbirka zadataka*, Fakultet organizacionih nauka, Beograd, 2006.

M. Tasić, P.S. Stanimirović, I.P. Stanimirović, M.D. Petković, N.V. Stojković, *Some useful MATHEMATICA teaching examples*, Facta Universitatis (Niš) Series electronics and energetics, 18(2), August 2005, 329-344.

M. W. Carter, C. C. Price, *Operations research, a practical introduction*, CRC Press LLC, N.W.Corporate Blvd., Boca Raton, 2000.

N. Christofides, *The vehicle routing problem*, RAIRO (Recherche operationnelle), 10 (1976), 55-70.

N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), 373-395.

N. Petković, *Matematički modeli optimizacije poslovnih procesa* (Doktorska disertacija), Univerzitet "Džon Nezbit" Beograd, Fakultet za Menadžment – Zaječar, Zaječar, 2016.

N. Vujanović, *Teorija pouzdanosti tehničkih sistema*, Vojnoizdavački Novinski Centar, Beograd, 1990.

O. Todorović, M. Pešić, *Operaciona istraživanja*, Zbirka rešenih problema, III izdanje, Ekonomski fakultet u Nišu, Niš, 2003.

O. Todorović, *Operaciona istraživanja*, Ekonomski fakultet u Nišu, Niš, 2004.

P. Bonami, L. Biegler, A. Conn, G. Cornuejols, I. Grossmann, C. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wachter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete optimization, 5 (2008), 186–204.

P. S. Stanimirović, G. V. Milovanović, I. M. Jovanović, *Primena linearnog i celobrojnog programiranja*, Prirodno-matematički fakultet u Nišu, Niš, 2008.

P. S. Stanimirović, N. V. Stojković, M. D. Petković, *Matematičko programiranje*, Prirodno-matematički fakultet u Nišu, Niš, 2007.

P. S. Stanimirović, *Operaciona istraživanja*, Autorizovana predavanja, Niš, 2009.

R. Bellman, *Dynamic programming*, Princeton University, Princeton, 1957.

R. Bixby, *Implementing the simplex method; the initial basis*, ORSA Journal on computing, 4 (1992), 267-284.

R. E. Gomory, *An algorithm for integer solutions to linear programs*, Princeton IBM mathematics research project, Technical report No. 1, 1958.

R. E. Gomory, *An all-integer programming algorithm*, Industrial sheduling, Englewood Cliffs, 1963.

R. J. Vanderbei, LOQO: *An interior-point code for quadratic programming*, Technical Report SOR-94-15, Department of civil engineering and operations research, Princeton University, Princeton, 1994.

R. K. Martin, *Large scale linear and integer programming*, Kluwer Academic, Boston, 1966.

R. Stanojević, *Linearno programiranje,* Institut za ekonomiku industrije, Beograd, 1999.

S. Borović, M. Milićević, *Zbirka zadataka iz odabranih oblasti operacionih istraživanja*, VIZ–Sektor za školstvo, obuku, naučni i izdavačku delatnost, Uprava za naučnu i izdavačku delatnost, Vojna akademija, Beograd, 2001.

S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, 2006.

S. Drewes, S. Ulbrich, *Mixed integer second order cone programming*, IMA hot topics workshop, mixed-integer nonlinear optimization: algorithmic advances and applications, November 17-21, 2008.

S. F. Hillier, G.J., Lieberman, *Introduction to operations research*, McGraw-Hill, New York, 1986.

S. Gass, *Linear Programming: Method and applications*, McGraw-Hill, New York, 1985.

S. J. Wright, *Primal-dual interior-point methods*, SIAM, Philadelphia, 1997.

S. Krčevinac, M. Čangalović, V. Kovačević-Vujčić, M. Martić, M. Vujočević, *Operaciona istraživanja 1,* Fakultet Organizacionih Nauka, Beograd, 2006.

S. L. Campbell, C. D. Meyer, *Generalized inverses of linear transformations*, Pitman, London, San Francisco, Melbourne, 1979.

S. Lin, B. W. Kernighan, *An effective heuristic algorithm for the traveling salesman problem*, Operational research, 21 (1973), 498-516.

S. Vukadinović, S. Cvejić, *Matematičko programiranje*, Univerzitet u Prištini, Priština, 1996.

S. Wolfram, *Mathematica: a system for doing mathematics by computer*, Addison-Wesley Publishing Co, Redwood City, 1991.

T. C. T. Kotiah, D. I. Steinberg, *Occurences of cycling and other phenomena arising in a class of linear programming models*, Communications of the ACM, 20 (1977), 107-112.

T. Ereg, *Problemi transporta i analiza osetljivosti* (Master rad), Univerzitet u Novom Sadu, Prirodno-matematički fakultet, Departman za matematiku i informatiku, Novi Sad, 2013.

T. Kiš, M. Čeleg, D. Vugdelija, O. Sedlak, *Kvantitativni metodi u ekonomiji*, Ekonomski fakultet u Subotici, Subotica, 2005.

T. M. Jovanović, *Operaciona istraživanja*, Mašinski fakultet u Beogradu, Beograd, 1998.

V. Ivanović, *Pravila za proračun potrebnog broja transportnih sredstava*, Vojnoizdavački glasnik, Sveska 1-3 (1940), 1-10.

V. Klee, G. L. Minty, *How good is the simplex method?*, O. Shisha, ed., Inequalities III, Academic Press, New York, 1972, pp. 159-175.

Y. Zang, *User's guide to LIPSOL*, Department of mathematics and statistics, University of Maryland, Baltimore County, Baltimore, July 1995.

W. L. Winston, *Operations research*, Duxbury Press, Belmont, 1993.

W. L. Winston, M. Venkataramanan, *Introduction to mathematical programming*, Thomson Brooks/Cole, Australia, 2002.

W. P. Adams, H. D. Sherali, *Linearization strategies for a class of zero-one mixed integer programming problems*, Operations research, 38(2) (1990), 217-226.

W. R. Stewart, B. L. Golden, *A Lagrangean relaxation heuristic for vehicle routing*, European journal of the operational research, 15 (1984), 84-88.

Dr Dejan Bogdanović, redovni profesor Tehničkog fakulteta u Boru je rođen 1965. godine u Boru. Diplomske i Magistarske studije je završio na Tehničkom fakultetu u Boru 1990., odnosno 1992. godine, a doktorsku disertaciju je odbranio 2001. godine na Rudarsko-geološkom fakultetu u Beogradu Univerziteta u Beogradu.

Nakon diplomiranja radio je na Tehničkom fakultetu u Boru kao asistent (1990-1994. god.), zatim u RTB Boru – pogon Jama (1995-1999. god), a onda je prešao u Institut za bakar Bor, gde je obavljao razne funkcije, od projektanta, preko Koordinatora odeljenja za podzemnu eksploataciju, do Pomoćnika Direktora za geologiju i rudarstvo. Od 2005. godine do 2008. godine je radio na Fakultetu za menadžment u Zaječaru – Megatrend Univerzitet. Od 2008. godine radi na Tehničkom fakultetu u Boru na katedri za inženjerski menadžment. U zvanje redovnog profesora izabran je 2018. godine. Angažovan je na predmetima: Operaciona istraživanja 1, Upravljanje procesima rada, Upravljanje promenama, Stručna praksa (osnovne akademske studije) i Portfolio projekt menadžment (master akademske studije).

Autor je oko 150 radova od kojih su 13 objavljeni u međunarodnim naučnim časopisima (M21-23), 2 monografije i 3 udžbenika. Učestvovao je na realizaciji više desetina domaćih i međunarodnih projekata (Kanada, Peru, Gvineja, Iran Crna Gora i dr.). Dugogodišnji je član Saveza inženjera i tehničara Srbije i Udruženja nastavnika inženjerskog menadžmenta.

Dr Ivan Jovanović vanredni profesor Tehničkog fakulteta u Boru je rođen 1965. godine u Pirotu. Diplomu o višoj stručnoj spremi, smer prenos i obrada informacija, stiče 1990. godine na Elektronskom fakultetu u Nišu Univerziteta u Nišu. U periodu od 2002. do 2010. godine pohađa i završava osnovne, magistarske i doktorske studije na Tehničkom fakultetu u Boru Univerziteta u Beogradu. Diplomu o stečenom visokom obrazovanju stiče 2004. godine pod nazivom diplomirani inženjer industrijskog menadžmenta. Diplomu o stečenom akademskom nazivu magistra tehničkih nauka iz oblasti industrijkog menadžmenta stiče 2006. godine. Diplomu o stečenom naučnom stepenu doktora nauka iz oblasti inženjerskog menadžmenta stiče 2010. godine.

U periodu od 1991. do 2004. god. radio je u MUP Republike Srbije, privatnom sektoru i TV Pirot. Od 2005. godine radi na Tehničkom fakultetu u Boru Univerziteta u Beogradu na katedri za Inženjerski menadžment najpre u zvanju asistenta pripravnika (2005-2007.), zatim u zvanju asistenta (2007-2010.). U zvanje docenta izabran je 2010. god, a u zvanje vanrednog profesora 2015. godine. Trenutno je angažovan na predmetima osnovnih studija: Preduzetništvo, Operaciona istraživanja 2, Teorija pouzdanosti, kao i na predmetu doktorskih studija Menadžment znanjem. Autor i koautor je preko 70 naučnih radova i saopštenja od kojih su 9 objavljeni u međunarodnim naučnim časopisima iz kategorije M20, 2 monografije i 5 udžbenika, više od 20 hetero citata radova iz kategorije M20. Učestvovao je na realizaciji većeg broja domaćih i međunarodnih projekata. Imao je veliko angažovanje kao mentor i član komisija, i solidan stručno-profesionalni doprinos akademskoj i široj zajednici. Dugogodišnji je član Saveza udruženja i tehničara Srbije, i Udruženja nastavnika inženjerskog menadžmenta.