Simpleks metoda

Jajac, Luka

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:644160

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

SIMPLEKS METODA

Rijeka, rujan 2024.

Luka Jajac
0069092529

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

SIMPLEKS METODA

Mentor: izv. prof. dr. sc. Ivan Dražić

Rijeka, rujan 2024.

Luka Jajac
0069092529

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 14.03.2024.

Zavod: Zavod za matematiku, fiziku i strane jezike
Predmet: Inženjerska matematika ET

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Luka Jajac (0069092529)**
Studij: Sveučilišni prijediplomski studij elektrotehnike (1030)

Zadatak: **Simpleks metoda / Simplex method**

Opis zadatka:

U radu je potrebno definirati apstraktni problem linearnog programiranja i detaljno objasniti simpleks metodu za rješavanje takvih problema. Potrebno je objasniti sve varijante simpleks metode u ovisnosti o tipu funkcije cilja i tipu ograničenja. U drugom dijelu rada treba opisati problematiku analize osjetljivosti i njenu povezanost sa simpleks metodom. U završnom dijelu rada simpleks metodu potrebno je primijeniti na nekoliko konkretnih problema s naglaskom na primjene u inženjerstvu, a dobivena rješenja provjeriti korištenjem odgovarajuće softverske podrške.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
izv. prof. dr. sc. Ivan Dražić

Predsjednik povjerenstva za
završni ispit:
prof. dr. sc. Dubravko Franković

IZJAVA

Sukladno članku 7. Stavku 1. Pravilnika o završnom radu, završnom ispitu i završetku sveučilišnih prijediplomskih studija Tehničkog fakulteta Sveučilišta u Rijeci od 4. travnja 2023., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 20. ožujka 2024.

Rijeka, 6. rujna 2024.

Luka Jajac

Luka Jajac

Prije svega, neizmjerno hvala mojoj obitelji na bezuvjetnoj ljubavi, podršci i strpljenju tijekom svih ovih godina. Hvala vam na svemu što ste mi pružili te što sam zbog vas tu gdje jesam.

Posebna zahvala mom mentoru, izv. prof. dr. sc. Ivanu Dražiću, na stručnom vodstvu i motivaciji tijekom izrade ovog rada. Njegovo strpljenje, posvećenost a opet toplina i pristupačnost su uvelike doprinijeli mom uspjehu.

Zahvala i mojoj dragoj Petri, čija je podrška, razumijevanje i vjera u mene bila neizmjerna tijekom cijelog ovog procesa.

Veliko hvala Josipu, koji je dijelio sa mnom čitavo vrijeme iste muke. Također, veliko hvala svim mojim prijateljima i kolegama koji su mi svojim savjetima, podrškom, pozitivnom energijom i samim prisustvom pomogli da ovaj rad bude što kvalitetniji. Hvala svima!

Sadržaj

1. Uvod

U današnjici koju obilježava konstantan rast kompleksnosti i dostupnosti podataka, potreba za učinkovitijim metodama donošenja odluka postaje sve izraženija. U tom kontekstu, linearno programiranje se ističe kao veoma koristan koncept za optimizaciju resursa i donošenje optimalnih odluka u širokom spektru područja, od ekonomije i financija, preko inženjerstva i logistike, do medicine i nutricionizma.

Kao jedna od fundamentalnih grana matematičkog programiranja, linearno programiranje se bavi optimizacijom linearne funkcije cilja, uzimajući u obzir zadana linearna ograničenja. Ovaj pristup omogućuje modeliranje i rješavanje širokog spektra problema u kojima je potrebno pronaći najbolji način za distribuciju ograničenih resursa kako bi se maksimizirao profit ili minimizirali troškovi.

U srži linearne optimizacije nalazi se simpleks metoda, revolucionarni algoritam koji je u drugoj polovici 20. stoljeća transformirao područje optimizacije. Razvijena od strane matematičara Georgea B. Dantziga, simpleks metoda je omogućila rješavanje složenih problema linearne optimizacije s kojima se do tada nije moglo učinkovito nositi. Ova metoda je omogućila razvoj naprednijih algoritama i softverskih alata koji danas predstavljaju nezaobilazan alat u rukama analitičara, inženjera i donositelja odluka diljem svijeta.

Ovaj rad predstavlja sveobuhvatan i detaljan pregled simpleks metode, prateći njenu povijest, matematičke temelje, principe rada i praktičnu primjenu. Polazeći od samih početaka linearne optimizacije, rad ocrtava povijesni put simpleks metode i ističući njenu važnost na evoluciju matematičkog modeliranja i optimizacije.

Rad se ne zaustavlja samo na teorijskom okviru, već zaranja u srž simpleks metode, objašnjavajući matematičku pozadinu problema linearne optimizacije, s posebnim naglaskom na koncept konveksnosti i njegovom značaju za razumijevanje i primjenu metode.

U radu su prikazane sve faze simpleks algoritma, od grafičkog prikaza koji vizualizira osnovne principe, do kompleksnijih koncepata poput simpleks tablica i pivotiranja koji predstavljaju srž algoritma. Specifične teme poput Charnesove M metode, koja proširuje primjenu simpleks metode na širu klasu problema, i analize osjetljivosti, koja omogućuje procjenu utjecaja parametara promjena na optimalno rješenje, dodatno produbljuju razumijevanje metoda i nude alate za rješavanje specifičnih problema.

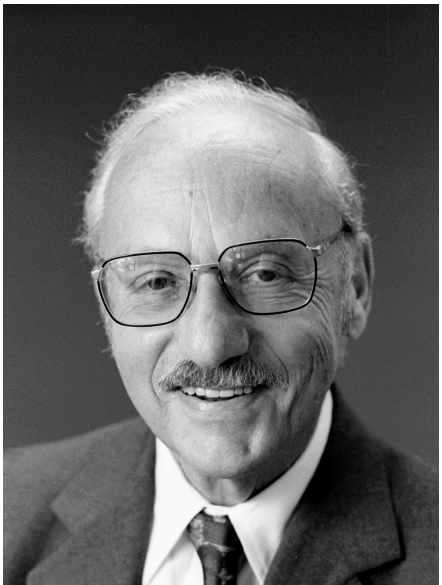
Konačno, rad demonstrira praktičnu snagu i svestranost simpleks metode kroz primjenu na konkretnom inženjerskom problemu, a pokazuje i primjenu Excel Solver-a, široko dostupnog softverskog alata baziranog na simpleks metodi.

2. Simpleks metoda u kontekstu povijesnog razvoja linearnog programiranja

Linearno programiranje je matematički pristup rješavanju problema optimizacije unutar zadanih ograničenja. Cilj je postići najbolji ishod (npr. maksimalni profit ili minimalni trošak) u nekom matematičkom modelu čija su ograničenja iskazana linearnim uvjetima. Ova grana matematike primjenjuje se za rješavanje praktičnih problema u različitim sektorima poput ekonomije, nutricionizma i organizacije rada.

Problemi optimizacije značajnije su se počeli istraživati za vrijeme Drugog svjetskog rata, pri čemu se posebno ističe ruski matematičar i ekonomist Leonid Vitalyevich Kantorovich. Kantorovich je kroz svoj rad na problemima vojne logistike prvi formalno postavio problem linearnog programiranja. O važnosti njegova doprinosa govori i činjenica da je za svoj rad u području linearnog programiranja nagrađen Nobelovom nagradom.

Iako je Kantorovich udario temelje teoriji linearnog programiranja, nije se posebno bavio algoritmima za rješavanje tih problema. U tom području pionir je bio George Bernard Dantzig koji je razvio simpleks metodu za rješavanje problema linearnog programiranja, što je i osnovna tema ovog rada.



Slika 2.1. George Bernard Dantzig. Izvor: [1]

George Bernard Dantzig rođen je 8. studenog 1914. godine u Portlandu, Oregon, SAD. Bio je sin Tobias Dantziga, matematičara, i Anje Ourisson Dantzig, lingvistice. Odrastao je u akademskom okruženju koje je poticalo njegov interes za matematiku i znanost. Dantzig je studirao na Sveučilištu Maryland, gdje je 1936. godine stekao diplomu prvostupnika matematike i fizike.

Nakon toga, nastavio je obrazovanje na Sveučilištu Michigan, gdje je 1937. godine stekao magisterij iz matematike. Tijekom Drugog svjetskog rata, Dantzig je radio kao civilni statističar za Ured za vojne zalihe (Office of the Army Air Forces), što je značajno utjecalo na njegov kasniji rad u matematičkoj optimizaciji.

Nakon rata, Dantzig je upisao doktorat na Sveučilištu Kalifornije u Berkeleyju, gdje je 1946. godine obranio doktorsku disertaciju pod mentorstvom matematičara Jerzy Neymana. Dantzig je proveo veći dio svoje karijere kao profesor na Stanfordskom sveučilištu, gdje je radio do umirovljenja. Tijekom svog života, primio je brojne nagrade i priznanja, uključujući Nacionalnu medalju za znanost 1975. godine. George B. Dantzig preminuo je 13. svibnja 2005. godine, ostavivši iza sebe trajno nasljeđe u matematici i optimizaciji, no ipak njegov najznačajniji doprinos matematici je razvoj simpleks algoritma. [2]

Istraživanja u području linearnog programiranja krajem prošlog stoljeća bila su usmjerena na poboljšanje i optimizaciju simpleks algoritma. To je rezultiralo razvojem alternativnih algoritama koji su imali potencijal da zamijene simpleks algoritam zbog svoje polinomijalne složenosti, suprotno eksponencijalnoj složenosti koju je imao simpleks algoritam. Na tom se planu prvi istaknuo Leonid Khachiyan.

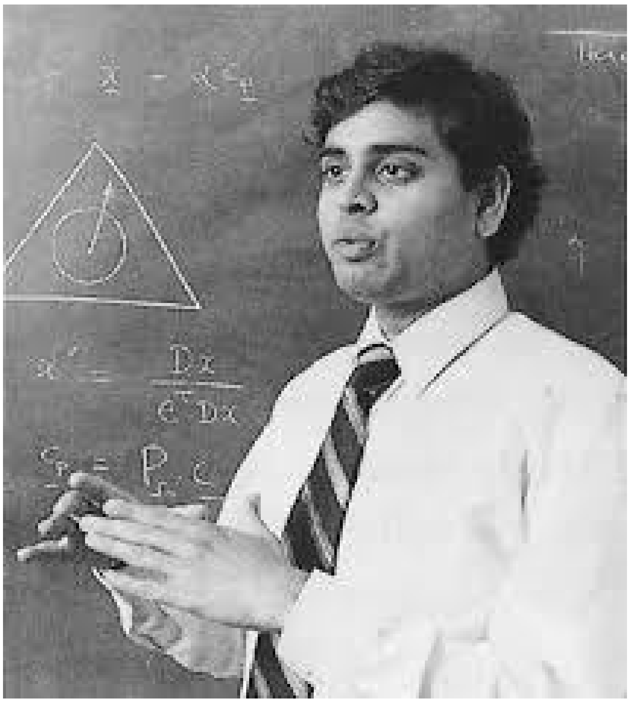
Slika 2.2. Leonid Khachiyan. Izvor: [3]

Leonid Khachiyan rođen je 3. svibnja 1952. godine u Lenjingradu (današnji Sankt Peterburg u Rusiji). Bio je sovjetski i američki matematičar poznat po svom revolucionarnom doprinosu teoriji optimizacije. Khachiyan je stekao diplomu iz primijenjene matematike na Moskovskom institutu za fiziku i tehnologiju (MIPT) 1969. godine, a zatim je doktorirao na Institutu za kontrolne probleme Sovjetske akademije znanosti.

Khachiyan je postao svjetski poznat 1979. godine kada je objavio svoj algoritam elipsoidne metode za rješavanje problema linearnog programiranja. Njegov rad je dokazao da se linearni programi mogu riješiti u polinomnom vremenu, što je imalo značajan utjecaj na područje optimizacije i računalnih znanosti. Ovaj algoritam nije bio praktično učinkovit kao Dantzigova simpleks metoda za većinu stvarnih problema, ali je imao duboke teoretske implikacije, posebno u dokazivanju kompleksnosti problema.

Godine 1989., Khachiyan se preselio u Sjedinjene Američke Države, gdje je nastavio svoj rad kao profesor na Sveučilištu Rutgers u New Jerseyju. Tamo je predavao i istraživao sve do svoje smrti 29. travnja 2005. godine. [3]

Iduća prekretnica u razvoju teorije linearnog programiranja dogodila se 1984. godine radom indijskog matematičara Narendre Karmarkara. Karmarkar je razvio algoritam polinomijalne vremenske složenosti nazvan metoda unutarnje točke. Za razliku od elipsoidne metode, ovaj algoritam se pokazao konkurentnim u odnosu na simpleks algoritam u konkretnim primjenama.



Slika 2.3. Narendra Karmarkar. Izvor: [4]

Narendra Karmarkar rođen je 1957. godine u indijskom gradu Gwalior, Madhya Pradesh. Od rane dobi pokazivao je izvanredan talent za matematiku i prirodne znanosti. Nakon završetka srednje škole, Karmarkar je upisao Indijski institut za tehnologiju (IIT) u Bombaju, gdje je 1978. godine diplomirao elektrotehniku. Njegova strast prema matematici i računalnim znanostima vodila ga je dalje na poslijediplomske studije, prvo na Kalifornijskom tehnološkom institutu (Caltech), gdje je stekao magisterij, a potom na Sveučilištu Kalifornije u Berkeleyju, gdje je 1983. godine doktorirao iz računalnih znanosti pod mentorstvom profesora Christosa Papadimitrioua.

Karmarkar je najpoznatiji po razvoju unutarnje točkaste metode za rješavanje problema linearnog programiranja, poznate kao Karmarkarov algoritam, koju je objavio 1984. godine dok je radio za Bell Laboratories. Ovaj algoritam predstavljao je revoluciju u području optimizacije, jer je bio prvi polinomni algoritam za linearno programiranje koji je u praksi bio bolji od simpleks metode, a u nekim slučajevima i značajno brži.

Nakon rada na Bell Laboratories, Karmarkar je radio na nekoliko akademskih i industrijskih pozicija, uključujući Institut za napredne studije Tata u Mumbaiju. Osim rada na linearnom programiranju, Karmarkar se bavio i istraživanjima u području superračunalstva i kvantne mehanike. Njegovi doprinosi matematici, optimizaciji i računalnim znanostima prepoznati su kroz brojne nagrade i priznanja, a posebno se ističe nagrada Kanellakis 2000. godine, koju dodjeljuje Association for Computing Machinery (ACM) i smatra se jednom od vodećih nagrada u području primijenjenog računarstva.[5]

Suvremeno linearno programiranje također obuhvaća napredne tehnike poput unaprijeđenih algoritama optimizacije, distribuiranog linearnog programiranja, primjene strojnog učenja u rješavanju linearnih problema, kao i integraciju s računalnim sustavima visokih performansi za rješavanje velikih problema u stvarnom vremenu. Osim toga, istraživanja su usmjerena na razvoj efikasnih alata i softverskih platformi za olakšavanje rada s linearnim programiranjem te na primjenu u područjima poput logistike, financija, telekomunikacija, proizvodnje i mnogim drugima.

3. Matematička formalizacija problema linearnog programiranja

Problem linearnog programiranja temelji se na precizno definiranoj linearnoj funkciji cilja i ograničenjima iskazanim putem linearne veze varijabli sustava, uz pretpostavku nenegativnosti varijabli. Ključni elementi linearnih optimizacijskih modela obuhvaćaju postavljanje cilja optimizacije, identifikaciju relevantnih varijabli, definiranje restrikcija u skladu s ograničenjima problema te pronalaženje optimalnog rješenja koje maksimizira ili minimizira ciljnu funkciju uz poštivanje zadanih ograničenja.

U ovom dijelu fokusiramo se na matematičku formalizaciju linearnog optimizacijskog modela, pri čemo se pretežno koristimo izvorom [6].

3.1. Funkcija cilja

U linearnom programiranju, funkcija cilja je temeljni element koji se koristi za optimizaciju problema. Ova linearna funkcija modelira optimizacijski kriterij, a formalno je definiramo na sljedeći način.

Definicija 3.1. *Funkcija cilja je izražena linearnom funkcijom oblika*

$$Z(x_1, x_2, ..., x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n, \tag{3.1}$$

gdje Z označava vrijednost koja se želi maksimizirati ili minimizirati, dok su $x_1, x_2, \ldots, x_n$ varijable koje se koriste za donošenje odluka. S druge strane, $a_1, a_2, \ldots, a_n$ predstavljaju koeficijente koji određuju doprinos svake varijable u postizanju cilja optimizacije.

Razmotrimo funkciju cilja kroz dva primjera

Primjer 3.1. *Pretpostavimo da imamo dva prehrambena proizvoda. Jedan proizvod je soja, a drugi proizvod je tuna. Želimo maksimizirati proteinsku vrijednost obroka. Ako je proteinska vrijednost po gramu soje 0.36 g, a proteinska vrijednost po gramu tune 0.23 g, tada je funkcija cilja:*

$$Z = 0.36x_1 + 0.23x_2, \tag{3.2}$$

gdje su:

- x_1 *količina soje u jednom obroku u gramima,*
- x_2 *količina tune u jednom obroku u gramima.*

U ovom slučaju, želimo maksimizirati funkciju cilja.

Primjer 3.1 je formuliran prema podacima s weba

Primjer 3.2. *Pretpostavimo opet da imamo dva prehrambena proizvoda (avokado i mango) te sada želimo minimizirati troškove. Znamo da je cijena po gramu avokada* 0.0066 *eura, a cijena po gramu manga* 0.0033 *eura. Tada je funkcija cilja:*

$$Z = 0.0066x_1 + 0.0033x_2, \tag{3.3}$$

gdje su:

- x_1 *količina avokada u jednom obroku u gramima,*
- x_2 *količina manga u jednom obroku u gramima.*

U ovom slučaju, želimo minimizirati funkciju cilja.

Primjer 3.2 je formuliran prema podacima s weba

3.2. Ograničenja

U linearnom programiranju, ograničenja postavljaju granice dopuštenom skupu rješenja problema. Ta ograničenja obično predstavljaju linearni skup nejednadžbi ili jednakosti koje moraju biti ispunjene u odnosu na varijable odlučivanja. Ograničenja se općenito mogu definirati na sljedeći način:

Definicija 3.2. *Neka je dan linearni optimizacijski problem u kojem su* $x_1, x_2, \ldots, x_m$ *varijable odlučivanja. Ograničenja mogu biti izražena kao*

$$b_{11}x_1 + b_{12}x_2 + \cdots + b_{1m}x_m \leq d_1, \tag{3.4}$$

$$b_{21}x_1 + b_{22}x_2 + \cdots + b_{2m}x_m \leq d_2, \tag{3.5}$$

$$\vdots$$

$$b_{n1}x_1 + b_{n2}x_2 + \cdots + b_{nm}x_m \leq d_n, \tag{3.6}$$

pri čemu su b_{ij} *koeficijenti ograničenja,* d_i *konstante na desnoj strani svake nejednadžbe, a* n *predstavlja ukupan broj svih ograničenja. Svaka od ovih nejednadžbi može imati znak* $\leq$, $\geq$ *ili* $=$.

U linearnom programiranju, jedan od temeljnih preduvjeta je nenegativnost varijabli odlučivanja, što znači da se uvijek pretpostavlja da su sve varijable x_i veće ili jednake nuli. Ova pretpostavka je bitna jer omogućava interpretaciju rezultata i olakšava praktičnu primjenu modela na stvarne probleme. Nenegativnost varijabli odlučivanja je ključna za osiguravanje smislenih rješenja unutar okvira linearnog programiranja.

Pokažimo na primjeru postavljanje ograničenja u linearnom programu.

Primjer 3.3. *U pitanju su 4 prehrambena proizvoda - krumpir, riža, mrkva i grašak. Gram krumpira sadrži* 0.17 *grama ugljikohidrata, gram riže sadrži* 0.8 *grama ugljikohidrata, gram mrkve* 0.1 *gram te gram grašaka sadrži* 0.14 *grama. Maksimalni unos ugljikohidrata koji nam dopušta dijeta je* 50 *grama. Ograničenja bi tada izgledala ovako:*

$$0.17x_1 + 0.8x_2 + 0.1x_3 + 0.14x_4 \leq 50 \tag{3.7}$$

gdje su:

- x_1 *količina krumpira u jednom obroku u gramima,*
- x_2 *količina riže u jednom obroku u gramima,*
- x_3 *količina mrkve u jednom obroku u gramima,*
- x_4 *količina graška u jednom obrok u gramima.*

Primjer 3.3 je formuliran prema podacima s weba

Primjer 3.4. *Ako imamo ista 4 prehrambena proizvoda kao u primjeru 3.3 a moramo ispuniti minimalni dnevni unos masti od* 75 *g te krumpir sadrži* 0.001 *grama masti u jednom gramu, riža sadrži* 0.003 *grama masti u jednom gramu, mrkva sadrži* 0.002 *grama masti u jednom gramu te grašak sadrži* 0.004 *grama masti u jednom gramu. Ograničenja će sada biti zapisana na sljedeći način:*

$$0.001x_1 + 0.003x_2 + 0.002x_3 + 0.004x_4 \geq 75 \tag{3.8}$$

gdje su opet:

- x_1 *količina krumpira u jednom obroku u gramima,*
- x_2 *količina riže u jednom obroku u gramima,*
- x_3 *količina mrkve u jednom obroku u gramima,*
- x_4 *količina graška u jednom obroku u gramima.*

Primjer 3.4 je formuliran prema podacima s weba

Primjer 3.5. *Svaki dan pripremamo isti doručak koji sadrži 3 namirnice (jaja, kruh, salata). Cilj nam je da doručak uvijek sadrži* 375 *kalorija . Kalorijska vrijednost jaja po gramu je* 1.55 *kcal, kalorijska vrijednost kruha je* 2.5 *kcal te kalorijska vrijednost salate je* 0.15 *kcal. Za ovaj slučaj, ograničenja će izgleda ovako:*

$$1.55x_1 + 2.5x_2 + 0.15x_3 = 375 \tag{3.9}$$

gdje su:

- x_1 *količina jaja u doručku u gramima,*
- x_2 *količina kruha u doručku u gramima,*
- x_3 *količina salate u doručku u gramima.*

Primjer 3.5 je formuliran prema podacima s weba

3.3. Linearni program

U konačnici, optimizacijski problem linearnog programiranja (linearni program) zadan je pomoću do sad navedenih glavnih elemenata: funkcijom cilja, ograničenjima i uvjetima nenegativnosti. Najopćenitije ga možemo zapisati na sljedeći način:

$$Z = a_1x_1 + a_2x_2 + \cdots + a_nx_n \to \min / \max, \tag{3.10}$$
$$b_{11}x_1 + b_{12}x_2 + \cdots + b_{1n}x_n \leq / \geq / = d_1, \tag{3.11}$$
$$b_{21}x_1 + b_{22}x_2 + \cdots + b_{2n}x_n \leq / \geq / = d_2, \tag{3.12}$$
$$\vdots$$
$$b_{m1}x_1 + b_{m2}x_2 + \cdots + b_{mn}x_n \leq / \geq / = d_m, \tag{3.13}$$
$$x_i \geq 0 \quad \text{za sve} \quad i = 1, 2, \ldots, n. \tag{3.14}$$

Kod teorijskih razmatranja ova forma se može zapisati pomoću matričnog zapisa:

$$Z = A \cdot X^T \to \min / \max, \tag{3.15}$$
$$B \cdot X^T \leq / \geq / = D, \tag{3.16}$$
$$X \geq 0, \tag{3.17}$$

gdje je

$$A = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix}, \tag{3.18}$$
$$X = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}, \tag{3.19}$$
$$B = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \ldots & b_{mn} \end{bmatrix}, \tag{3.20}$$

$$D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}. \tag{3.21}$$

Pokažimo sada na primjerima kako se postavlja linearni program.

Primjer 3.6. *Skupina nutricionista radi na planiranju obroka s ciljem osiguravanja uravnotežene prehrane. Žele minimizirati troškove planiranih obroka. Svaki obrok treba sadržavati minimalno* 20 *grama proteina i minimalno* 40 *grama ugljikohidrata. Također, ukupna kalorijska vrijednost obroka treba biti najmanje* 250 *kcal. Imamo dva sastojka koji se koriste u obrocima:*

- *Tuna - košta* 1.06 *eura po gramu, sadrži* 0.35 *grama proteina i* 0 *grama ugljikohidrata po gramu, te ima ukupno* 1.44 *kcal po gramu,*
- *Soja - košta* 0.53 *eura po gramu, sadrži* 0.38 *grama proteina i* 0.32 *grama ugljikohidrata po gramu, te ima ukupno* 1.73 *kcal po gramu.*

Količinu tune u obroku ćemo označavati s x_1 *dok ćemo količinu soje u obroku označavati s* x_2

Ako znamo da je cijena tune po gramu jednaka 1.06 eura a cijena soje po gramu je 0.53 eura, funkcija cilja tada glasi:

$$Z = 1.06x_1 + 0.53x_2. \tag{3.22}$$

Funkciju cilja želimo minimizirati.

Poznato je da gram tune sadrži 0.35 grama proteina, oko 0 grama ugljikohidrata te 1.44 kalorije a gram soje sadrži 0.38 grama proteina, 0.32 grama ugljikohidrata te 1.73 kalorija. Pošto moramo imati minimalno 20 grama proteina po obroku te minimalno 40 grama ugljikohidrata po obroku i uz to moramo paziti da nam obrok sadrži barem 250 kalorija, ograničenja glase:

$$0.35x_1 + 0.38x_2 \geq 20, \tag{3.23}$$

$$0x_1 + 0.32x_2 \geq 40, \tag{3.24}$$

$$1.44x_1 + 1.73x_2 \geq 250. \tag{3.25}$$

Također moramo navesti i uvjet ne-negativnosti zato što x_1 *i* x_2 *predstavljaju neku količinu te ne mogu biti negativni brojevi:*

$$x_1, x_2 \geq 0. \tag{3.26}$$

Nakon što smo sve postavili, linearni program izgleda ovako:

$$Z = 1.06x_1 + 0.53x_2 \to \min, \tag{3.27}$$

$$0.35x_1 + 0.38x_2 \geq 20, \tag{3.28}$$

$$0x_1 + 0.32x_2 \geq 40, \tag{3.29}$$

$$1.44x_1 + 1.73x_2 \geq 250, \tag{3.30}$$

$$x_1, x_2 \geq 0. \tag{3.31}$$

Primjer 3.6 je djelomično izmišljen, a djelomično formuliran prema podacima s weba.

Primjer 3.7. *Treba optimizirati proizvodnju stolova kako bi profit bio maksimalan. Proizvode se 3 vrste stolova:*

- *radni stol cijene* 70 *eura za koji treba* 7 *daski te* 8 *sati rada,*
- *kuhinjski stol cijene* 100 *eura za koji treba* 9 *daski i* 10 *sati rada,*
- *dječji stol cijene* 50 *eura za koji treba* 5 *daski te* 6 *sati rada.*

Na raspolaganju je 60 *daski te je potrebno napraviti sve unutar* 52 *sata*

Broj proizvedenih radnih stolova označavati ćemo s x_1, broj proizvedenih kuhinjskih stolova označavati ćemo s x_2 te ćemo broj proizvedenih dječjih stolova označavati s x_3.

Ako znamo da je cijena radnog stola 70 *eura, cijena kuhinjskog stola* 100 *eura te cijena dječjeg stola iznosi* 50*, funkcija cilja tada glasi:*

$$Z = 70x_1 + 100x_2 + 50x_3. \tag{3.32}$$

Funkciju cilja želimo maksimizirati.

Ako je poznato da nam za radni stol treba 7 daski i 8 sati rada, za kuhinjski stol 9 daski i 10 sati rada te za dječiji stol 5 daski i 6 sati rada a pri tome na raspolaganju imamo 60 daski i 52 sata za rad, ograničenja glase:

$$7x_1 + 9x_2 + 5x_3 \leq 60, \tag{3.33}$$

$$8x_1 + 10x_2 + 6x_3 \leq 52. \tag{3.34}$$

Također moramo navesti i uvjet ne-negativnosti zato što x_1, x_2 i x_3 predstavljaju količinu proizvedenih stolova te ne mogu biti negativni brojevi:

$$x_1, x_2, x_3 \geq 0. \tag{3.35}$$

Za kraj moramo spomenuti da je riječ o cjelobrojnom programiranju gdje se zahtijeva da neke ili sve varijable odlučivanja budu cijeli brojevi pa moramo još zapisati:

$$x_1, x_2, x_3 \in \mathbb{Z}. \tag{3.36}$$

Nakon što smo sve postavili, linearni program izgleda ovako:

$$Z = 70x_1 + 100x_2 + 50x_3 \to \max, \tag{3.37}$$

$$7x_1 + 9x_2 + 5x_3 \leq 60, \tag{3.38}$$

$$8x_1 + 10x_2 + 6x_3 \leq 52, \tag{3.39}$$

$$x_1, x_2, x_3 \geq 0, \tag{3.40}$$

$$x_1, x_2, x_3 \in \mathbb{Z}. \tag{3.41}$$

Primjer 3.7 ima izmišljene podatke

Primjer 3.8. *Cilj tvrtke je minimizirati ukupne proizvodne troškove. Tvrtka proizvodi četiri različita komada namještaja:*

- *Stolice - cijena je 20 eura te su potrebne 2 daske i 1 komad metala te 5 sati rada,*
- *Stolove - cijena je 50 eura te je potrebno 5 daski i 2 komada metala te 10 sati rada,*
- *Ormare - cijena je 100 eura te je potrebno 10 daski i 5 komada metala te 15 sati rada,*
- *Krevete - cijena je 150 eura te je potrebno 15 daski i 3 komada metala te 20 sati rada.*

Tvrtka na raspolaganju ima 300 daski, 100 komada metala te vrijeme za rad koje iznosi 200 sati.

Broj proizvedenih stolica ćemo označavati s x_1, broj proizvedenih stolova s x_2, broj proizvedenih ormara s x_3 te broj proizvedenih kreveta s x_4

Ako znamo cijenu svakog od navedenih komada namješatja (stolica - 20 eura, stol - 50 eura, ormar - 100 eura i krevet - 150 eura), funkcija cilja će imati sljećei oblik:

$$Z = 20x_1 + 50x_2 + 100x_3 + 150x_4. \tag{3.42}$$

Funkciju cilja želimo minimizirati.

Poznavajući informacije koliko nam je drva, metala i vremena potrebno za izradu pojedinog komada namješataja te informacijama da tvrtka na raspolaganju ima 300 daski, 100 komada metala i 200 radnih sati, ograničenja glase:

$$2x_1 + 5x_2 + 10x_3 + 15x_4 \leq 300, \tag{3.43}$$

$$1x_1 + 2x_2 + 5x_3 + 3x_4 \leq 100, \tag{3.44}$$

$$5x_1 + 10x_2 + 15x_3 + 20x_4 \leq 200. \tag{3.45}$$

Također moramo navesti i uvjet ne-negativnosti zato što x_1, x_2, x_3 i x_4 predstavljaju količinu i ne mogu biti negativni brojevi:

$$x_1, x_2, x_3, x_4 \geq 0. \tag{3.46}$$

Kako je kao i u predhodnom primjeru riječ o cjelobrojnom programiranju, za kraj moramo navesti:

$$x_1, x_2, x_3, x_4 \in \mathbb{Z} \tag{3.47}$$

Nakon što smo sve postavili, linearni program izgleda ovako:

$$Z = 20x_1 + 50x_2 + 100x_3 + 150x_4 \to \min, \tag{3.48}$$

$$2x_1 + 5x_2 + 10x_3 + 15x_4 \leq 300, \tag{3.49}$$

$$1x_1 + 2x_2 + 5x_3 + 3x_4 \leq 100, \tag{3.50}$$

$$5x_1 + 10x_2 + 15x_3 + 20x_4 \leq 200, \tag{3.51}$$

$$x_1, x_2, x_3, x_4 \geq 0, \tag{3.52}$$

$$x_1, x_2, x_3, x_4 \in \mathbb{Z}. \tag{3.53}$$

Primjer 3.8 ima izmišljene podatke

4. Teorijski aspekt linearnog programiranja

Cilj ovog poglavlja je dati matematičku pozadinu problemu linearnog programiranja. Matematički aparat na kojem počivaju problemi optimizacije je konveksnost te ćemo najprije definirati taj pojam, točnije pojam konveksnog skupa i konveksne funkcije. Ovo poglavlje je obrađeno pomoću izvora [7] i [8].

Krenimo s definicijom spojnice.

Definicija 4.1. *Neka su zadane dvije točke $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$. Skup*

$$\left\{\mathbf{z} \in \mathbb{R}^2 | \mathbf{z} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2,\ \lambda \in [0,1]\right\} \tag{4.1}$$

zovemo spojnicom točaka x_1 i x_2.

Naime, parametarska jednadžba pravca kroz točke $\mathbf{x}_1$ i $\mathbf{x}_2$ zadana je s

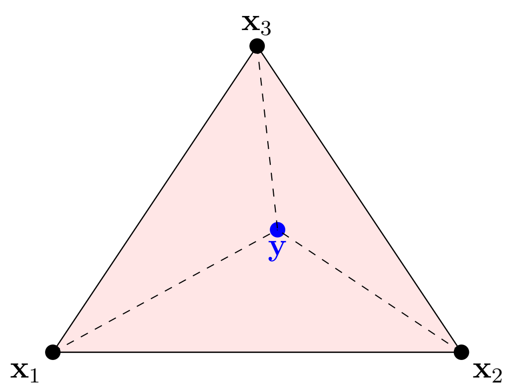
$$\mathbf{z} = \mathbf{x}_1 + \lambda(\mathbf{x}_1 - \mathbf{x}_2),\ \lambda \in \mathbb{R}. \tag{4.2}$$

Uočimo da točke pravca koje se nalaze između točaka $\mathbf{x}_1$ i $\mathbf{x}_2$ dobivamo ako je $\lambda \in [0,1]$, što objašnjava definiciju spojnice.

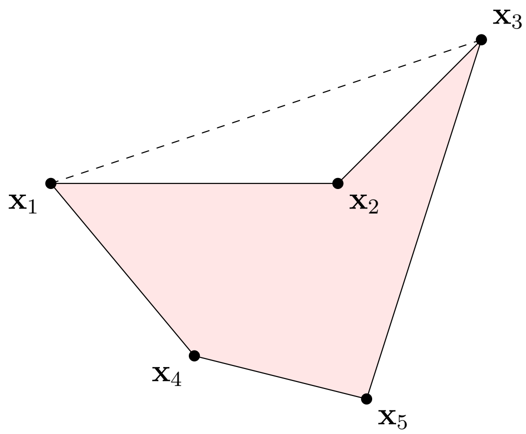
Sada možemo definirati konveksan skup.

Definicija 4.2. *Skup $S \subseteq \mathbb{R}^2$ zovemo* ***konveksnim*** *ako za svake dvije točke $\mathbf{x}_1, \mathbf{x}_2 \in S$ i svaki $\lambda \in [0,1]$, točka $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ također pripada skupu S. Drugim riječima skup je konveksan ako je svaka spojnica svakih dviju točaka skupa sadržana u tom skupu.*

Prikažimo sada grafički konveksan i nekonveksan skup.



Slika 4.1. Prikaz konveksnog skupa. Izvor: Izrada autora



Slika 4.2. Prikaz skupa koji nije konveksan. Izvor: Izrada autora

Na slici 4.1 posebno su označene tzv. vršne točke ili vrhovi konveksnog skupa. To su točke koje će uvijek biti krajevi spojnica, odnosno nikada se neće naći u unutrašnjosti neke spojnice točaka. Pojam spojnice može se generalizirati, pa tako uvodimo pojam konveksne kombinacije.

Definicija 4.3. *Točka* $\mathbf{y} \in \mathbb{R}^2$ *je konveksna kombinacija točaka* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ *ako postoji skup skalara (*$\lambda_1, \lambda_2, \ldots, \lambda_k \geq 0$*) takav da je*

$$\mathbf{y} = \sum_{i=1}^{k} \lambda_i \mathbf{x}_i \tag{4.3}$$

pri čemu vrijedi

$$\sum_{i=1}^{k} \lambda_i = 1. \tag{4.4}$$

Na slici 4.1 su prikazane tri vršne točke ($\mathbf{x}_1$), ($\mathbf{x}_2$), i ($\mathbf{x}_3$), koje definiraju trokut. Točka ($\mathbf{y}$) unutar trokuta prikazuje konveksnu kombinaciju vršnih točaka:

$$\mathbf{y} = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \lambda_3 \mathbf{x}_3 \tag{4.5}$$

gdje su $(\lambda_1, \lambda_2, \lambda_3 \geq 0)\ i\ (\lambda_1 + \lambda_2 + \lambda_3 = 1)$.

Ovaj rezultat ne vrijedi samo za trokut, nego i za bilo koji konveksan skup što vidimo iz idućeg teorema, koji zbog složenosti u ovom radu ne dokazujemo.

Teorem 4.1. *Neka je zadan konveksan skup* $S \subseteq \mathbb{R}^2$*. Svaku točku* $\mathbf{y} \in S$ *možemo izraziti kao konveksnu kombinaciju vršnih točaka skupa* S*.*

Za problematiku linearnog programiranja od presudne važnosti je sljedeći teorem.

Teorem 4.2. *Neka je* A *matrica s* m *redova i* n *stupaca,* $\mathbf{b}$ *vektor s* m *komponenti te* X *vektor s* n *promjenjivih komponenti. Skup svih rješenja linearnog sustava nejednadžbi* $AX \leq \mathbf{b}$ *čini konveksan skup.*

Dokaz. Kako bismo utvrdili konveksnost skupa rješenja sustava $AX \leq \mathbf{b}$ moramo dokazati da za dva proizvoljna rješenja X_1 i X_2 i bilo koji parametar $\lambda \in [0,1]$, konveksna kombinacija $X_\lambda = \lambda X_1 + (1-\lambda)X_2$ zadovoljava i uvjet $AX_\lambda \leq \mathbf{b}$.

Ako su X_1 i X_2 rješenja danog sustava nejednadžbi, vrijedi:

$$AX_1 \leq \mathbf{b} \quad \text{i} \quad AX_2 \leq \mathbf{b}. \tag{4.6}$$

Nadalje je

$$AX_\lambda = A(\lambda X_1 + (1-\lambda)X_2) = \lambda AX_1 + (1-\lambda)AX_2 \leq \lambda\mathbf{b} + (1-\lambda)\mathbf{b} \leq \mathbf{b}, \tag{4.7}$$

pa zaključujemo kako je konveksna kombinacija $X_\lambda = \lambda X_1 + (1-\lambda)X_2$ također rješenje danog sustava nejednadžbi, čime je tvrdnja dokazana. □

Ovaj teorem nam kaže da će ograničenjima linearnog programa biti zadan konveksan skup mogućih rješenja zadanog programa.

Sada možemo uvesti pojam konveksne funkcije.

Definicija 4.4. *Funkcija $f : S \to \mathbb{R}$, gdje je $S \subseteq \mathbb{R}^2$ konveksan skup, naziva se **konveksna** ako za svake dvije točke $\mathbf{x}_1, \mathbf{x}_2 \in S$ i svaki $\lambda \in [0,1]$ vrijedi:*

$$f(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2). \tag{4.8}$$

U ovom se radu posebno bavimo linearnim funkcijama, pa pokažimo da je linearna funkcija konveksna.

Teorem 4.3. *Neka je zadana linearna funkcija $f : S \to \mathbb{R}$ s*

$$f(x,y) = Ax + By + C, \tag{4.9}$$

gdje su A, B i C realne konstante. Funkcija f konveksna je funkcija.

Dokaz. Neka je $\mathbf{x} = (x_1, y_1)$ i $\mathbf{y} = (x_2, y_2)$. Tada je

$$f(\mathbf{x}) = Ax_1 + By_1 + C, \tag{4.10}$$

te

$$f(\mathbf{y}) = Ax_2 + By_2 + C. \tag{4.11}$$

Nadalje je

$$\lambda\mathbf{x} + (1-\lambda)\mathbf{y} = \lambda(x_1, y_1) + (1-\lambda)(x_2, y_2) = (\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2), \tag{4.12}$$

pa vrijedi

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) = A(\lambda x_1 + (1-\lambda)x_2) + B(\lambda y_1 + (1-\lambda)y_2) + C \tag{4.13}$$

$$= \lambda(Ax_1 + By_1 + C) + (1 - \lambda)(Ax_2 + By_2 + C) = \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}). \tag{4.14}$$

Prema tome vrijedi

$$f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) = \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}), \tag{4.15}$$

što znači da je zadana funkcija konveksna. □

Za konveksne funkcije vrijedi sljedeći teorem na kojem je bazirana grafička metoda.

Teorem 4.4. *Neka je $S \subseteq \mathbb{R}^2$ konveksan skup u ravnini, i neka je $f : S \to \mathbb{R}$ konveksna funkcija. Tada f postiže svoje ekstremne vrijednosti u nekoj od vršnih točaka danog skupa S.*

Dokaz. Neka je S konveksan skup u $\mathbb{R}^2$ s vršnim točkama $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k$. Svaka točka $\mathbf{y} \in S$ može se izraziti kao konveksna kombinacija vršnih točaka, pa je

$$\mathbf{y} = \sum_{i=1}^{k} \lambda_i \mathbf{t}_i, \tag{4.16}$$

gdje su koeficijenti $\lambda_i \geq 0$ i $\sum_{i=1}^{k} \lambda_i = 1$.

Jednostavnosti radi promotrimo samo spojnicu točaka $\mathbf{t}_1$ i $\mathbf{t}_2$, tj. točka $\mathbf{y}$ bit će proizvoljna unutarnja točka te spojnice.

Budući je f konveksna funkcija, vrijedi:

$$f(\mathbf{y}) = f\left(\lambda\mathbf{t}_1 + (1 - \lambda)\mathbf{t}_2\right) \leq \lambda f(\mathbf{t}_1) + (1 - \lambda) f(\mathbf{t}_2). \tag{4.17}$$

Ako bi maksimum funkcije f na promatranoj spojnici bio u točki y tada bi vrijedilo

$$f(\mathbf{y}) > f(\mathbf{t}_1) \tag{4.18}$$

i

$$f(\mathbf{y}) > f(\mathbf{t}_2). \tag{4.19}$$

Uvrštavanjem (4.18) i (4.19) u (4.17) slijedi

$$f(\mathbf{y}) < \lambda f(\mathbf{y}) + (1 - \lambda) f(\mathbf{y}) = f(\mathbf{y}), \tag{4.20}$$

što dovodi do kontradikcije. Drugim riječima, maksimalna ili minimalna vrijednost funkcije f ne može biti veća (ili manja) od vrijednosti funkcije f u vršnim točkama, čime je teorem dokazan. □

Ovaj teorem nam govori da za naći rješenje linearnog programa dovoljno je odrediti vrhove konveksnog skupa u kojem se nalazi rješenje, te direktnim uvrštavanjem odrediti vrh u kojem se postiže tražena ekstremna vrijednost.

5. Simpleks metoda

U prethodnom poglavlju pokazali smo da se problem linearnog programiranja rješava na način da najprije odredimo konveksni skup mogućih rješenja problema, zatim odredimo vrhove tog konveksnog skupa i na kraju direktnom provjerom utvrdimo u kojem se od dobivenih vrhova nalazi rješenje.

U ovom poglavlju opisati ćemo algoritam koji prema opisanom postupku traži rješenje linearnog programa, neovisno o broju varijabli odlučivanja, a taj se algoritam zove simpleks algoritam ili simpleks metoda. Ovo poglavlje će biti izrađeno prema izvorima: [6], [9] i [10].

5.1. Grafička metoda rješavanja linearnog programa

Ako problem ima samo dvije varijable, tada je pripadni skup mogućih rješenja konveksni poligon u ravnini i problem ima jasnu i direktnu geometrijsku interpretaciju kako nam pokazuje sljedeći primjer. Korištenje direktne geometrijske interpretacije kod rješavanja linearnog programa zove se grafička metoda.

Primjer 5.1. *Riješimo grafičkom metodom sljedeći linearni program:*

$$Z = 8x_1 + 12x_2 \to \max, \tag{5.1}$$

$$6x_1 + 5x_2 \leq 18, \tag{5.2}$$

$$3x_1 + 4x_2 \leq 12, \tag{5.3}$$

$$x_1, x_2 \geq 0. \tag{5.4}$$

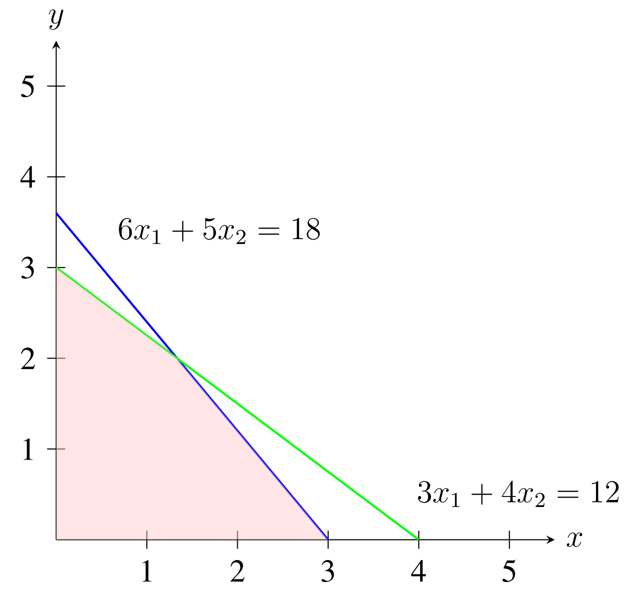
Prvi korak je iscrtavanje konveksnog skupa definiranog ograničenja. Kako bi odredili rubove tog skupa, nejednažbe iz ograničenja zamjenjujemo pripadnim jednadžbama, odnosno pravcima. Zbog znaka $\leq$ u oba ograničenja promatramo područje koje se nalazi ispod tih pravaca u prvom kvadrantu. Da se radi o prvom kvadrantu jasno je iz uvjeta nenegativnosti. Pripadni konveksni skup prikazan je na slici 5.1.

Kako smo već rekli, potencijalna rješenja vrhovi su prikazanog poligona i tu se radi o točkama $T_1(0,0)$, $T_2(0,3)$, $T_3(\frac{4}{3},2)$ i $T_4(3,0)$, kako prikazuje slika 5.2. Do navedenih točaka analitički dolazimo traženjem sjecišta pravaca koji definiraju pojedinu točku.

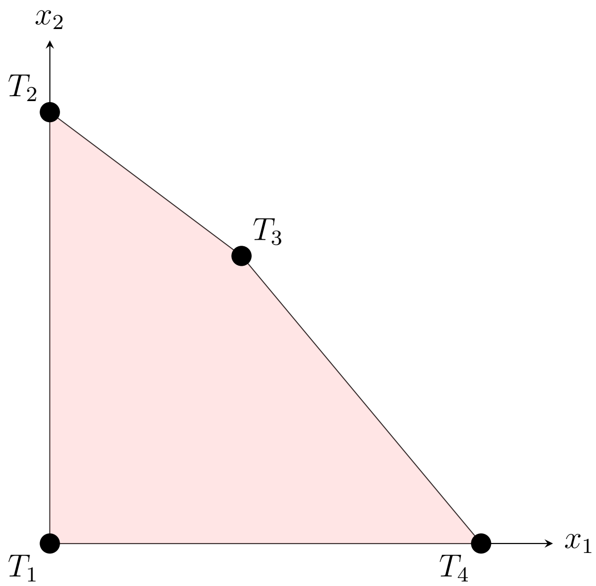
Sada direktnom provjerom određujemo rješenje linearnog programa.

Uvrštavanjem prve točke u funkciju cilja slijedi:

$$Z = 8 \cdot 0 + 12 \cdot 0 = 0. \tag{5.5}$$



Slika 5.1. Konveksni skup koji predstavlja skup mogućih rješenja zadanog linearnog programa. Izvor: izrada autora



Slika 5.2. Potencijalna rješenja zadanog linearnog programa. Izvor: izrada autora

Kada uvrstimo drugu točku dobivamo:

$$Z = 8 \cdot 0 + 12 \cdot 3 = 36. \tag{5.6}$$

Vrijednost funkcije cilja u trećoj točki jednaka je: Kada ubacimo treću točku u funkciju cilja:

$$Z = 8 \cdot \frac{4}{3} + 12 \cdot 2 = \frac{104}{3}. \tag{5.7}$$

I na posljetku, kada uvrstimo četvrtu točku u funkciju cilja slijedi:

$$Z = 8 \cdot 3 + 12 \cdot 0 = 24. \tag{5.8}$$

Sada možemo vidjeti da funkcija cilja postiže maksimalnu vrijednost $Z = 36$ koja se postiže u točki T_2, odnosno za $x_1 = 0$ i $x_2 = 3$.

Rješavanjem ovog primjera jasno je da ovu metodu možemo primijeniti samo na slučajeve u kojima sudjeluju dvije varijable odlučivanja. Naime, veći broj varijabli odlučivanja zahtijeva veći broj prostornih dimenzija što nije moguće adekvatno grafički predočiti. Međutim, 1947. godine George Dantzig generalizirao je metodu na proizvoljan broj varijabli odlučivanja stvorivši tako simpleks metodu koju u ovom radu opisujemo.

5.2. Osnovni pojmovi simpleks metode

U ovom dijelu uvodimo terminologiju ključnu za razumijevanje simpleks metode. Tako ćemo objasniti što su to dodatne ili slack varijable, kada varijablu zovemo baričnom, a kada nebazičnom. Povezati ćemo grafičku metodu sa simpleks metodom, objasniti pojam iteracije i pivotiranja te vezu simpleks metode i linearnih sustava.

Objasnimo najprije naziv simpleks. Već smo pokazali da je skup rješenja linearnog programa konveksan. Geometrijski gledano najmanji konveksni skup u dvije dimenzije je trokut, a ako bismo to generalizirali na tri dimenzije dobili bismo tetraedar. Ovi se geometrijski objekti mogu generalizirati i na veći broj dimenzija i tada ih nazivamo simpleksima. Tako bi točka bila 0-simpleks, dužina 1-simpleks, trokut 2-simpleks, a tetraedar 3-simpleks. Kako se prema tome svaki konveksni skup, neovisno o dimenziji, sastoji od niza simpleksa - metoda koja analizira vrhove konveksnog skupa naziva s simpleks metoda.

Veći broj dimenzija nosi dva osnovna problema. Prvi je nemogućnost grafičke reprezentacije problema, a drugi je računska složenost postupka koja zahtjeva algoritamski pristup i softversku podršku. Stoga je prvi problem koji moramo riješiti analitički zapis vrhova simpleksa koji će biti pogodan za računalnu implementaciju neovisno od broju dimenzija.

Objasnimo taj problem na jednom konkretnom primjeru. Promotrimo konveksni skup prikazan na slici 5.3 definiran ograničenjima:

$$5x_1 + 3x_2 \leq 12, \tag{5.9}$$

$$3x_1 + x_2 \leq 6, \tag{5.10}$$

$$x_1, x_2 \geq 0. \tag{5.11}$$

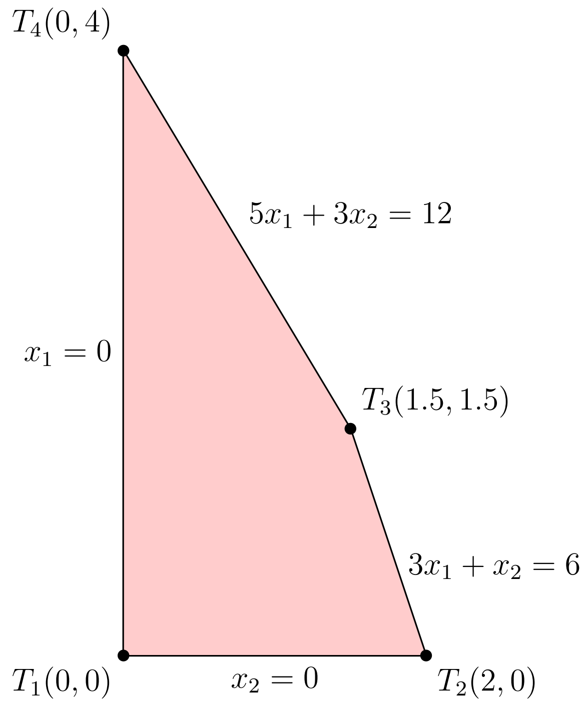
Ovaj skup nejednadžbi možemo transformirati u sustav jednadžbi uvođenjem dodatnih (eng. slack) varijabli na sljedeći način:

$$5x_1 + 3x_2 + s_1 = 12, \tag{5.12}$$

$$3x_1 + x_2 + s_2 = 6, \tag{5.13}$$

$$x_1, x_2, s_1, s_2 \geq 0, \tag{5.14}$$

gdje su s_1 i s_2 dodatne varijable.



Slika 5.3. Konveksni skup zadan ograničenjima. Izvor: izrada autora

Uočimo da ako varijable s_1 i s_2 postavimo na nulu da će rješenje dobivenog sustava biti točka T_3. Ako varijable x_1 i x_2 postavimo na nulu direktno dobivamo točku T_1. Ako su na nulu postavljene varijable x_1 i s_1 dobivamo da je $x_2 = 4$, što odgovara točci T_4, a ako je x_2 i s_2 postavljeno na nulu dobivamo točku T_2. Ovo znači da do vrhova poligona možemo doći rješavanjem dobivenog sustava postavljajući neke varijable na nulu. Drugim riječima, varijable sustava čije vrijednosti nisu nula definiraju vrh.

Sada možemo varijable klasificirati na bazične i nebazične. Bazične varijable su one koje imaju vrijednost različitu od nule, dok nebazične varijable imaju vrijednost nula. Za bazične varijable ćemo reći da čine bazu, pri čemu baza definira jedan vrh konveksnog poligona. Dakle, ako jedna bazična varijabla izađe iz baze, a jedna nebazična uđe u bazu geometrijski gledano prešli smo s vrha na vrh konveksnog poligona.

Sada se postavlja pitanje kako algoritamski realizirati ulazak i izlazak varijabli iz baze. Pogledajmo stoga kako bi izgledale matrice sustava (5.12)-(5.13) kod pojedinih vrhova poligona.

1. $T_1(0,0)$, $B = \{s_1, s_2\}$:

$$\begin{bmatrix} 5 & 3 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{5.15}$$

2. $T_2(2,0)$, $B = \{x_1, s_1\}$:

$$\begin{bmatrix} 5 & 3 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.16}$$

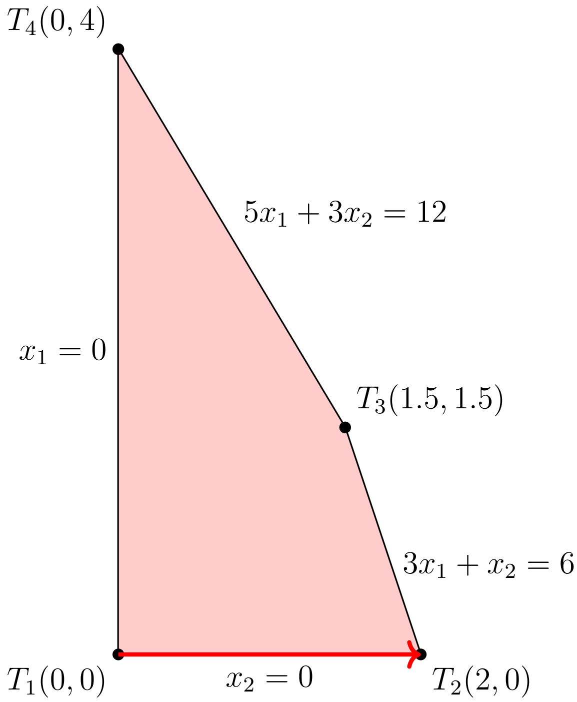
3. $T_3(1.5, 1.5)$, $B = \{x_1, x_2\}$:

$$\begin{bmatrix} 5 & 3 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.17}$$

4. $T_4(0,4)$, $B = \{x_2, s_2\}$:

$$\begin{bmatrix} 5 & 3 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.18}$$

Razmotrimo sada kako bi proveli tranziciju s vrha T_1 na vrh T_2 koja je prikazana na slici 5.4. Ta tranzicija znači da iz baze treba izaći varijabla s_2, a ući varijabla x_1. U matričnom smislu to znači da elementarnim transformacijama na recima iz matrice (5.15) treba doći do matrice (5.16).



Slika 5.4. Prikaz tranzicije s vrha T_1 u vrh T_2. Izvor: izrada autora

Iz svega navedenog jasno je da će simpleks metoda počivati na linearnom sustavu koji se dobije uvođenjem dodatnih varijabli te elementarnim transformacijama na recima kojima će se prelaziti s vrha na vrh konveksnog skupa koji opisuje rješenje.

Prelazak s vrha na vrh zove se iteriranje, a izbor vrha na koji se prelazi temeljit će se na poboljšanja vrijednosti funkcije cilja.

U svakoj iteraciji, simpleks algoritam određuje nebazičnu varijablu koja će ući u bazu (ulazna varijabla) i bazičnu varijablu koja će izaći iz baze (izlazna varijabla). Taj proces se ponavlja sve dok se ne dođe do optimalnog rješenja, odnosno vrha u kojem nije moguće daljnje poboljšanje vrijednosti funkcije cilja. Izbor ulazne i izlazne varijable naziva se pivotiranje.

5.3. Simpleks tablica

U prethodnom dijelu opisali smo simpleks metodu, no za njeno provođenje u pravilu se formira jedna specifična tablica koja se naziva simpleks tablica. Njeno formiranje objasniti ćemo na primjeru.

Neka je zadan sljedeći linearni program:

$$Z = 3x_1 + 5x_2 \to \max, \tag{5.19}$$

$$x_1 \leq 4, \tag{5.20}$$

$$3x_2 \leq 12, \tag{5.21}$$

$$3x_1 + 2x_2 \leq 18, \tag{5.22}$$

$$x_1, x_2 \geq 0. \tag{5.23}$$

Uvedimo dodatne varijable kako bi ograničenja zapisali u tzv. kanonskom obliku. Linearni program sada poprima oblik:

$$Z = 3x_1 + 5x_2 \to \max, \tag{5.24}$$

$$x_1 + s_1 = 4, \tag{5.25}$$

$$3x_2 + s_2 = 12, \tag{5.26}$$

$$3x_1 + 2x_2 + s_3 = 18, \tag{5.27}$$

$$x_1, x_2, s_1, s_2, s_3 \geq 0. \tag{5.28}$$

Linearni program sada unosimo u jednu posebnu tablicu koja se zove simpleks tablica. Za navedeni primjer to vidimo u tablici 5.1

Tablica 5.1. Simpleks tablica linearnog programa.

c_j		3	5	0	0	0	
	BAZA	x_1	x_2	s_1	s_2	s_3	b
0	s_1	1	0	1	0	0	4
0	s_2	0	3	0	1	0	12
0	s_3	3	2	0	0	1	18
	z_j	0	0	0	0	0	0
	$z_j - c_j$	-3	-5	0	0	0	

Objasnimo sada elemente simpleks tablice:

- Prvi red (red c_j) sadrži koeficijente funkcije cilja za svaku varijablu programa. Prva se dva koeficijenta odnosno na zadane varijable, a druga tri koeficijenta na dodatne varijable. Kako se u funkciji cilja ne nalaze dodatne varijable tako su ovdje koeficijenti koji se na njih odnose postavljeni na nulu.
- Drugi red prikazuje sve varijable uključene u linearni program. S x su označene zadane varijable, a sa s dodatne varijable koje smo uveli. Stupac b sadrži vrijednosti baričnih varijabli u iteraciji koju trenutno analiziramo. Kako su u prvoj iteraciji zadane varijable postavljene na nulu u toj je iteraciji taj stupac jednak desnoj strani jednadžbi ograničenja.
- U drugom stupcu označene su varijable koje se trenutno nalaze u bazi, a u prvom stupcu su njihovi koeficijenti iz funkcije cilja.
- U redu z_j nalaze se doprinosi pojedinih varijabli vrijednosti funkcije cilja. Za nebazične varijable ta je vrijednost nula, a za bazične varijable dobije se množenjem vrijednosti iz stupca c_j i vrijednosti iz odgovarajućeg stupca.
- Red $z_j - c_j$ pokazuje za koji iznos bi se promijenila vrijednost funkcije cilja povećavanjem vrijednosti pojedine varijable za 1, odnosno predstavlja svojevrsnu stopu promjene funkcije cilja. Ovaj će nam redak biti od velike važnosti jer ćemo za ulazak u bazu birati onu varijablu koja najviše mijenja funkciju cilja. Nas zanimaju samo negativne vrijednosti ovog retka jer u tom slučaju ulazak varijable može poboljšati vrijednost funkcije cilja. Algoritam završava kada više nije moguće poboljšati funkciju cilja, odnosno kada sve vrijednosti u ovom retku budu nenegativne. Negativna vrijednost u ovom retku znači da je trenutni doprinos varijable manji od koeficijenta te da aktiviranjem koeficijenta (ulaskom varijable u bazu) poboljšavamo rješenje.

5.4. Pivotiranje

U prethodnom dijelu formirali smo simpleks tablicu i time dobili prvu iteraciju simpleks metode, odnosno analizu rješenja za $x_1 = x_2 = 0$. Sljedeću iteraciju dobivamo postupkom pivotiranja, odnosno odabiremo varijablu koja će iz baze izaći i varijablu koja će u bazu ući.

Već smo kod formiranja simpleks tablice objasnili da je kriterij ulaska u bazu redak $z_j - c_j$, odnosno da će u bazu ući onda varijabla kod koje je vrijednost u tom retku najnegativnija. U našem slučaju prema tablici 5.1 je to varijabla x_2, a njen stupac ćemo zvati pivotnim stupcem.

Kako bi odredili varijablu koja izlazi iz baze dodajemo u simpleks tablicu novi stupac koji označavamo s R, a on će predstavljati kvocijente stupca b i elemenata pivotnog stupca. To vidimo u tablici 5.2. Iz baze izlazi varijabla koja se nalazi u retku s najmanjim omjerom, dakle varijabla s_2. Taj ćemo red zvati pivotnim redom, a element koji se nalazi na presjeku pivotnog reda i pivotnog elementa pivotom.

Najmanji omjer označava prvu varijablu koja će postati nula ako se ulazna varijabla nastavi povećavati. Dakle, ta varijabla je ona koja bi trebala napustiti bazu jer se njezina vrijednost iscrpljuje prva (tj. postaje nula prva) kad se vrijednost ulazne varijable povećava.

Tablica 5.2. Simpleks tablica linearnog programa s označenim pivotnim stupcem, pivotnim redom, pivotom i stupcem kvocijenata.

c_j		3	5	0	0	0		
	BAZA	x_1	x_2	s_1	s_2	s_3	b	R
0	s_1	1	0	1	0	0	4	∞
0	s_2	0	**3**	0	1	0	12	4
0	s_3	3	2	0	0	1	18	9
	z_j	0	0	0	0	0	0	
	$z_j - c_j$	-3	-5	0	0	0		

Presjekom odabranog reda i stupca dobivamo broj 3 koji se još naziva i pivot. On je ključni element koji se koristi za transformaciju simpleks tablice iz jedne iteracije u drugu. Kako smo ukrstili red x_4 i stupac x_2 u zapisu sljedeće tablice na mjesto x_4 ćemo taj red zapisivati kao x_2.

Sada možemo formirati iduću iteraciju simpleks tablice. Ulazak varijable x_2 kako je već prije najavljeno provodimo elementarnim operacijama na redovima. Drugi redak dijelimo s pivotom kako bi na njegovu mjestu dobili broj 1 i onda pomoću te jedinice u ostalim redovima tog stupca postižemo nule. Konkretno množenjem drugo reda s -2 i dodavanja zadnjem redu. Dobiveni rezultat prikazan je u tablici 5.3

Tablica 5.3. Prva iteracija simpleks tablice linearnog programa.

c_j		3	5	0	0	0	
	BAZA	x_1	x_2	s_1	s_2	s_3	b
0	s_1	1	0	1	0	0	4
5	x_2	0	1	0	1/3	0	4
0	s_3	3	0	0	-1	1	6
	z_j	0	5	0	5/3	0	20
	$z_j - c_j$	-3	0	0	5/3	0	

Sada krećemo na sljedeću iteraciju. Zadnji redak u tablici 5.3 kaže nam da će u idućoj iteraciji u bazu ući varijabla x_1 te da će pripadni stupac biti pivotni stupac. Ponovno formiramo tablicu sa stupcem kvocijenata pomoću kojih određujemo varijablu koja izlazi iz baze te pivotni red i pivotni element. To je prikazano u tablici 5.4.

Tablica 5.4. Prva iteracija simpleks tablice s označenim pivotom.

c_j		3	5	0	0	0		
	BAZA	x_1	x_2	s_1	s_2	s_3	b	R
0	s_1	1	0	1	0	0	4	4
5	x_2	0	1	0	1/3	0	4	∞
0	s_3	**3**	0	0	-1	1	6	2
	z_j	0	5	0	5/3	0	20	
	$z_j - c_j$	-3	0	0	5/3	0		

Najmanji kvocijent odgovara varijabli s_3 što znači da ona izlazi iz baze, a da na prvom stupcu provodimo opisani postupak elementarnih transformacija s redovima. Dakle, treći red dijelimo s 3, a nakon toga od prvog reda oduzmemo treći red.

Tablica 5.5. Druga iteracija simpleks tablice linearnog programa.

c_j		3	5	0	0	0	
	BAZA	x_1	x_2	s_1	s_2	s_3	b
0	s_1	0	0	1	1/3	-1/3	2
5	x_2	0	1	0	1/3	0	4
3	x_1	1	0	0	-1/3	1/3	2
	z_j	3	5	0	2/3	1	26
	$z_j - c_j$	0	0	0	2/3	1	

U zadnjem redu ove simpleks tablice više nema negativnih elemenata, što znači da je proces završen. Iz tablice čitamo da je optimalna vrijednost funkcije cilja $Z = 26$ te da se ona postiže za $x_1 = 2$ i $x_2 = 4$.

Riješimo sada simpleks metodom jedan praktičan primjer linearnog programa. Napomenimo da je primjer izmišljen.

Primjer 5.2. *Tvrtka proizvodi dvije vrste odjevnih predmeta, ljetne košulje i hlačice. Na raspolaganju ima 200 metara tkanine. Znamo da nam za košulje trebaju 2 metra tkanine, a za hlačice 1 metar. Kupci neće kupiti više od 100 komada odjeće. Cijena ljetne košulje iznosi 60 eura, dok je cijena hlačica 40 eura. Nije potrebno izraditi više od 60 komada košulja. Odredimo koliko komada svakog odjevnog predmeta tvrtka mora proizvesti kako bi zarada bila maksimalna.*

Broj komada košulja označavat ćemo s x_1, a hlačica s x_2. Kako nam za košulje treba 2 metra tkanine, a za hlačice 1 metar tkanine od ukupno 200 metara, prvo ograničenje možemo zapisati na sljedeći način:

$$2x_1 + x_2 \leq 200. \tag{5.29}$$

Ukupan zbroj odjevnih predmeta ne smije biti veći od 100. Dakle, naše drugo ograničenje izgleda ovako:

$$x_1 + x_2 \leq 100. \tag{5.30}$$

Također, ne treba nam više od 60 komada košulja stoga naše treće ograničenje glasi:

$$x_1 \leq 60. \tag{5.31}$$

Kako se radi o broju odjevnih predmeta logično je da se radi o pozitivnim brojevima te moramo postaviti i uvjet ne-negativnosti:

$$x_1, x_2 \geq 0. \tag{5.32}$$

Napomenimo da kod ovog primjera tražimo cjelobrojno rješenje, ali zbog jednostavnosti to ograničenje nećemo nametati.

Kako bi postavili funkciju cilja Z koju želimo maksimizirati, moramo pogledati cijene predmeta. Košulja vrijedi 60 eura, dok hlačice vrijede 40 eura. To možemo zapisati kao:

$$Z = 60x_1 + 40x_2 \to \max. \tag{5.33}$$

Pripadni linerni program sada poprima sljedeći oblik:

$$\begin{align} Z = 60x_1 + 40x_2 \to \max, \tag{5.34}\\ 2x_1 + x_2 \leq 200, \tag{5.35}\\ x_1 + x_2 \leq 100, \tag{5.36}\\ x_1 \leq 60, \tag{5.37}\\ x_1, x_2 \geq 0, \tag{5.38} \end{align}$$

dok njegova kanonska forma glasi

$$\begin{align} Z = 60x_1 + 40x_2 \to \max, \tag{5.39}\\ 2x_1 + x_2 + s_1 = 200, \tag{5.40}\\ x_1 + x_2 + s_2 = 100, \tag{5.41}\\ x_1 + s_3 = 60, \tag{5.42}\\ x_1, x_2, s_1, s_2, s_3 \geq 0. \tag{5.43} \end{align}$$

Sada možemo formirati simpleks tablicu. Ispunjavamo ju varijablama i koeficijentima uz varijable na isti način kao što smo to radili u prošlom primjeru, no ovdje navodimo tablicu 5.6 u kojoj je već određen pivot.

Tablica 5.6. Simpleks tablica linearnog programa s označenim pivotnim stupcem, pivotnim redom, pivotom i stupcem kvocijenata.

c_j		60	40	0	0	0		
	BAZA	x_1	x_2	s_1	s_2	s_3	b	R
0	s_1	2	1	1	0	0	200	100
0	s_2	1	1	0	1	0	100	100
0	s_3	**1**	0	0	0	1	60	60
	z_j	0	0	0	0	0	0	
	$z_j - c_j$	-60	-40	0	0	0		

Iz tablice 5.6 je jasno da će x_1 u sljedećoj iteraciji ući u bazu te da uzimamo stupac x_1 kao stupac za pivotiranje. Pivotni red je treći red, odnosno varijabla s_3 izlazi iz baze. Prva iteracija u kojoj se izvršila naznačen ulaz i izlaz varijabli prikazan je u tablici 5.7, dok je druga iteracija prikazana u tablici 5.8.

Tablica 5.7. Prva iteracija simpleks tablice.

c_j		60	40	0	0	0		
	BAZA	x_1	x_2	s_1	s_2	s_3	b	R
0	s_1	0	1	1	0	-2	80	80
0	s_2	0	**1**	0	1	-1	40	40
60	x_1	1	0	0	0	1	60	∞
	z_j	60	0	0	0	60	3600	
	$z_j - c_j$	0	-40	0	0	60		

Tablica 5.8. Druga iteracija simpleks tablice.

c_j		60	40	0	0	0		
	BAZA	x_1	x_2	s_1	s_2	s_3	b	R
0	s_1	0	0	1	-1	-1	40	
40	x_2	0	1	0	1	-1	40	
60	x_1	1	0	0	0	1	60	
	z_j	60	40	0	40	20	5200	
	$z_j - c_j$	0	0	0	40	20		

Kako u zadnjem redu više nema negativnih vrijednosti nova iteracija nije potrebna te iz ove tablice možemo pročitati rješenje linearnog programa. Maksimalna vrijednost funkcije cilja iznosi $Z = 5200$, a ona se postiže za $x_1 = 60$ i $x_2 = 40$. Dakle, tvrtka mora proizvesti 60 ljetnih košulja i 40 komada kratkih hlačica kako bi imala maksimalan profit koji iznosi 5200 eura.

5.5. Problem minimuma

Nakon što smo pokazali dva primjera kod kojih se maksimizirala funkcije cilja, pokazati ćemo i jedan numerički primjer gdje se traži minimum funkcije cilja.

Primjer 5.3. *Riješimo sljedeći linearni program:*

$$Z = -7x_1 - 2x_2 - x_3 \rightarrow min \tag{5.44}$$

$$5x_1 + 3x_2 \leq 15, \tag{5.45}$$

$$x_1 + x_3 \leq 5, \tag{5.46}$$

$$2x_1 + 6x_2 + 3x_3 \leq 18, \tag{5.47}$$

$$x_1, x_2, x_3 \geq 0. \tag{5.48}$$

Kako je simpleks metoda originalno razvijena za problem maksimizacije, moramo primjeniti određene transformacije kako bi se mogli baviti problemom minizacije.

Prvi korak je transformacija funkciju cilja. To ćemo učiniti tako da njene koeficijente pomnožimo s -1 *pa će funkcija cilja biti jednaka:*

$$Z' = 7x_1 + 2x_2 + x_3, \tag{5.49}$$

gdje je:

$$Z' = -Z, \tag{5.50}$$

čime smo promijenili smjer optimizacije. Drugim riječima, ono što je prije bila minimizacija sada postaje maksimizacija.

Daljnji postupak je jednak kao i u do sada riješenim primjerima.

Najprije ograničenja prebacujemo u kanonski oblik:

$$5x_1 + 3x_2 + s_1 = 15, \tag{5.51}$$

$$x_1 + x_3 + s_2 = 5, \tag{5.52}$$

$$2x_1 + 6x_2 + 3x_3 + s_3 = 18, \tag{5.53}$$

te krećemo u formiranje simpleks tablica za dani problem. U tablici 5.9 dana je osnovna tablica na kojoj provodimo pivotiranje.

Tablica 5.9. Osnovna simpleks tablice s označenim pivotom.

c_j		7	2	1	0	0	0		
	BAZA	x_1	x_2	x_3	s_1	s_2	s_3	b	R
0	s_1	**5**	3	0	1	0	0	15	3
0	s_2	1	0	1	0	1	0	5	5
0	s_3	2	6	3	0	0	1	18	9
	z_j	0	0	0	0	0	0	0	
	$z_j - c_j$	-7	-2	-1	0	0	0		

U idućoj iteraciji iz baze nam izlazi varijabla $s1$*, a ulazi varijabla* x_1*, što je prikazano u tablici 5.10.*

Tablica 5.10. Prva iteracija simpleks tablice s označenim novim pivotom.

c_j		7	2	1	0	0	0		
	BAZA	x_1	x_2	x_3	s_1	s_2	s_3	b	R
7	x_1	1	3/5	0	1/5	0	0	3	∞
0	s_2	0	-3/5	**1**	-1/5	1	0	2	2
0	s_3	0	24/5	3	-2/5	0	1	12	4
	z_j	7	21/5	0	7/5	0	0	21	
	$z_j - c_j$	0	11/5	-1	7/5	0	0		

U zadnjoj iteraciji u bazu stavljamo varijablu x_3, pri čemu iz baze izlazi varijabla s_2. Ta je iteracija prikazana u tablici 5.11.

Tablica 5.11. Druga iteracija simpleks tablice.

c_j		7	2	1	0	0	0	
	BAZA	x_1	x_2	x_3	s_1	s_2	s_3	b
7	x_1	1	3/5	0	1/5	0	0	3
1	x_3	0	-3/5	1	-1/5	1	0	2
0	s_3	0	33/5	0	1/5	-3	0	6
	z_j	7	18/5	1	6/5	1	0	23
	$z_j - c_j$	0	8/5	0	6/5	1	0	

Možemo primjetiti da u zadnjem redu tablice 5.11 nema više negativnih koeficijenata što znači kako smo došli do posljednje iteracije.

Funkcija cilja Z' jednaka je 23. To nam je maksimum transformirane funkcije cilja, a kako bismo odredili minimum moramo promijeniti predznak iznosa dobivene funkcije cilja pa je $Z = -23$.

Zaključujemo da minimum naše početne funkcije cilja uz dana ograničenja iznosi -23 za $x_1 = 3$, $x_2 = 0$ i $x_3 = 2$.

6. Charnesova M metoda

M-metoda je oblik simpleks metode koji služi za rješavanje problema linearnog programiranja koji uključuju složena ograničenja ili probleme koji se ne mogu direktno riješiti standardnom simpleks metodom. Uveo ju je američki matematičar Abraham Charnes, po kojemu je metoda i dobila naziv Charnesova M metoda. Ovo poglavlje je napravljeno prema izvoru: [11]

Služi ponajprije za rješavanje problema znakova jednakosti ili nejednakosti kod postavljenih ograničenja. Uobičajeno je da se u ograničenjima nalazi znak $\leq$, no pomoću M-metode i dodavanjem umjetnih varijabli vrlo jednostavno možemo riješiti i slučajeve kada imamo znak jednakosti $=$ ili znak $\geq$, primjenjujući postupak opisan u nastavku.

Prvo dodajemo umjetne varijable svakoj jednadžbi tipa "veće ili jednako" ($\geq$) ili jednakosti ($=$) kako bi se omogućilo početno osnovno rješenje te kako bi se mogao koristiti poznati simpleks algoritam.

Sljedeći korak je dodavanje konstante M pomnožene s umjetnim varijablama u funkciju cilja kako bi se osiguralo da umjetne varijable ne ostanu u konačnom rješenju. Kada se traži maksimum, konstanta M dolazi s negativnim predznakom, dok kod minimizacije dolazi s pozitivnim predznakom, kako bi se spriječilo beskonačno povećanje vrijednosti funkcije cilja.

Ostatak postupka je analogan primjeni simpleks algoritma, s ciljem eliminiranja umjetnih varijabli.

Pokazati ćemo sada to na jednom jednostavnom numeričkom primjeru koji glasi:

$$Z = 30x_1 + 40x_2 \to \min \tag{6.1}$$

$$2x_1 + 3x_2 \geq 8, \tag{6.2}$$

$$4x_1 + 8x_2 \geq 20, \tag{6.3}$$

$$x_1, x_2 \geq 0 \tag{6.4}$$

Ograničenja ćemo zapisati u kanonskom obliku uvodeći dodatne varijable s negativnim predzankom:

$$2x_1 + 3x_2 - s_1 = 8, \tag{6.5}$$

$$4x_1 + 8x_2 - s_2 = 20, \tag{6.6}$$

$$x_1, x_2, s_1, s_2 \geq 0. \tag{6.7}$$

Zbog dodatnih varijabli s negativnim predznakom, suočavamo se s problemom da ta rješenja ne moraju biti dopustiva (ne zadovoljavaju sva ograničenja). Kako bismo riješili ovaj problem, uvodimo umjetne varijable kako bismo dobili početno dopustivo rješenje. Umjetne varijable će nam pomoći da pronađemo rješenje koje zadovoljava sva ograničenja. Umjetne varijable imaju vrlo visok koeficijent M u funkciji cilja kako bi se osiguralo da nestanu iz konačnog rješenja.

Naš problem sada ima sljedeći oblik:

$$Z = 30x_1 + 40x_2 + Mv_1 + Mv_2 \rightarrow \min \tag{6.8}$$

$$2x_1 + 3x_2 - s_1 + v_1 = 8 \tag{6.9}$$

$$4x_1 + 8x_2 - s_2 + v_2 = 20 \tag{6.10}$$

$$x_1, x_2, s_1, s_2, v_1, v_2 \geq 0. \tag{6.11}$$

Sada postavljamo prvu simpleks tablicu na jednak način kao i do sada, s time da se u bazi na početku nalaze umjetne varijable.

Tablica 6.1. Početna simpleks tablica.

c_j		30	40	0	0	M	M		
	BAZA	x_1	x_2	s_1	s_2	v_1	v_2	b	R
M	v_1	2	3	-1	0	1	0	8	8/3
M	v_2	4	**8**	0	-1	0	1	20	5/2
	z_j	6M	11M	-M	-M	M	M	28M	
	$z_j - c_j$	6M - 30	11M - 40	-M	-M	0	0		

Kriteriji za ulazak i izlazak iz baze su jednaki, ali ovdje zbog toga što se radi o problemu minimizacije gledamo najpozitivniji element u zadnjem redu, a to je $11M - 40$. Što se retka tiče, tražimo manji omjer koji je tu $5/2$. Dakle iz baze izlazi v_2, dok u bazu ulazi x_2 Pivot nam je 8 pa moramo cijeli red podijeliti s 8 kako bi za pivot dobili 1. Nakon toga od prvog reda oduzimamo trostruki drugi red.

Tablica 6.2. Prva iteracija simpleks tablice.

c_j		30	40	0	0	M	M		
	BAZA	x_1	x_2	s_1	s_2	v_1	v_2	b	R
M	v_1	**1/2**	0	-1	3/8	1	-3/8	0.5	1
40	x_2	1/2	1	0	-1/8	0	1/8	5/2	5
	z_j	1/2M+20	40	-M	3/8M-5	M	-3/8M+5	1/2M + 100	
	$z_j - c_j$	1/2M - 10	0	-M	3/8M-5	0	-11/8M+5		

Kako i dalje imamo pozitivnih elemenata u zadnjem retku postupak ponavljamo. Sada će u bazu ući x_1, a iz baze izaći v_1.

Tablica 6.3. Druga iteracija simpleks tablice.

c_j		30	40	0	0	M	M		
	BAZA	x_1	x_2	s_1	s_2	v_1	v_2	b	R
30	x_1	1	0	-2	**3/4**	2	-3/4	1	4/3
40	x_2	0	1	1	-1/2	-1	1/2	2	-4
	z_j	30	40	-20	5/2	20	-5/2	110	
	$z_j - c_j$	0	0	-20	5/2	20-M	-5/2-M		

U zadnjem retku ostala je još jedna pozitivna vrijednost pa radimo još jednu iteraciju, u kojoj u bazu ulazi varijabla s_2, a iz baze izlazi varijabla x_1 budući je pripadni omjer jedini pozitivni omjer. Ta je iteracija prikazana u idućoj tablici.

Tablica 6.4. Treća iteracija simpleks tablice.

c_j		30	40	0	0	M	M		
	BAZA	x_1	x_2	s_1	s_2	v_1	v_2	b	
0	s_2	4/3	0	-8/3	1	8/3	-1	4/3	
40	x_2	2/3	1	-1/3	0	1/3	0	8/3	
	z_j	80/3	40	-40/3	0	20	40/3	320/3	
	$z_j - c_j$	-10/3	0	-40/3	0	40/3-M	-M		

Kako u zadnjem retku više nema pozitivnih elemenata, ovo je zadnja iteracija iz koje možemo pročitati i rješenje linearnog programa. Minimum će iznositi $320/3$ i postići će se za $x_1 = 0$ i za $x_2 = 8/3$.

7. Analiza osjetljivosti

U ovom poglavlju ćemo analizirati osjetljivost optimalnog rješenja linearnog programskog problema na promjenu koeficijenata u funkciji cilja i ograničenjima. Ovo poglavlje je napravljeno prema izvorima: [12] i [13]

Razmotrit ćemo sljedeći linearni problem iz primjera 5.2:

$$Z = 60x_1 + 40x_2 \to max \tag{7.1}$$

$$2x_1 + x_2 \leq 200, \tag{7.2}$$

$$x_1 + x_2 \leq 100, \tag{7.3}$$

$$x_1 \leq 60, \tag{7.4}$$

$$x_1, x_2 \geq 0, \tag{7.5}$$

čija je kanonska forma:

$$Z = 60x_1 + 40x_2 \to \max, \tag{7.6}$$

$$2x_1 + x_2 + s_1 = 200, \tag{7.7}$$

$$x_1 + x_2 + s_2 = 100, \tag{7.8}$$

$$x_1 + s_3 = 60, \tag{7.9}$$

$$x_1, x_2, s_1, s_2, s_3 \geq 0. \tag{7.10}$$

Analiza osjetljivosti ispituje kako promjene u koeficijentima ciljne funkcije i u desnim stranama ograničenja utječu na optimalno rješenje.

Pretpostavimo da se koeficijent 60 uz x_1 mijenja za Δ a_1 i koeficijent 40 uz x_2 mijenja za Δ a_2. Nova ciljna funkcija postaje:

$$Z = (60 + \Delta a_1)x_1 + (40 + \Delta a_2)x_2 \tag{7.11}$$

Optimalno rješenje ostaje optimalno sve dok promjene ne naruše optimalnost simpleks tablice. Naše rješenje ostaje optimalno ako:

$$\Delta a_1 \leq \textit{granica}, \tag{7.12}$$

$$\Delta a_2 \leq \text{granica} \tag{7.13}$$

Pretpostavimo da se desna strana prvog ograničenja mijenja za Δ b_1, drugog ograničenja za $Delta$ b_2, i trećeg ograničenja za Δ b_3. Nova ograničenja postaju:

$$2x_1 + x_2 + s_1 = 200 + \Delta b_1, \tag{7.14}$$

$$x_1 + x_2 + s_2 = 100 + \Delta b_2, \tag{7.15}$$

$$x_1 + s_3 = 60 + \Delta b_3. \tag{7.16}$$

Optimalno rješenje ostaje optimalno sve dok promjene ne naruše dopuštenost i optimalnost simpleks tablice. Naše rješenje ostaje optimalno ako:

$$\Delta b_1 \leq \text{granica}, \tag{7.17}$$

$$\Delta b_2 \leq \text{granica}, \tag{7.18}$$

$$\Delta b_3 \leq \text{granica} \tag{7.19}$$

Analiza osjetljivosti omogućuje nam da razumijemo koliko se optimalno rješenje može mijenjati kada se promijene koeficijenti ciljne funkcije ili desne strane ograničenja. Ovo je vrlo korisno kod donošenja odluka u stvarnim situacijama gdje su podaci podložni promjenama.

U našem konkretnom slučaju, može se pokazati da će povećanje ograničenja na ukupni broj dostupnih resursa imati različite utjecaje na maksimalnu vrijednost funkcije cilja. Promjena u ograničenju prvog resursa neće imati utjecaja na ishod. S druge strane, povećanje ukupne dostupnosti drugog ograničenja smanjilo bi maksimalnu vrijednost funkcije cilja za 40 jedinica po jedinici povećanja. Isto tako, povećanje ograničenja na treći resurs bi dovelo do smanjenja vrijednosti funkcije cilja za 20 jedinica po svakoj dodatnoj jedinici resursa.

8. Primjena simpleks metode na problem iz inženjerstva

U ovom ćemo poglavlju prikazati kako se simpleks metoda primjenjuje na jednom realnom problemu iz inženjerstva. Ipak, iako se radi o realnom problemu podaci primjera su djelomično izmišljeni, a djelomično formulirani prema podacima s weba.

Primjer 8.1. *Novonastala tvrtka sudjeluje u projektu koji potiče prelazak na obnovljive izvore energije. Cilj projekta je smanjiti ovisnosti o fosilnim gorivima te povećati energetsku sigurnost i održivost.*

Tvrtka se bavi proizvodnjom solarnih panela, a za to koristi dvije vrste silicija: monokristalni silicij i polikristalni silicij. Polikristalni silicij nabavljaju po cijeni od 20 *eura po kg, dok monokristalni silicij nabavljaju po cijeni od* 80 *eura po kg. Njihovi kapaciteti dozvoljavaju skladištenje najviše* 80 *kg polikristalnog silicija i* 100 *kg monokristalnog silicija, dok je ukupni kapacitet skladišta za silicij maksimalno* 160 *kg. Trenutni proračun tvrtke za nabavu silicija je* 6400 *eura, a cilj je nabaviti što više silicija.*

Količinu polikristalnog silicija ćemo označavati s x_1*, dok ćemo količinu monokristalnog silicija označavati s* x_2*.*

Navedeno je kako tvrtka ima mogućnost skladištenja 160 *kilograma silicija stoga nam prvo ograničenje izgleda ovako:*

$$x_1 + x_2 \leq 160. \tag{8.1}$$

Kako je propisana maksimalna količina polikristalnog silicija, naše drugo ograničenje je ovog oblika:

$$x_1 \leq 80, \tag{8.2}$$

a zbog istog ograničenja za monokromatski silicij dobivamo ograničenje oblika

$$x_2 \leq 100. \tag{8.3}$$

Nakon što smo postavili ograničenja vezano za skladišne kapacitete, moramo formirati i ograničenje koje se odnosi na proračun tvrtke. Iz danih podataka slijedi:

$$20x_1 + 80x_2 \leq 6400. \tag{8.4}$$

Kako nam x_1 *i* x_2 *predstavljaju količinu, moramo postaviti i uvjet ne-negativnosti:*

$$x_1, x_2 \geq 0. \tag{8.5}$$

Osim ograničenja, ključno je definirati i funkciju cilja. Funkcija cilja se odnosi na maksimizaciju nabavljenog silicija (polikristalnog i monokristalnog) pa ima sljedeći, vrlo jednostavan *oblik:*

$$Z = x_1 + x_2 \to \max . \tag{8.6}$$

Sada ćemo formirati kanonsku formu linearnog programa, tj. u svako ograničenje dodat ćemo dodatnu varijablu (s_1, s_2, s_3 i s_4). Linearni program će sada izgledati ovako:

$$Z = x_1 + x_2 \to \max \tag{8.7}$$

$$x_1 + x_2 + s_1 = 160, \tag{8.8}$$

$$x_1 + s_2 = 80, \tag{8.9}$$

$$x_2 + s_3 = 100, \tag{8.10}$$

$$20x_1 + 70x_2 + s_4 = 6400 \tag{8.11}$$

$$x_1, x_2, s_1, s_2, s_3, s_4 \geq 0. \tag{8.12}$$

Postavivši problem, možemo krenuti na sljedeći korak a to je izrada početne simpleks tablice prema pravila objašnjenima u prethodnim poglavljima. Ispunjavamo ju varijablama i koeficijentima uz varijable sukladno načinu na koji smo ispunjavali tablice u prethodnim primjerima, a kako je prikazano u tablici 8.1.

Tablica 8.1. Prva iteracija simpleks tablice s označenim redom pivotiranja i stupcem pivotiranja

c_j		1	1	0	0	0	0		
	BAZA	x_1	x_2	x_3	x_4	x_5	x_6	b	R
0	s_1	1	1	1	0	0	0	160	160
0	s_2	**1**	0	0	1	0	0	80	80
0	s_3	0	1	0	0	1	0	100	∞
0	s_4	20	80	0	0	0	1	6400	320
	z_j	0	0	0	0	0	0	0	
	$z_j - c_j$	-1	-1	0	0	0	0		

Jednako kao i do sada, moramo se osvrnuti na posljednji red tablice kako bismo pronašli gdje je najmanji član i pomoću njega odredili koja varijabla ulazi u bazu. Uočavamo da u ovom slučaju imamo 2 potpuno jednaka negativna člana koji iznose -1. To nam ukazuje da je došlo do pojave degeneracije.

Degeneracija nam ukazuje na to da postoji više od jednog kandidata za pivot stupac, što znači da postoji više jednako "dobrih" načina za poboljšanje trenutnog rješenja.

Problem koji se javlja jest da bez jasnog pravila za odabir stupca pivotiranja, simpleks algoritam može upasti u beskonačnu petlju. Iz tog razloga ćemo se poslužiti Blandovim pravilom

Blandovo pravilo nam govori koju varijablu treba izabrati za ulazak u bazu - između svih varijabli s negativnim koeficijentom u zadnjem redu, odabire se ona s najmanjim indeksom (rednim brojem).

Zaključujemo da onda varijabla x_1 ulazi u bazu te da taj stupac promatramo.

Kako bismo odredili varijablu koja izlazi iz baze, primjenjujemo identičan postupak kao i u prethodnim primjerima. Dijelimo svaki član stupca b s pripadnim članom našeg odabranog stupca te promatramo gdje se pojavio najmanji kvocijent. Veoma lako možemo uočiti da je minimalni kvocijent koji možemo dobiti broj 80 *u četvrtom redu tablice. To nam govori da taj red promatramo te da nam varijabla s_2 izlazi iz baze.*

U sjecištu odabranog reda i odabranog stupca nam se nalazi pivot. On je 1 *pa nije potrebno dodatno dijeliti čitavi red kako bi ga namjestili na tu vrijednost.*

Tako dolazimo do prve iteracije simpleks algoritma koja je prikazana u tablici 8.2.

Tablica 8.2. Druga iteracija simpleks tablice s označenim pivotom

c_j		1	1	0	0	0	0		
	BAZA	x_1	x_2	x_3	x_4	x_5	x_6	b	R
0	s_1	0	1	1	-1	0	0	80	80
1	x_1	1	0	0	1	0	0	80	∞
0	s_3	0	1	0	0	1	0	100	100
0	s_4	0	**80**	0	-20	0	1	4800	60
	z_j	1	0	0	1	0	0	80	
	$z_j - c_j$	0	-1	0	1	0	0		

Ako pogledamo zadnji red tablice i dalje možemo uočiti kako se nismo riješili svih negativnih vrijednosti, pa ulazimo u iduću iteraciju koja je prikazana u tablici 8.3. U toj iteraciji u bazu ulazi varijabla x_2, dok iz nje izlazi varijabla s_4.

Tablica 8.3. Zadnja iteracija simpleks tablice

c_j		1	1	0	0	0	0	
	BAZA	x_1	x_2	x_3	x_4	x_5	x_6	b
0	s_1	0	0	1	-3/4	0	-1/80	20
1	x_1	1	0	0	1	0	0	80
0	s_3	0	0	0	1/4	1	-1/80	40
1	x_2	0	1	0	-1/4	0	1/80	60
	z_j	1	1	0	3/4	0	1/80	140
	$z_j - c_j$	0	0	0	3/4	0	1/80	

Možemo uočiti da više ne postoje negativne vrijednosti u redu $z_j - c_j$ stoga zaključujemo da smo došli do kraja algoritma, tj. da nove iteracije nisu potrebne.

Iz tablice čitamo da se maksimum od 140 *postiže za $x_1 = 80$ i za $x_2 = 60$. Drugim riječima, uz dani proračun tvrtke, maksimalna nabava silicija od* 140 *kilograma postići će se ako nabavimo* 80 *kilograma polikristalnog silicija i* 60 *kilograma monokristalnog silicija.*

8.1. Excel Solver

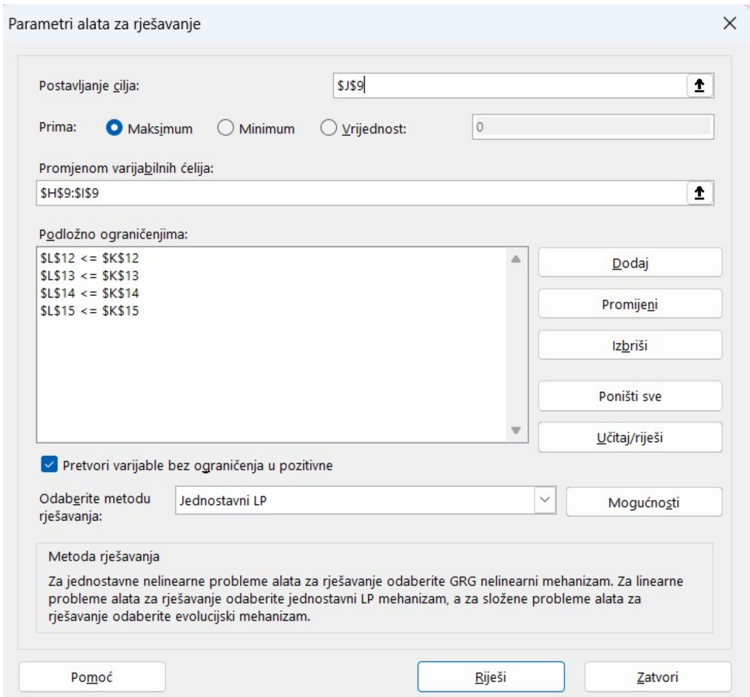
Za provjeru točnosti našeg rješenja, poslužiti ćemo se Excel Solverom. Excel Solver je alat u Microsoft Excelu koji korisnicima omogućava rješavanje složenih optimizacijskih problema. Koristi se za pronalaženje optimalnog rješenja za određenu formulu smještenu u jednoj ćeliji, uzimajući u obzir postavljena ograničenja u drugim ćelijama. Pomoću Solvera, korisnici mogu maksimizirati ili minimizirati vrijednost ciljne ćelije, prilagođavajući vrijednosti varijabli unutar definiranih ograničenja.

Prvo ćemo odrediti ćelije koje će nam odgovarati varijablama što su u našem slučaju x_1 i x_2. Drugi korak nam je definiranu funkciju cilja upisati u jednu od ćelija

Nakon toga ćemo zapisati sva ograničenja. Definirati ćemo 6 stupaca. U prvi ćemo samo navesti redni broj ograničenja dok ćemo u zadnjem staviti opis ograničenja radi bolje preglednosti. U stupac "LHS" ćemo upisati koeficijente naših varijabli u ograničenju. U stupac "RHS" ćemo upisati granice naših ograničenja. Sredinu ćemo ispuniti odgovarajućim znakovima nejednakosti. U stupac "Rješenja" ćemo upisati formulu ograničenja po kojoj se definira kako se množe i zbrajaju ćelije.

Postavivši problem, preostaje nam još sve ubaciti u alat za rješavanje. Moramo unjeti ćeliju koja odgovara funkciji cilja. Zatim moramo izabrati što želimo s funkcijom cilja (u našem slučaju ju želimo maksimizirati). Potom moramo unjeti ćelije koje smo odredili da su nam varijable i za kraj moramo ubaciti sva ograničenja

Kad smo sve navedeno ispunili, preostaje nam samo pritisnuti tipku "Riješi" nakon čega nam Solver izbacuje traženo rješenje.



Slika 8.1. Unošenje podataka u alat za rješavanje. Izvor: Izrada autora

Na slici 8.1 prikazano je sučelje za unos parametara u Solver. Slika 8.2 prikazuje formiran problem u Excelu, dok slika 8.3 prikazuje konačno rješenje.

x1	x2	
Monokristalni	Polikristalni	Vrsta silicija

x1	x2	Z	
0	0	0	Konačno rješenje

Ograničenja	LHS			RHS	Rješenja	Opis ograničenja
1.	1	1	<=	160	0	Kapaciteti skladišta
2.	1		<=	80	0	Max količina monokristala
3.		1	<=	100	0	Max količina polikristala
4.	20	80	<=	6400	=H9*H15+I9*I15	potrošnje u €

Slika 8.2. Zadani problem nakon postavljanja u Solver i prikaz kako se popunjavaju ćelije rješenja. Izvor: Izrada autora

x1	x2	
Monokristalni	Polikristalni	Vrsta silicija

x1	x2	Z	
80	60	140	Konačno rješenje

Ograničenja	LHS			RHS	Rješenja	Opis ograničenja
1.	1	1	<=	160	140	Kapaciteti skladišta
2.	1		<=	80	80	Max količina monokristala
3.		1	<=	100	60	Max količina polikristala
4.	20	80	<=	6400	6400	Cijena max potrošnje u €

Slika 8.3. Rješenje linearnog problema. Izvor: Izrada autora

Možemo primjetiti da se rješenja Solvera podudaraju s onima koja smo dobili preko simpleks tablica.

9. Zaključak

Ovaj rad pruža pregled simpleks metode, polazeći od samih početaka optimizacije kao matematičke discipline. Rad ocrtava povijesni put simpleks algoritma, odajući priznanje ličnostima koje su ga oblikovale i istovremeno ističući njegovu važnost za evoluciju matematičkog modeliranja i optimizacijskih tehnika.

Rad je detaljno objasnio matematičku osnovu linearnog programiranja, definirajući ključne komponente poput funkcije cilja i ograničenja. Posebna pažnja posvećena je konceptu konveksnosti, s detaljnim objašnjenjem definicije konveksnih skupova i funkcija. Dokazano je da su linearne funkcije konveksne, a analizirana je i važnost ekstrema konveksnih funkcija u kontekstu linearne optimizacije.

Simpleks metoda predstavljena je kroz nekoliko koraka, počevši od intuitivnog grafičkog pristupa koji ilustrira osnovne principe. Potom su uvedeni ključni pojmovi poput jednostavnih tablica i pivotiranja, s detaljnim opisom koraka algoritma. Rad je također obradio specifične slučajeve, poput problema minimuma i primjene Charnesove M metode za rješavanje problema s početnom nedozvoljenom točkom.

Objašnjena je i analiza osjetljivosti, ključni aspekt primjene linearne optimizacije u praksi, s naglaskom na važnost razumijevanja utjecaja promjena parametara na optimalno rješenje.

Praktična primjena simpleks metode ilustrirana je kroz konkretan inženjerski problem, demonstrirajući kako se ova metoda može koristiti za optimizaciju resursa i donošenje boljih odluka u realnim situacijama.

Iako je simpleks metoda i dalje nezamjenjiv alat u području linearne optimizacije, važno je spomenuti da postoje i druge metode, poput metode unutarnjih točaka, koje mogu biti učinkovitije za rješavanje problema velikih dimenzija.

Zaključno, ovaj rad pruža osnovu za razumijevanje i primjenu simpleks metode u rješavanju problema linearne optimizacije, pri čemu smo pokazali da metoda vrlo efikasno dolazi do rješenja različitih problema iz poslovanja i inženjerske prakse.

Bibliografija

[1] Souza, E.:"Stanford News Service", s Interneta, https://news.stanford.edu/stories/2005/05/george-b-dantzig-operations-research-professor-dies-90, 16. travnja 2024.

[2] s Interneta, https://en.wikipedia.org/wiki/GeorgeDantzig, 16. travnja 2024.

[3] s Interneta, https://en.wikipedia.org/wiki/Leonid_Khachiyan, 20. travnja 2024.

[4] Martha, S.:"Narendra Karmarkar", s Interneta, https://www.scrolldroll.com/famous-indian-mathematicians/narendra-karmarkar-famous-mathematicians/, 21. travnja 2024.

[5] "Narendra Karmarkar", s Interneta, https://www.javatpoint.com/narendra-karmarkar, 21. travnja 2024.

[6] Kelava, L.:"Modeli linearnog programiranja te njihova primjena u ekonomiji", s Interneta, https://repozitorij.unios.hr/islandora/object/efos%3A4708/datastream/PDF/view, 14. svibnja 2024.

[7] Vandenberghe, L.:"Convex Optimization", s Interneta, https://web.stanford.edu/~boyd /cvxbook/bv_cvxbook.pdf, 10. lipnja 2024.

[8] s Interneta, https://ocw.mit.edu/courses/6-079-introduction-to-convex-optimization-fall-2009/ 10. lipnja 2024.

[9] Mašek, M."Posebna svojstva simpleks algoritma", s Interneta, https://repozitorij.foi.unizg.hr/islandora/object/foi%3A3965/datastream/PDF/view, 21. srpnja 2024.

[10] Lukačević, A."Linearno programiranje", s Interneta, https://repozitorij.pmf.unizg.hr /islandora/ object/pmf%3A8390/datastream/PDF/view, 21 srpnja 2024.

[11] Lončarević, R."OPERACIJSKA ISTRAŽIVANJA U PROMETU", Veleučilište Marko Marulić u Kninu, Gospić, 2021.

[12] Završki, I."Metode optimalizacije u građevinarstvu", s Interneta, https://www.grad.unizg.hr/_download/repository/MO_predavanje_8_-_ Interpretacija_simpleks_tabele_-_analiza_osjetljivosti.pdf, 27. kolovoza 2024.

[13] Ivančić, F."Problem optimizacije proizvodnje papirnate i plastične ambalaže", s Interneta, https://zir.nsk.hr/islandora/object/foi%3A7745/datastream/PDF/view, 28. kolovoza 2024

Sažetak i ključne riječi

Ovaj rad predstavlja analizu simpleks metode, temeljnog algoritma u domeni linearng programiranja. U uvodnom dijelu rada, istražuje se povijesni razvoj simpleks metode, naglašavajući njen značaj i doprinos evoluciji ove matematičke discipline. Slijedi detaljno objašnjenje matematičkog okvira problema linearne optimizacije, uključujući definicije funkcija cilja, ograničenja i strukture samog linearnog programa.

Poseban fokus stavljen je na koncept konveksnosti i njegovu ključnu ulogu u primjeni simpleks metode. Definiraju se konveksni skupovi i konveksne funkcije, te se matematički dokazuje konveksnost linearnih funkcija.

Rad postupno otkriva mehanizme simpleks metode, krećući se od grafičke metode, preko ključnih elemenata poput simpleks tablice i tehnike pivotiranja, do rješavanja problema minimizacije. Osim osnovnih principa, obrađuju se i napredne teme poput Charnesove M metode, koja proširuje primjenu simpleks metode na probleme s početnom nedozvoljenom točkom, te analize osjetljivosti, koja omogućuje evaluaciju utjecaja promjena ulaznih parametara na optimalno rješenje.

Konačno, praktična primjena simpleks metode ilustrirana je kroz primjenu na konkretnom inženjerskom problemu, pri čemu je rješavanje provedeno korištenjem Excel Solver-a.

Ključne riječi: Simpleks metoda, linearno programiranje, funkcija cilja, ograničenja, konveksnost, grafička metoda, simpleks tablice, pivotiranje, minimizacija, Charnesova M metoda, Excel Solver

Summary and key words

This work presents analysis of the simplex method, a fundamental algorithm in the field of linear programming. The introductory part delves into the historical development of the simplex method, emphasizing its significance and contribution to the evolution of this mathematical discipline. A detailed explanation of the mathematical framework of linear optimization problems follows, including definitions of objective functions, constraints, and the structure of linear programs themselves.

Particular attention is given to the concept of convexity and its crucial role in the application of the simplex method. Convex sets and convex functions are defined, and the convexity of linear functions is mathematically proven.

The work gradually unveils the mechanisms of the simplex method, starting from the graphical method, moving through key elements like the simplex tableau and the pivoting technique, and arriving at the solution of minimization problems. Beyond the basic principles, advanced topics are addressed, such as the Charnes M method, which extends the application of the simplex method to problems with an infeasible initial solution, and sensitivity analysis, which allows for the evaluation of the impact of changes in input parameters on the optimal solution.

Finally, the practical relevance of the simplex method is illustrated through its application to a concrete engineering problem, where the solution is obtained using Excel Solver.

Keywords: Simplex method, linear programming, objective function, constraints, convexity, graphical method, simplex tableau, pivoting, minimization, Charnes' M method, Excel Solver