Univerza v Mariboru

Fakulteta za logistiko

Tomaž Kramberger in Simona Šinko

# *OPTIMIZACIJSKE METODE V LOGISTIKI:*

## *Upravljanje s pretoki in odločanje*

Celje, 2022

Urednika: prof. dr.Tomaž Kramberger in asist. Simona Šinko, mag. inž. log.
Recenzenta: zasl. prof. ddr. Janez Usenik in doc. dr. Tea Vizinger

Jezikovni pregled: asist. dr. Vesna Mia Ipavec, univ. dipl. etn. in kult. ant

**Avtorji po poglavjih:**
Uvod – prof. dr.Tomaž Kramberger in asist. Simona Šinko, mag. inž. log.
1 Omrežni pretoki – mag. bioinf. Ivana Radić in asist. Simona Šinko, mag. inž. log.
2 Analiza ovojnice podatkov – DEA – prof. dr.Tomaž Kramberger in asist. Simona Šinko, mag. inž. log.
3 Analitično-hierarhični proces (AHP) – prof. dr.Tomaž Kramberger in asist. Simona Šinko, mag. inž. log.

**OPTIMIZACIJSKE METODE V LOGISTIKI: Upravljanje s pretoki in odločanje**
/Elektronska izdaja/

URL (pdf): https://estudij.um.si/ (dostopno z geslom)
Prva elektronska izdaja, 2022

*Izdal* Katedra za kvantitativno modeliranje v logistiki FL UM

*Založila* Fakulteta za logistiko Univerze v Mariboru, Celje

Prva izdaja, 2022

# KAZALO

# KAZALO SLIK

# SEZNAM OKRAJŠAV

| | |
|---|---|
| AHP | Analitični-hierarhični proces |
| DEA | Analiza ovojnice podatkov |
| DMU | posamezne odločitvene enote |

# BESEDA AVTORJEV

Knjiga, ki je pred vami je nadaljevanje knjige *Optimizacijske metode v logistiki: Osnovni problemi linearnega programiranja*. Bralcu, ki mu je morda kot prva prišla pod roke ta knjiga, avtorja svetujeva, da najprej poseže po prvem delu, ki obravnava osnove optimizacije in linearnega programiranja.

Brez razumevanja osnov namreč ni mogoče razumeti tega dela. V tem delu boste spoznali nekaj problemov, ki niso problemi linearnega programiranja, jih pa lahko rešujemo tudi s pomočjo linearnega programirja. Vsi problemi, ki jih obravnava knjiga, so izzivi, s katerimi se logisti srečujejo pri svojem delu.

V knjigi je naprej predstavljeno poglavje, ki obsežno obravnava omrežne pretoke. Z namenom, da bi jih bralec bolje razumel, najprej pojasnimo osnovne pojme s področja teorije grafov. V nadaljevanju obravnavamo probleme kot so: problem najkrajših poti, problem pretovora, problem maksimalnega pretoka in asignacije. Vsi problemi so pojasnjeni teoretično, potem pa sledijo še praktični primeri. Bralec se ob knjigi nauči osnov uporabe programa LINGO, ki je namenjen reševanju različnih optimizacijskih modelov.

Zatem se posvetimo metodi analize ovojnice podatkov, ki predstavlja orodje ugotavljanja učinkovitosti poljubnih realnih elementov med seboj. Njena prednost je vsekakor ta, da se elementi ne primerjajo med seboj nujno glede na zaslužek, kot je to običajno, ampak lahko za primerjavo izberemo katerikoli faktor.

Kot zadnjo tematiko predstavljamo še analitično-hierarhični proces, ki predstavlja orodje za pomoč pri odločanju. Uporabno je pri odločanju glede nakupa raznovrstne opreme, nepremičnin, odločanju glede vlaganja v novo tehnologijo, pa tudi pri čisto osebnih odločitvah.

Tudi poglavji analize ovojnice podatkov in analitično-hierarhičnega procesa sta predstavljeni na konkretnih primerih iz realnega sveta.

Avtorja upava, da so tematike predstavljene na tak način, da so bralcem razumljive in da bo knjiga pripomogla k boljšemu razumevanju obravnavanih problemov.

Ob izzidu knjige bi se iskreno zahvalila kolegici mag. bioinf. Ivani Radić, ki je prispevala k nastanku te knjige in napisala osnovo za poglavje o omrežnih pretokih. Zahvalila pa bi se tudi zasl. prof. ddr. Janezu Useniku in doc. dr. Tei Vizinger, ki sta s svojimi recenzijami in nasveti pripomogla, da je knjiga *Optimizacijske metode v logistiki*: *Upravljanje s pretoki in odločanje* zares celovita in razumljiva.

Zahvaljujeva se tudi asist. dr. Vesni Mii Ipavec, univ. dipl. etn. in kult. ant., ki je pripomogla k temu, da je knjiga berljiva in da v njej ni velikih pravopisnih napak.

Celje, januar 2022 Tomaž Kramberger in Simona Šinko

# BESEDA RECENZENTOV

Avtorja Tomaž Kramberger in Simona Šinko v knjigi *Optimizacijske metode v logistiki*: *Upravljanje s pretoki in odločanje* nadaljujeta oziroma nadgrajujeta poglavja, zajeta v knjigi *Optimizacijske metode v logistiki*: *Problemi linearnega programiranja.* Tudi ta knjiga je načeloma namenjena dodiplomskim in podiplomskim študentom Fakultete za logistiko Univerze v Mariboru, vendar JE tudi tu treba poudariti, da jo glede na širok nabor vsebin in poglobljen sistemski pristop, ilustriran z mnogimi praktičnimi primeri, lahko koristno uporabljajo tudi študenti drugih fakultet. Še več, branje priporočam vsem, ki jih optimizacijske metode zanimajo in jih želijo uspešno uporabljati v praksi, saj današnja računalniška strojna in programska oprema omogoča uporabo teh metod v zelo konkretnih problemih z velikim številom spremenljivk.

Avtorja sta knjigo razdelila na pet poglavij, pri tem pa se (upravičeno) sklicujeta in opozarjata na dejstvo, naj bralec predhodno dobro spozna teorijo linearnega programiranja.

V prvem poglavju z naslovom omrežni pretoki spoznamo osnove teorije grafov, omrežnih tokov, vozlišč in lokalnih povezav. Reševanje problemov je zaupano računalniškemu programu LINGO in Excel. V tem poglavju spoznamo še dualni program postopka analize občutljivosti. Opisane so metode in postopki pri problemih najkrajše poti, pretovora, maksimalnega pretoka in asignacije. Zanimiv in predvsem uporaben je še opis uvoza podatkov iz Excela-a v program LINGO.

Drugo poglavje nosi ime analiza ovojnice podatkov (DEA). Gre za dobrih štirideset let znano metodo merjenja/ugotavljanja učinkovitosti posameznih odločitvenih enot v celovitem postopku odločanja v dinamičnih realnih sistemih. Avtorja razložita grafični način reševanja in v nadaljevanju navezavo na postopek linearnega programiranja. Tudi tu prikažeta možnosti uporabe računalniških programov LINGO in Excel. Opisana pa je še možnost reševanja problemov ovojnice podatkov s programom DEAP (Data Envelopment Analysis Program), ki je relativno enostavna aplikacija, prijazna do uporabnika ter namenjena analizam ovojnice podatkov.

V tretjem poglavju z naslovom Analitični-hierarhični proces (AHP) avtorja predstavita temeljne principe AHP metodologije s poudarkom na več različnih odločevalcih. Opišeta tudi računalniški program za reševanje problemov tega področja.

Četrto poglavje bralca seznani z uporabo nabora podatkov v računalniškem programu LINGO, s poudarkom na uporabi zank.

Ob koncu je dodan še kratek slovensko-angleški slovarček pojmov, ki jih avtorja uporabljata v knjigi. In prav na koncu najdemo nepogrešljiv seznam uporabljene literature, ki lahko bralca vzpodbudi k širšemu spoznavanju poglavij v knjigi.

V povzetku lahko povemo, da knjiga Tomaža Krambergerja in Simone Šinko *Optimizacijske metode v logistiki*: *Upravljanje s pretoki in odločanje*, zanimivo in strokovno korektno bralcu predstavi velik nabor praktičnih optimizacijskih problemov, ki se tesno navezujejo na teorijo linearnega programiranja. Omeniti je treba tudi, da je knjiga grafično lepo opremljena. Posebej bi izpostavil še številne praktične primere, ki dajejo trdno zagotovilo, da gre za uporabna orodja optimizacije v postopku odločanja.

Novo mesto, marec 2022 prof. ddr. Janez Usenik

Knjiga Tomaža Krambergerja in Simone Šinko *Optimizacijske metode v logistiki: Upravljanje s pretoki in odločanje* predstavlja nadaljevanje 1. dela knjige, ki opisuje osnovne probleme linearnega programiranja. Primerna je za bralca, ki je že seznanjen z osnovami in želi pridobiti sofisticirana znanja iz linearnega programiranja iz področij upravljanja s pretoki in problemov odločanja.
V knjigi je najprej predstavljeno zanimivo in posebej uporabno področje omrežnih pretokov, uvedni so osnovni pojmi iz področja teorije grafov, nato so predstavljeni tipični problemi, od problema najkrajše poti, problema pretovora, maksimalnega pretoka in problema asignacije.

Avtorja nato predstavita metodo Analize ovojnice podatkov, kot tudi Analitični hierarhični proces. Obe metodologiji sta posebej primerni v procesih odločanja in zanju obstaja vrsta najrazličnejših aplikacij. Poleg uporabe računalniških orodij LINGO in Excel avtorja predstavita tudi uporabo programov DEAP in AHP. Tako kot 1. del knjige *Optimizacijske metode v logistiki* je tudi ta del napisan zanimivo, razumljivo, je primerno grafično podprt in tako lahko vsakega potencialnega bralca prepriča, da so operacijske raziskave zelo primerne in uporabne v širokem spektru odločitvenih problemov. Knjiga tako ni primerna le za študente logistike, temveč jo lahko pridoma uporabljajo tudi študentje drugih smeri, ter vsa zainteresirana javnost.

Maribor, april 2022 doc. dr. Tea Vizinger

# 1 OMREŽNI PRETOKI

V poglavju bomo opisali in prikazali postopke ter algoritme za reševanje tipičnih problemov omrežnega pretoka[1]. Med te sodijo problem pretovora[2] ali model vozlišča in lokalnih povezav[3], problem maksimalnega pretoka[4] in problem dodeljevanja virov[5], ki bodo podrobneje predstavljeni v nadaljevanju.

Za lažje razumevanje omrežnih pretokov najprej na kratko predstavimo nekaj osnov področja teorije grafov.

## 1.1 Uvod v teorijo grafov

Poglavje povzeto po Kramberger, 2010

Teorija grafov se uvršča med znane probleme matematike in teoretičnega računalništva. Njena glavna dejavnost je proučevanje lastnosti grafov.

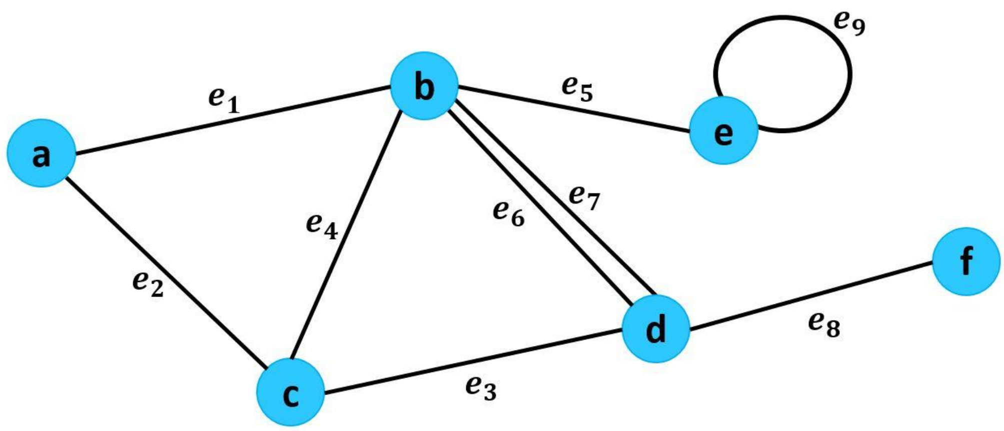
Graf $G$ je množica točk v prostoru in povezav med točkami. Graf označimo z veliko tiskano črko $G$. Graf $G = (V(G), E(G))$ je določen z množicama $V(G)\ in\ E(G)$, pri čemer množica $V(G)$ vsebuje vozlišča (točke) grafa, množica $E(G)$ pa povezave grafa. Povezavi na grafu priredimo začetno in končno vozlišče. V primeru, da sta obe krajišči povezave isto vozlišče, povezavi rečemo zanka. Povezavi, ki imata isto začetno in končno vozlišče pravimo vzporedni povezavi.

Na sliki 1 vidimo graf, sestavljen iz 6 vozlišč in 9 povezav. Množica vozlišč $V(G) = \{a, b, c, d, e, f\}$ in množica povezav $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$.

Povezava $e_1$ ima krajišči a in b. Povezava $e_9$ ima le eno vozlišče e, ki je hkrati njeno začetno in končno vozlišče. Povezavi $e_9$ rečemo zanka. Povezavi $e_6$ in $e_7$ imata isto začetno in končno vozlišče, zato sta vzporedni povezavi.

[1] Network flow problem
[2] Transshipment problem
[3] Hub and Spoke model
[4] Maximum flow problem
[5] Assignment flow problem

Slika 1: Graf G




Graf $G$ je enostaven, če ne vsebuje zank in vzporednih povezav.

Graf na sliki 1 torej ni enostaven graf, saj vsebuje zanko in vzporedne povezave.

Vsako vozlišče na grafu je lahko povezano z enim, večimi ali nobenim vozliščem. Od tega, s koliko povezavami je vozlišče v grafu povezano, je odvisna stopnja vozlišča.

Stopnja vozlišča $v \in V(G)$ je število povezav, ki se začnejo ali zaključijo v vozlišču $v$. Vsaka zanka prispeva 2 k stopnji vozlišča.

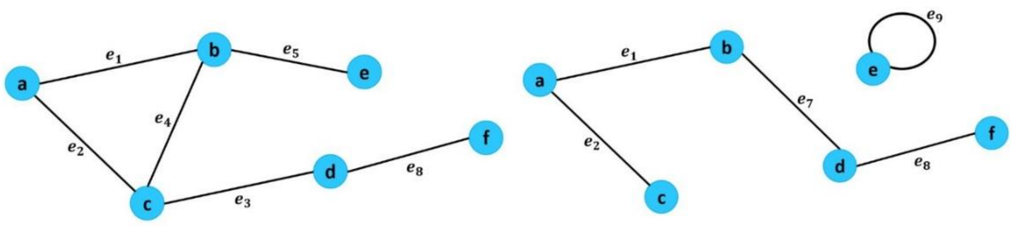
Stopnje vozlišča grafa $G$ na sliki 109 so naslednje:

| | |
|---|---|
| $st(a) = 2$ | $st(d) = 4$ |
| $st(b) = 5$ | $st(e) = 3$ |
| $st(c) = 3$ | $st(f) = 1$ |

Graf lahko vsebuje več podgrafov. »Graf $H = \big(V(H), E(H)\big)$ je podgraf grafa $G$, če velja $V(H) \subseteq V(G)$ in $E(H) \subseteq E(G)$.«

Na sliki 2 predstavljamo dva primera mnogih podgrafov grafa $G$. Vidimo, da sta množici vozlišč podgrafov podmnožici množice točk grafa, enako velja za množici povezav.

Slika 2: Podgrafa grafa G



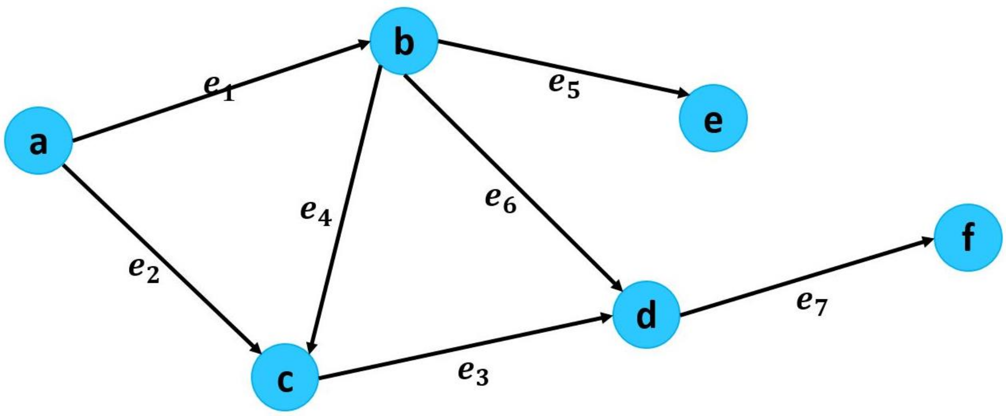

Graf je lahko povezan ali pa nepovezan. Graf je povezan kadar lahko iz poljubnega vozlišča na grafu pridemo do vsakega drugega vozlišča na grafu. V nasprotnem primeru je graf nepovezan.

Levi graf na sliki 2 je povezan, desni pa nepovezan, saj je vozlišče $e$ nepovezano z ostalimi vozlišči.

Do sedaj predstavljeni grafi v poglavju Uvod v teorijo grafov so neusmerjeni grafi. Smer povezave namreč ni določena. V nekaterih primerih pa je pomembno kako so usmerjene povezave.

Povezave v tem primeru označimo kot usmerjeno povezavo oziroma puščico. Takemu grafu rečemo usmerjeni graf.

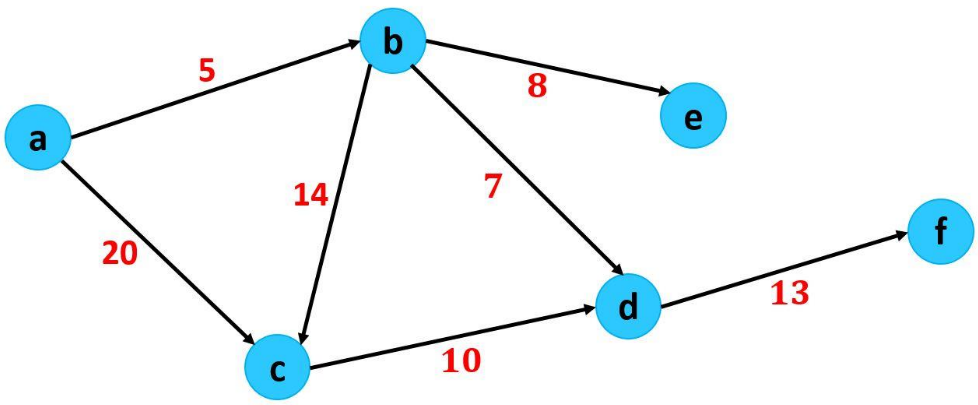
Slika 3: Usmerjeni graf G




Graf $G$ na sliki 3 je enostavni usmerjeni graf. Graf ne vsebuje zank in vzporednih povezav, zato rečemo, da je enostaven. Vse povezave v grafu so usmerjene. Iz vozlišča $b$ tako ni mogoče priti v vozlišče $a$, temveč le od vozlišča $a$ do vozlišča $b$, saj tako določa usmerjena povezava med vozliščema.

»Utežen graf je graf $G = (V(G), E(G))$ z dano preslikavo $\lambda : E(G) \rightarrow \mathbb{R}$, ki vsaki povezavi grafa priredi utež«. Uteži na povezavah pogosto poimenujemo dolžine ali cene povezav. Uteži na grafu lahko predstavljajo razdalje med posameznimi vozlišči; čas, potreben za transport med posameznimi vozlišči, cena za prevoz med vozlišči itd. V primeru, ko želimo poiskati najkrajšo povezavo med dvema vozliščema, je dolžina posameznih povezav nujen podatek.

Slika 4: Uteženi graf G




Na sliki 4 je predstavljen uteženi graf $G$. Dolžina povezave $a - b$ je 5 enot, dolžina povezave $a - c$ je 20 enot in dolžina povezave $b - c$ 14 enot. Če želimo poiskati najkrajšo pot med vozliščema $a$ in $c$, je bolje uporabiti pot $a - b - c$, saj njena dolžina znaša 19 enot, medtem ko dolžina neposredne povezave $a - c$ znaša 20 enot.

## 1.2 Uvod v omrežne tokove

Omrežni pretok je usmerjeni graf $G = (V, E)$ z medsebojno različnima vozliščema $s$ (izvor) in $t$ (ponor) v katerem je vsaka povezava grafa $(u, v) \in E$ obtežena s poljubno nenegativno kapaciteto $c(u, v)$.

Predpostavimo, da množica povezav $\mathrm{E}$ nikoli ne vsebuje vzporednih povezav – torej obratnih povezav $(u,v)$ in $(v,u)$ za en par vozlišč hkrati. Tako eliminiramo vzporedne povezave.

Prav tako velja, da je za vsak element množice vozlišč $(u,v) \in \mathrm{V}$, ki ni hkrati tudi v množici povezav $(u,v) \notin \mathrm{E}$ utež $c(u,v)$ enaka 0.

Za probleme omrežnega pretoka mora veljati tudi naslednje:

- omrežje ima eno ali več ***vhodnih vozlišč***[6], ki lahko pošiljajo tovor, sprejeti pa ga ne morejo;
- omrežje ima eno ali več ***izhodnih vozlišč***[7], ki lahko sprejmejo tovor, pošiljati pa ga ne morejo;
- omrežje ima eno ali več ***pretovornih vozlišč***[8], ki lahko tako sprejemajo kot odpošiljajo tovor.

Vsako vozlišče igra vlogo vhodnega vozlišča, izhodnega vozlišča ali pretovornega vozlišča, za katera lahko zapišemo:

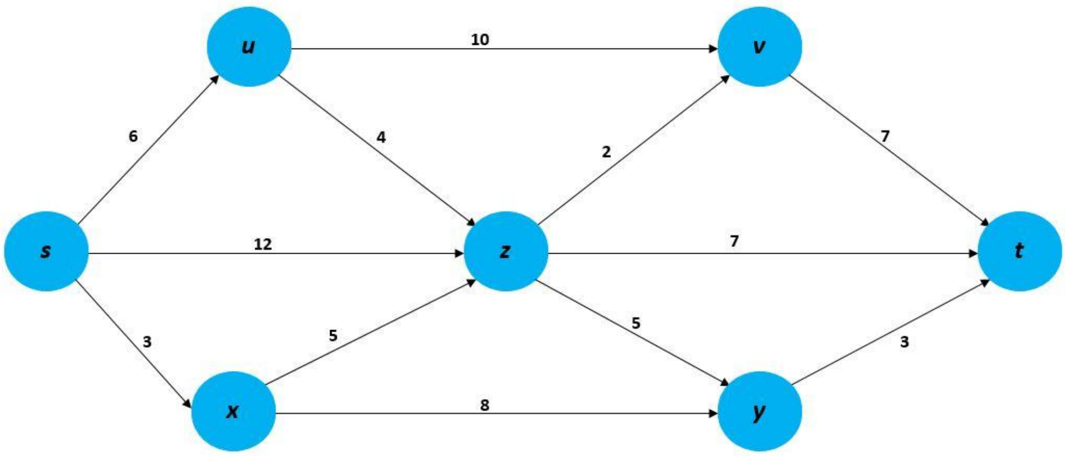
- če je vozlišče $i$ **vhodno vozlišče**, potem je povpraševanje $\boldsymbol{b_i}$ po materialu v tem vozlišču, manjše od 0. Pomeni, da je v tem vozlišču $-\boldsymbol{b}$ enot materiala v povpraševanju. Torej velja, $\boldsymbol{b_i} < \mathbf{0}$;
- če je vozlišče $i$ **izhodno vozlišče,** velja $\boldsymbol{b_i} > \mathbf{0}$. Kar pomeni, da je v tem vozlišču povpraševanje po $b$ enotah;
- če je vozlišče $i$ pretovorno vozlišče, velja $\boldsymbol{b_i} = \mathbf{0}$. V pretovornem vozlišču je 0 enot v ponudbi in prav tako 0 enot v povpraševanju.

Vsaka povezava med vozlišči omogoča pretok materiala ali storitev po omrežju, od vozlišča do vozlišča. Povezave lahko imajo določene omejitve kot so kapaciteta oz. dopustna količina materiala, ki lahko teče na tej povezavi in cena pretoka 1 enote materiala na tej povezavi.

[6] supply point
[7] demand point
[8] trans-shippment point

Za lažjo predstavo si omrežje, po katerem želimo poslati določeno količino blaga. Iz začetnega vozlišča $s$ pošljemo želeno količino blaga. Blago, odposlano iz izvora $s$ mora prispeti v končno vozlišče $t$. Blago lahko potuje po povezavah med vozlišči kot je predstavljeno na sliki 5. Pri tem ima vsaka povezava svojo ceno na enoto prepeljanega blaga. Želimo, da iz vozlišča $s$ odpotuje 20 enot blaga in ga toliko tudi prispe v vozlišče $t$.

Slika 5: Omrežni pretok




Slika 5 prikazuje usmerjeni graf s sedmimi vozlišči in povezavami med njimi. Vsako omrežje je mogoče opisati z incidenčno matriko. Incidenčna matrika je matrika dimenzij $n \times m$, ki predstavlja graf z $n$ vozlišči in $m$ povezavami. Stolpci imajo samo po dva od nič različnih elementov. V usmerjenih grafih brez zank imajo enkrat vrednost 1 (kadar vozlišče predstavlja začetno vozlišče povezave) in enkrat -1 (kadar vozlišče predstavlja končno vozlišče povezave).

Incidenčna matrika za omrežje, predstavljeno na sliki 5, bo matrika dimenzij 7 x 12, saj ima predstavljeno omrežje sedem vozlišč in 12 povezav med vozlišči. Vozlišča predstavljajo vrstice v matriki, obstoječe povezave pa stolpce.

| | $(s,u)$ | $(s,x)$ | $(s,z)$ | $(u,v)$ | $(u,z)$ | $(x,z)$ | $(x,y)$ | $(z,v)$ | $(z,t)$ | $(z,y)$ | $(v,t)$ | $(y,t)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | | | | | | | | | | | | |
| $u$ | | | | | | | | | | | | |
| $x$ | | | | | | | | | | | | |
| $z$ | | | | | | | | | | | | |
| $v$ | | | | | | | | | | | | |
| $y$ | | | | | | | | | | | | |
| $t$ | | | | | | | | | | | | |

Z vrednostmi 1, -1 in 0 opišemo odnos med opazovanim vozliščem in povezavo:

- 1 opisuje, da povezava vodi v vozlišče;
- -1 opisuje, da povezava vodi iz vozlišča;
- 0 pa opisuje, da povezava ni povezana z opazovanim vozliščem (zaradi preglednosti incidenčne matrike vrednost 0 izpustimo in pustimo prazne celice).

Povezava $(s, u)$ vodi iz vozlišča $s$, zato odnos med vozliščem $s$ in povezavo $(s, u)$ označimo z -1. Ravno tako iz vozlišča $s$ vodita povezavi $(s, x)$ in $(s, z)$, zato tudi ta odnosa ozačimo z -1. Vse ostale povezave z vozliščem $s$ nimajo nič, torej vse ostale celice v vrstici vozlišča $s$ pustimo prazne.

Povezava $(s, u)$ vodi v vozlišče $u$, odnos med vozliščem $u$ in povezavo $(s, u)$ označimo z 1. Povezavi $(u, v)$ in $(u, z)$ vodita iz vozlišča, torej odnosu dodelimo oznako -1.

Na isti način označimo še ostale odnose med vozlišči in povezavami, da dobimo spodnjo matriko:

| | $(s,u)$ | $(s,x)$ | $(s,z)$ | $(u,v)$ | $(u,z)$ | $(x,z)$ | $(x,y)$ | $(z,v)$ | $(z,t)$ | $(z,y)$ | $(v,t)$ | $(y,t)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | -1 | -1 | -1 | | | | | | | | | |
| $u$ | 1 | | | -1 | -1 | | | | | | | |
| $x$ | | 1 | | | | -1 | -1 | | | | | |
| $z$ | | | 1 | | 1 | 1 | | -1 | -1 | -1 | | |
| $v$ | | | | 1 | | | | 1 | | | -1 | |
| $y$ | | | | | | | 1 | | | 1 | | -1 |
| $t$ | | | | | | | | | 1 | | 1 | 1 |

S primerom smo pokazali, da ima vsak stolpec res samo dva od nič različna elementa. Ta lastnost izhaja iz lastnosti usmerjenega omrežja, kjer vsaka povezava enemu vozlišču predstavlja vhodno povezavo, drugemu pa izhodno povezavo.

V incidenčno matriko v zadnjo vrstico zapišemo tudi vrednosti uteži $c(u,v)$, v zadnji stolpec pa vrednosti ponudbe oziroma povpraševanja v vozliščih.

Iz slike 5 lahko razberemo, da je utež povezave $(s,u)$ 6 enot, utež povezave $(s,z)$ 12 enot in $(s,x)$ 3 enote. Vpišemo uteži vseh povezav v dodatno vrstico v incidenčni matriki.

| | $(s,u)$ | $(s,x)$ | $(s,z)$ | $(u,v)$ | $(u,z)$ | $(x,z)$ | $(x,y)$ | $(z,v)$ | $(z,t)$ | $(z,y)$ | $(v,t)$ | $(y,t)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | -1 | -1 | -1 | | | | | | | | | |
| $u$ | 1 | | | -1 | -1 | | | | | | | |
| $x$ | | 1 | | | | -1 | -1 | | | | | |
| $z$ | | | 1 | | 1 | 1 | | -1 | -1 | -1 | | |
| $v$ | | | | 1 | | | | 1 | | | -1 | |
| $y$ | | | | | | | 1 | | | 1 | | -1 |
| $t$ | | | | | | | | | 1 | | 1 | 1 |
| $c$ | 6 | 3 | 12 | 10 | 4 | 5 | 8 | 2 | 7 | 5 | 7 | 10 |

Pri definiciji primera smo zapisali, da želimo, da iz se iz začetnega vozlišča $s$ odpelje 20 enot blaga, v končno vozlišče $t$ pa prispe enaka količina. To pomeni, da v vmesnih vozliščih ne sme ostati noben kos blaga, kajti le tako bomo zadostili povpraševanju v vozlišču $t$. Povpraševanje v vmesnih vozliščih je torej 0. Zapišimo še količino povpraševanja in ponudbe v stolpec $b$.

| | $(s,u)$ | $(s,x)$ | $(s,z)$ | $(u,v)$ | $(u,z)$ | $(x,z)$ | $(x,y)$ | $(z,v)$ | $(z,t)$ | $(z,y)$ | $(v,t)$ | $(y,t)$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | -1 | -1 | -1 | | | | | | | | | | -20 |
| $u$ | 1 | | | -1 | -1 | | | | | | | | 0 |
| $x$ | | 1 | | | | -1 | -1 | | | | | | 0 |
| $z$ | | | 1 | | 1 | 1 | | -1 | -1 | -1 | | | 0 |
| $v$ | | | | 1 | | | | 1 | | | -1 | | 0 |
| $y$ | | | | | | | 1 | | | 1 | | -1 | 0 |
| $t$ | | | | | | | | | 1 | | 1 | 1 | 20 |
| $c$ | 6 | 3 | 12 | 10 | 4 | 5 | 8 | 2 | 7 | 5 | 7 | 10 | |

Pri vozliščih, iz katerih pošiljamo blago, ponudbo označimo z negativnim predznakom, saj ta definira količino izhodnega toka blaga.

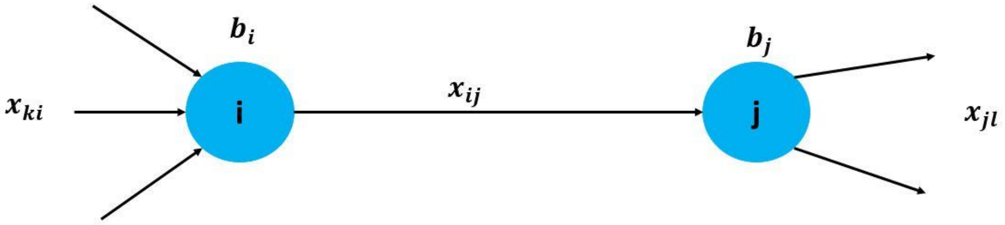
Na opisan način lahko torej definiramo vsako omrežje s pomočjo incidenčne matrike, ki nam je v pomoč pri pisanju linearnega programa. V nadaljevanju bomo predstavili nekatere probleme omrežnega pretoka in si pri zapisu linearnega programa pomagali z incidenčno matriko.

Povezava iz vozlišča $i$ v vozlišče $j$ je $i \xrightarrow[\left[l_{ij},k_{ij}\right]]{c_{ij}} j$, pri čemer je $c_{ij}$ cena enote materiala ali storitve na tej povezavi, $l_{ij}$ ter $k_{ij}$ pa predstavljata spodnjo in zgornjo mejo omejitve s kapaciteto te povezave.

S pomočjo slike 6 bomo definirali nekatere spremenljivke[9], ki opišejo dano omrežje in s pomočjo katerih bomo lahko v nadaljevanju zapisali linearni program.

[9] Variables

Slika 6: Prikaz povezave dveh vozlišč v omrežju




Definiramo odločitveno spremenljivko $x_{ij}$, ki predstavlja število enot materiala ali storitev, poslanih po povezavi, od vozlišča $i$ do vozlišča $j$. Nadomestimo v zapisani incidenčni matriki povezave s pripadajočimi odločitvenimi spremenljivkami.

| | $x_{su}$ | $x_{sx}$ | $x_{sz}$ | $x_{uv}$ | $x_{uz}$ | $x_{xz}$ | $x_{xy}$ | $x_{zv}$ | $x_{zt}$ | $x_{zy}$ | $x_{vt}$ | $x_{yt}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | -1 | -1 | -1 | | | | | | | | | | -20 |
| $u$ | 1 | | | -1 | -1 | | | | | | | | 0 |
| $x$ | | 1 | | | | -1 | -1 | | | | | | 0 |
| $z$ | | | 1 | | 1 | 1 | | -1 | -1 | -1 | | | 0 |
| $v$ | | | | 1 | | | | 1 | | | -1 | | 0 |
| $y$ | | | | | | | 1 | | | 1 | | -1 | 0 |
| $t$ | | | | | | | | | 1 | | 1 | 1 | 20 |
| $c_{ij}$ | 6 | 3 | 12 | 10 | 4 | 5 | 8 | 2 | 7 | 5 | 7 | 10 | |

Distribucijo blaga po omrežju označimo z $f$[10].

Iščemo minimum vsote celotne cene po enoti prepeljanega materiala ali storitev po omrežju, torej je namenska funkcija[11] sledeča:

$$\min f = \sum_{ij} c_{ij} \cdot x_{ij}$$

Z omejitvami sistema omrežja:

$$\sum_{k} x_{ki} - \sum_{l} x_{jl} \gtreqless b_i$$

[10] V poglavju omrežni pretoki namensko funkcijo označujemo s $f$ (flow)
[11] Objective function

ki se nanašajo na ravnovesni tok vozlišča. Razlika med vhodnim tokom (količino prispelega blaga) v vozlišče in izhodnim tokom (količino odpeljanega blaga) iz vozlišča je lahko enaka, večja ali manjša od povpraševanja ali ponudbe tega vozlišča.

Pri zapisu linearnega programa obstaja še en pogoj, ki se nanaša na omejitve kapacitete vsake povezave v omrežju in je določena z zgornjo in spodnjo mejo. V dosedanjih problemih, ki smo jih spoznali, smo opredelili zgolj pogoj nenegativnosti, lahko pa stvar malce nadgradimo in odločitveno spremenljivko omejimo tako z zgornjo kot s spodnjo mejo. Zapišemo:

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

Tok materiala ali storitev lahko zavzame vrednosti med $l_{ij}$ in $u_{ij}$.

V nadaljevanju se bomo osredotočili na zapis omejitev[12] sistema, ki se lahko razlikujejo od problema do problema. Glavno vodilo za zapis omejitev sta povpraševanje in ponudba določenega vozlišča zasnovanega problema:

- če je povpraševanje enako ponudbi, potem velja:
$$[\sum tokov\ v\ vozlišče\ i] - [\sum tokov\ iz\ vozlišča\ i] = b_i;$$
- če je ponudba manjša od povpraševanja in je potrebno odpremiti vso količino blaga, ki je na voljo, potem velja:
$$[\sum tokov\ v\ vozlišče\ i] - [\sum tokov\ iz\ vozlišča\ i] \leq b_i;$$
- če je ponudba večja od povpraševanja in želimo zadostiti celotnemu povpraševanju po materialu v $i$-tem vozlišču, potem velja:
$$[\sum tokov\ v\ vozlišče\ i] - [\sum tokov\ iz\ vozlišča\ i] \geq b_i.$$

Iz zapisanega lahko sklepamo, da se bodo omejitve pri vsakem vozlišču spreminjale glede na povpraševanje in ponudbo tega vozlišča. Npr. lahko da je povpraševanje v vozlišču 1 po vsaj 30 enotah nekega materiala, v vozlišču 2 pa je povpraševanje po točno 30 enotah nekega materiala. V prvem primeru bomo zapisali omejitev z neenačajem, medtem ko bo omejitev drugega vozlišča striktno enaka povpraševanju.

[12] Contrains

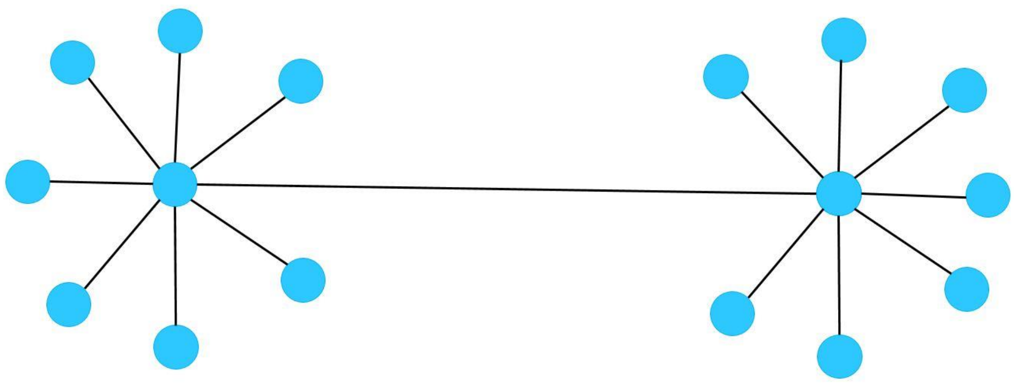
*Pomnimo:* število odločitvenih spremenljivk v omrežju je enako številu povezav v omrežju. Število omejitev, ki se nanašajo na ravnovesni tok vozlišča je enako številu vozlišč v omrežju.

## 1.3 Model vozlišča in lokalnih povezav

Poglavje povzeto po Bradley, Hax & Magnanti, 1977

Model vozlišča in lokalnih povezav[13] je poseben primer omrežnih pretokov. Promet se v omrežjih po principu vozlišč[14] in lokalnih povezav[15] preusmerja iz številnih oskrbovalnih vozlišč do vozlišč, kjer nastaja povpraševanje oz. končnih vozlišč. Med njimi se nahajajo pretovorna vozlišča, kjer se blago lahko združuje v večje pošiljke ali pa se prerazporedi, kar lahko morda zniža stroške transporta. Model je bil razvit leta 1987 (O'Kelly, 1987) in je v zadnjih letih močno vplival na poslovanje podjetij, med katerimi je največji razmah doživel v letalskem prometu, prevozu tovora in telekomunikacijskih storitvah.

Slika 7: Osnovni model vozlišča in lokalnih povezav



Na sliki 7 predstavljamo osnovni model vozlišča in lokalnih povezav. Na njej sta dve osrednji ali centralni vozlišči, iz katerih vodijo povezave do vseh ostalih vozlišč v grafu.

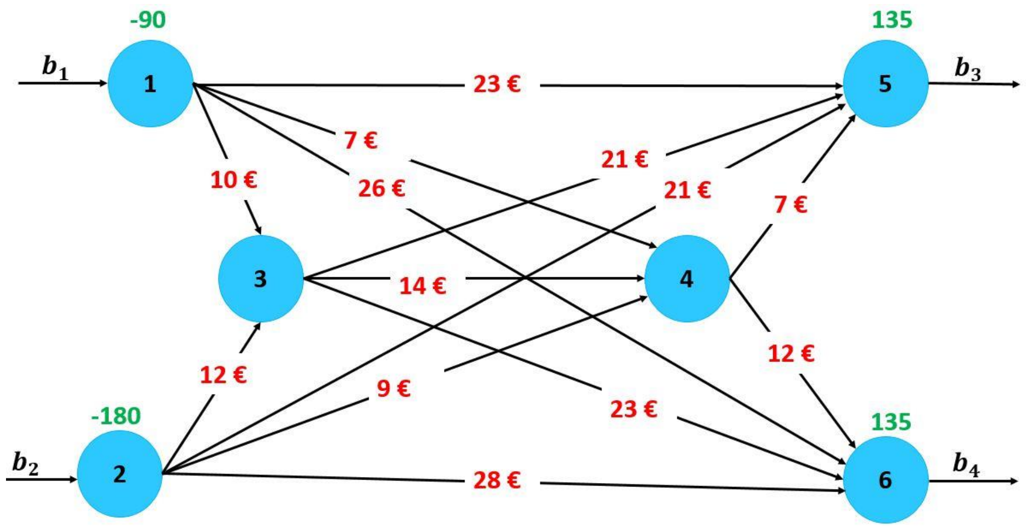
[13] Hub and Spoke model
[14] hub
[15] spoke

**Primer 1:**

Poglavje povzeto po Thore & Iannone, 2005

Poglejmo model vozlišča in lokalnih povezav na konkretnem primeru letalske industrije. Podatki problema so podani s sliko omrežja (podana je cena prevoza na povezavi za 1 enoto) in z incidenčno matriko. Incidenčna matrika je sestavljena tako, da stolpci predstavljajo povezave, vrstice pa vozlišča. Incidenčna matrika nam pokaže, katera vozlišča so vsebovana v posameznih povezavah.

Slika 8: Omrežje modela vozlišča in lokalnih povezav




Vozlišči 1 in 2 predstavljata izhodiščno letališče, kjer skladiščijo določeno vrsto blaga. Vidimo, da imajo na letališču 1 90 enot blaga, na letališču 2 pa 180 enot blaga. Blago lahko potuje čez vozlišče 3 ali vozlišče 4, lahko pa potuje tudi neposredno na končni vozlišči 5 in 6. Končni vozlišči 5 in 6 predstavljata letališči kupcev blaga, pri čemer oba kupca povprašujeta po 135 enotah blaga. Želimo opraviti takšen transport blaga skozi vozlišče, da bodo transportni stroški najmanjši.

Slika 9: Incidenčna matrika

| | | povezave | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | (1,3) | (1,4) | (1,5) | (1,6) | (2,3) | (2,4) | (2,5) | (2,6) | (3,4) | (3,5) | (3,6) | (4,5) | (4,6) |
| vozlišča | 1 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 2 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 |
| | 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 |
| | 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 |
| | 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| | 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

Podane podatke na slikah 8 in 9 zapišemo v enotno tabelo, kjer za problem že definiramo odločitvene spremenljivke. Želimo določiti koliko enot blaga naj se pelje po povezavah v omrežju, torej:

$$x_{ij} = \check{s}tevilo\ prepeljanih\ enot\ blaga\ po\ določeni\ povezavi,$$

pri čemer $i$ predstavlja zaporedno številko vozlišča, iz katerega blago peljemo in $j$ zaporedno številko vozlišča, v katero peljemo blago po povezavi.

| | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $x_{23}$ | $x_{24}$ | $x_{25}$ | $x_{26}$ | $x_{34}$ | $x_{35}$ | $x_{36}$ | $x_{45}$ | $x_{46}$ | b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $V1$ | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\geq -90$ |
| $V2$ | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | $\geq -180$ |
| $V3$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 | $\geq 0$ |
| $V4$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 | $\geq 0$ |
| $V5$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | $\geq 135$ |
| $V6$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | $\geq 135$ |
| Cena | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | |

V prvem stolpcu so navedena vsa vozlišča v danem omrežju. V prvi vrstici so zapisane odločitvene spremenljivke, ki pomenijo število prepeljanih enot po posamezni povezavi. V zadnji vrstici so navedena cene vsake povezave, po kateri bo prepeljano določeno število enot, zadnji stolpec pa predstavlja količino povpraševanja in ponudbe v posameznem vozlišču.

Iz izrisanega omrežja vidimo, da ima vozlišče 1 v ponudbi 90 enot, vozlišče 2 pa ima v ponudbi 180 enot. Vozlišči 3 in 4 igrata vlogo pretovornega vozlišča. V tem konkretnem primeru v pretovornih vozliščih nimamo zahtev po materialu (povpraševanje je 0) in prav tako nimamo ponudbe materiala. Vozlišči 5 in 6 sta povpraševalni vozlišči, z zahtevo po 135 enotah blaga za vsako od obeh vozlišč. Povezave med vozlišči so opremljene s cenami na enoto transportiranega blaga.

Cilj rešitve modela vozlišča in lokalnih povezav je izvesti takšno distribucijo blaga skozi omrežje, da bodo skupni stroški transporta na koncu najmanjši. Iskali bomo torej minimum funkcije stroškov.

Skupna cena posamezne povezave bo na koncu odvisna od tega, koliko enot blaga bo po posamezni povezavi prepeljanih. Namenska funkcija mora zajeti vsoto stroškov po vseh povezavah v omrežju.

**Minimum namenske funkcije**

$$\begin{aligned} min\ \mathrm{f}(x_{ij}) = {} & 10x_{13} + 7x_{14} + 23x_{15} + 26x_{16} + 12x_{23} + 9x_{24} + 21x_{25} + 28x_{26} \\ & + 14x_{34} + 21x_{35} + 23x_{36} + 7x_{45} + 12x_{46} \end{aligned}$$

**Omejitve** - ravnotežni tok vozlišč

V predstavljenem primeru je ponudba enaka povpraševanju, zato mora biti vsota količine blaga, ki prispe v vozlišče, enaka vsoti blaga, ki zapusti vozlišče. Ali povedano z drugimi besedami: v notranjih vozliščih se blago ne kopiči:

$$\left[\sum pripeljanega\ blaga\ v\ i\right] - \left[\sum odpeljanega\ blaga\ iz\ i\right] = b_i$$

Če želimo minimirati stroške transporta v omrežju, bomo s pomočjo linearnega programa prišli do rešitve, da se po določeni povezavi pošlje natanko toliko enot, kot je povpraševanje v vozlišču, v katerega vodi povezava in nič več. Zato lahko namesto zapisane omejitve s strogim enačajem $[\sum pripeljanega\ blaga\ v\ i] - [\sum odpeljanega\ blaga\ iz\ i] = b_i$, uporabimo obliko $[\sum pripeljanega\ blaga\ v\ i] - [\sum odpeljanega\ blaga\ iz\ i] \geq b_i$. Ker bomo na tem primeru pogledali tudi izpeljavo dualnega programa, nam bo takšna oblika zapisa omejitev kasneje v pomoč.

Pri pisanju omejitev si lahko sedaj pomagamo z incidenčno matriko. Množimo vrstico za posamezno vozlišče (kjer imamo definiran odnos med vozliščem in povezavami) z definiranimi odločitvenimi spremenljivkami. Desno stran neenačb v omejitvah pa predstavlja količina ponudbe/povpraševanja v vozlišču, ki smo jo zapisali v zadnji stolpec matrike.

| | | |
|---|---|---|
| Vozlišče 1: | $-x_{13} - x_{14} - x_{15} - x_{16} \geq -90$ | Količina blaga, ki zapusti vozlišče 1 ne more preseči 90 enot, saj jih je na voljo samo toliko. |
| Vozlišče 2 | $-x_{23} - x_{24} - x_{25} - x_{26} \geq -180$ | Količina blaga, ki zapusti vozlišče 2 ne more preseči 180 enot, saj jih je na voljo samo toliko. |
| Vozlišče 3: | $x_{13} + x_{23} - x_{34} - x_{35} - x_{36} \geq 0$ | Vsota blaga, ki vstopi v pretovorno vozlišče 3 mora biti enaka ali večja vsoti izhoda iz pretovornega vozlišča, če želimo zadostiti potrebam po enotah na izhodnem vozlišču 5 in 6. |
| Vozlišče 4: | $x_{14} + x_{24} + x_{34} - x_{45} - x_{46} \geq 0$ | Vsota blaga, ki vstopi v pretovorno vozlišče mora biti enaka ali večja vsoti izhoda iz pretovornega vozlišča, če želimo zadostiti potrebam po enotah na izhodnem vozlišču 5 in 6. |
| Vozlišče 5: | $x_{15} + x_{25} + x_{35} + x_{45} \geq 135$ | Količina blaga, ki prispe v končno vozlišče 5 je lahko večja ali enaka povpraševanju – da zadosti potrebam po povpraševanju. |
| Vozlišče 6: | $x_{16} + x_{26} + x_{36} + x_{46} \geq 135$ | Količina blaga, ki prispe v končno vozlišče 6 je lahko večja ali enaka povpraševanju – da zadosti potrebam po povpraševanju. |

## Pogoj nenegativnosti

$$x_{ij} \geq 0; i = 1, 2, 3, 4 \; in \; j = 3, 4, 5, 6$$

### 1.3.1 Reševanje vozlišča in lokalnih povezav s pomočjo računalniškega programa LINGO

V program LINGO vnesemo linearni program v sintaksi programa:

```
!Namenska funkcija;
min = 10*x13+ 7*x14 + 23*x15 + 26*x16+ 12*x23+9*x24 + 21*x25
+ 28*x26+ 14*x34 + 21*x35 + 23*x36+ 7*x45 + 12*x46;
!Omejitve;
-x13- x14 - x15 - x16  >= -90;
-x23- x24 - x25 - x26  >= -180;
x13+ x23 - x34 - x35  - x36 >= 0;
x14+ x24 + x34 - x45  - x46 >= 0;
x15+ x25 + x35 + x45  >= 135;
x16+ x26 + x36 + x46  >= 135;
```

Poročilo o rešitvi[16] za model vozlišča in lokalnih povezav:

```
  Global optimal solution found.
  Objective value:                              4815.000
  Infeasibilities:                              0.000000
  Total solver iterations:                            10
  Elapsed runtime seconds:                          0.03

  Model Class:                                        LP

                              Variable           Value        Reduced Cost
                                   X13        0.000000            12.00000
                                   X14        90.00000            0.000000
                                   X15        0.000000            9.000000
                                   X16        0.000000            7.000000
                                   X23        0.000000            12.00000
                                   X24        180.0000            0.000000
                                   X25        0.000000            5.000000
                                   X26        0.000000            7.000000
                                   X34        0.000000            5.000000
                                   X35        0.000000            5.000000
                                   X36        0.000000            2.000000
                                   X45        135.0000            0.000000
                                   X46        135.0000            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        4815.000           -1.000000
                                     2        0.000000           -2.000000
                                     3        0.000000            0.000000
                                     4        0.000000            0.000000
                                     5        0.000000           -9.000000
                                     6        0.000000           -16.00000
                                     7        0.000000           -21.00000
```

[16] Solution Report

Optimalna rešitev za dani problem znaša 4.812 € in pomeni minimalni strošek transporta enot blaga po poti skozi omrežje tako, da zadosti potrebam povpraševanja v vozliščih 5 in 6. Pot in količine za transport enote blaga, da bo strošek transporta minimalen, so sledeče:

- 90 enot blaga naj prepeljejo iz vozlišča 1 do vozlišča 4;
- 180 enot blaga naj prepeljejo iz vozlišča 2 do vozlišča 4;
- od vozlišča 4 do vozlišča 5 se bo peljalo 135 enot blaga;
- in prav tako iz vozlišča 4 do vozlišča 6 135 enot blaga.

Ob takšni distribuciji blaga bo strošek transporta najnižji.

**Reducirani strošek** predstavlja spremembo optimalne vrednosti namenske funkcije, če se spremeni pogoj spremenljivke (oziroma povezave), ki ni vključena v končno rešitev iz 0 na 1. V tem primeru se ta spremenljivka pojavi kot del rešitve. Preverimo kaj se zgodi, če želimo vozlišče 3 vključiti v rešitev oz. želimo, da se vsaj ena enota blaga prepelje skozi vozlišče 3. V ta namen spremenimo pogoj nenegativnosti spremenljivke $x_{13}$ tako, da povečamo desno stran neenačbe za 1 enoto: $x_{13} \geq 1$.

Sintaksa zapisa linearnega programa je enaka kot v prejšnjem primeru, z vključenim dodatnim pogojem $x_{13} \geq 1$. Izpis LINGO rešitve v primeru, da želimo $x_{13}$ vključiti v rešitev:

```
  Global optimal solution found.
Objective value:                              4829.000
Infeasibilities:                              0.000000
Total solver iterations:                            10
Elapsed runtime seconds:                          0.04

Model Class:                                        LP

                              Variable           Value        Reduced Cost
                                   X13        1.000000            0.000000
                                   X14        89.00000            0.000000
                                   X15        0.000000            9.000000
                                   X16        0.000000            7.000000
                                   X23        0.000000            14.00000
                                   X24        180.0000            0.000000
                                   X25        0.000000            5.000000
                                   X26        0.000000            7.000000
                                   X34        0.000000            3.000000
                                   X35        0.000000            3.000000
                                   X36        1.000000            0.000000
                                   X45        135.0000            0.000000
                                   X46        134.0000            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        4829.000           -1.000000
                                     2        0.000000           -4.000000
                                     3        0.000000           -2.000000
                                     4        0.000000            0.000000
                                     5        0.000000           -11.00000
                                     6        0.000000           -18.00000
                                     7        0.000000           -23.00000
                                     8        0.000000           -14.00000
```

Vrednost namenske funkcije se je spremenila za $14$ €, spremenljivki $x_{13}$ in $x_{36}$ pa sta del rešitve. Ker gre za omrežje (ohranjanje pretoka skozi vozlišča) povišanje vrednosti spremenljivke $x_{13}$ za 1 enoto posledično pomeni tudi povišanje ene izmed spremenljivk, ki izhajajo iz vozlišča 3. Tako, da je skupna vrednost, za katero se bo spremenila optimalna vrednost namenske funkcije, vsota reduciranih stroškov tistih spremenljivk, ki so se povišale za 1 enoto.

Reduciran strošek spremenljivke $x_{13}$ je enak 12 €, spremenljivke $x_{36}$ pa 2 €, kar je točno toliko, za kolikor se je povišala vrednost namenske funkcije z dodatnim pogojem. V novi rešitvi se pojavita tudi povezavi $(1{,}3)$ in $(3{,}6)$ po katerih se prepelje 1 enota blaga. Bolj podrobno analizo si bomo pogledali v nadaljevanju.

### 1.3.2 Dualni problem in analiza občutljivosti[17]

Rešitev linearnega programa lahko poda več informacij o modelu ali problemu kot zgolj vrednosti odločitvenih spremenljivk in optimalne vrednosti namenske funkcije. Vsakemu linearnemu programu lahko priredimo dualni program.

Analiza dualnega linearnega problema lahko poda tudi informacije o tem, za koliko se lahko spremenijo vrednosti koeficientov namenske funkcije brez spremembe optimalne rešitve. V nadaljevanju bomo primeru 60 priredili dualni linearni problem.

V dualnem programu uvedemo dualne spremenljivke, ki so vezane na vozlišča in jih označimo z $u_i$. Poglejmo povezavo med vozliščem 1 in vozliščem 3. Senčna cena[18] za vozlišče 1 je $u_1$ in za vozlišče 3 je $u_3$.

Tabela s podatki za dualni linearni program primera 1:

| Senčne cene | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $x_{23}$ | $x_{24}$ | $x_{25}$ | $x_{26}$ | $x_{34}$ | $x_{35}$ | $x_{36}$ | $x_{45}$ | $x_{46}$ | b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\geq -90$ |
| $u_2$ | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | $\geq -180$ |
| $u_3$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 | $\geq 0$ |
| $u_4$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 | $\geq 0$ |
| $u_5$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | $\geq 135$ |
| $u_6$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | $\geq 135$ |
| Cena | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | |

v tabeli $u_i$; $i = 1, \dots, 6$ predstavlja senčno ceno toka blaga pri določenem vozlišču v omrežju. Vsako vozlišče ima svojo senčno ceno.

V primeru dualnega linearnega programa želimo maksimirati skupno povečanje (akumuliranje) senčne cene v omrežju ($vrednost\ pretoka\ v\ omrežju \times senčna\ cena$).

[17] Range report
[18] Shadow price / Marginal value / Pi value / Dual value

To dobimo z razliko senčne cene toka transporta ob izhodu iz omrežja ($135u_5 + 135u_6$) in senčne cene transportnega toka na vhodu v omrežje ($90u_1 + 180u_2$). Skupna vrednost senčne cene pretoka v omrežju je enaka:

$$135u_5 + 135u_6 - 90u_1 - 180u_2.$$

Za vsako povezavo v omrežju določimo dualne omejitve. Pomembno načelo izčrpavanja vrednosti pravi, da mora biti senčna cena, ob vsakem pozitivnem toku v omrežju, v celoti izčrpana s transportnim stroškom 1 enote.

Po načelu izčrpavanja vrednosti lahko zapišemo: $u_3 \leq u_1 + 10$ ali $-u_1 + u_3 \leq 10$. Senčna cena v vozlišču 3 ne more preseči vsote senčne cene vozlišča 1 in cene za transport 1 enote materiala od vozlišča 1 do vozlišča 3. Če obstaja pozitiven tok med dvema vozliščema $(1,3)$, potem je senčna cena v vozlišču 3 natanko enaka senčni ceni vozlišča 1 plus transportnim stroškom ene enote materiala med vozliščema – takrat dosežemo prelomno točko, kjer pokrijemo stroške transporta.

Splošni zapis za omejitve dualnega programa:

$$-u_i + u_j \leq c_i.$$

Kadar v osnovnem linearnem problemu iščemo minimum stroškov, v dualnem programu maksimiramo vrednosti. Dualni linearni program za dano omrežje je torej:

$$\max w(u_i) = 135u_5 + 135u_6 - 90u_1 - 180u_2$$

pri omejitvah:

$$-u_1 + u_3 \leq 10$$
$$-u_1 + u_4 \leq 7$$
$$-u_1 + u_5 \leq 23$$
$$-u_1 + u_6 \leq 26$$
$$-u_2 + u_3 \leq 12$$
$$-u_2 + u_4 \leq 9$$
$$-u_2 + u_5 \leq 21$$
$$-u_2 + u_6 \leq 28$$
$$-u_3 + u_4 \leq 14$$

$$-u_3 + u_5 \leq 21$$
$$-u_3 + u_6 \leq 23$$
$$-u_4 + u_5 \leq 7$$
$$-u_4 + u_6 \leq 12$$
$$u_1, u_2 u_3, u_{4,} u_5, u_6 \geq 0$$

Z namensko funkcijo smo določili, da želimo maksimirati skupno vrednost senčne cene preko omrežja ob pogojih, da se višanje senčne vrednosti izčrpava iz cene transporta1 enote po povezavi ter pogoja nenegativnosti. Senčna cena ne more biti negativna. Lahko je 0 in sicer v primeru, ko se blago kopiči v kakšnem vozlišču in tam nastane neželen presežek.

Iz osnovnega programa smo ugotovili, da so aktivne le povezave $(1,4), (2,4), (4,5), (4,6)$. Izključno in samo vzdolž teh povezav je seštevek senčne cene enote prepeljanega blaga enaka stroškom prepeljanega blaga na enoto. Vzdolž vseh ostalih povezav bo seštevek senčne cene enote nižji od stroškov transporta enote, zato se hipotetičnemu pošiljatelju ne bo zdelo vredno opraviti prevoza vzdolž teh povezav.

Tako je senčna cena vzdolž povezave $(1,4)$ enaka 7, vzdolž povezave $(2,4)$ enaka 9, vzdolž povezave $(4,5)$ je enaka 7 in vzdolž povezave $(5,6)$ je enaka 12.

Izkaže se, da pod temi pogoji opišemo le večanje senčne cene enote blaga vzdolž vsake od teh povezav v omrežju in ne absolutne senčne cene. Označimo senčno vrednost enote pri vozlišču $2$ z $a$, ki predstavlja poljubno pozitivno vrednost.

Če imamo v vozlišču $2$ neko senčno ceno $\mathrm{a}$ in ob enem ceno transporta na enoto po povezavi $(2,4)$ enako $9$, potem je absolutna senčna cena pri vozlišču $4$ enaka $a + 9$. Sedaj izhajamo iz vozlišča 4 do vozlišča 1 po povezavi $(4,1)$. Če je senčna cena vozlišča 4 enaka $a + 7$ in je strošek na enoto povezave $(4,1)$ enak 7, potem je senčna cena pri vozlišču 1 enaka $a + 7 - 5 = a + 2$. Podobno določimo senčne cene pri vozliščih 5 in 6, ki sta enaki $\mathrm{a} + 16$ in $\mathrm{a} + 21$.

Tako je optimalna vrednost dualnega programa enaka:

$$-90(a+2) - 180a + 135(a+16) + 135(a+21) = 4815$$

Za izhodno senčno ceno enote blaga, s pomočjo katere računamo vrednosti v vseh vozliščih omrežja, lahko poljubno izberemo katerokoli izmed izvornih vozlišč.

Iz zapisane enačbe vidimo, da se konstanta $a$ izniči in jo lahko izpustimo. Vidimo, da je optimalna vrednost primarnega linearnega programa enaka optimalni vrednosti dualnega programa. Upoštevano je načelo izčrpavanja vrednosti: skupno povečanje senčne vrednosti preko celotnega omrežja je enako skupnim stroškom prevoza v omrežju.

### 1.3.3 Reševanje dualnega programa s pomočjo programa LINGO

Sintaksa dualnega problema v programu LINGO je sledeča:

```
!Namenska funkcija;
max = 135*u5+ 135*u6- 90*u1- 180*u2;

!Omejitve;
-u1+u3<=10;
-u1+u4<=7;
-u1+u5<=23;
-u1+u6<=26;
-u2+u3<=12;
-u2+u4<=9;
-u2+u5<=21;
-u2+u6<=28;
-u3+u4<=14;
-u3+u5<=21;
-u3+u6<=23;
-u4+u5<=7;
-u4+u6<=12;
```

V prvi vrstici je zapisana namenska funkcija ter v naslednjih vrsticah vse omejitve danega problema.

Poročilo o rešitvi dualnega programa:

```
Global optimal solution found.
  Objective value:                              4815.000
  Infeasibilities:                              0.000000
  Total solver iterations:                             4
  Elapsed runtime seconds:                          0.04

  Model Class:                                        LP


                                Variable           Value        Reduced Cost
                                      U5        16.00000            0.000000
                                      U6        21.00000            0.000000
                                      U1        2.000000            0.000000
                                      U2        0.000000            0.000000
                                      U3        0.000000            0.000000
                                      U4        9.000000            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1        4815.000            1.000000
                                       2        12.00000            0.000000
                                       3        0.000000            90.00000
                                       4        9.000000            0.000000
                                       5        7.000000            0.000000
                                       6        12.00000            0.000000
                                       7        0.000000            180.0000
                                       8        5.000000            0.000000
                                       9        7.000000            0.000000
                                      10        5.000000            0.000000
                                      11        5.000000            0.000000
                                      12        2.000000            0.000000
                                      13        0.000000            135.0000
                                      14        0.000000            135.0000
```

**Vrednost namenske funkcije:** je enaka kot vrednost namenske funkcije primarnega linearnega programa. Senčne cene posamične omejitve, ki so nastopale v osnovnem linearnem programu so enake vrednostim odločitvenih spremenljivk dualnega programa $u_1$, $u_2$,$u_3$ in $u_4$.

Senčna cena predstavlja spremembo vrednosti optimalne namenske funkcije ob spremembi 1 enote resursa, ki je v danem primeru količina prepeljanih enot blaga.

Če pogledamo na sistem kot na celoto vidimo, da se senčna cena v zaporednih vozliščih povečuje točno za višino transportnih stroškov na enoto blaga po tej poti. Pri interpretaciji rezultatov dualnega problema vidimo, da se optimum namenske funkcije, ob povečanju pretoka blaga skozi sistem za 1 enoto, spremeni točno za razliko senčne cene končnega vozlišča in senčne cene začetnega vozlišča, skozi katerega je potovala ta dodatna enota blaga, kar pa je ekvivalentno skupnim stroškom na enoto blaga po tej isti poti.

Če bi se povečala količina vhodnega blaga na vozlišču 1 za 1 enoto, in obenem količina zahtevanega blaga na vozlišču 5 za eno enoto, bi se prvotna optimalna vrednost namenske funkcije povišala za $u5 - u1$, kar je $18\,€/\mathrm{enoto}$ blaga – $4\,€/\mathrm{enoto}$ blaga kar je enako $14\,€/\mathrm{enoto}$ blaga. Skupni stroški in vrednost namenske funkcije bi se povišali z 4.815 na 4.829 €.

Skupni stroški transporta blaga po danem omrežju bi se povišali za 14 €, če bi transportirali dodatno enoto skozi omrežje preko vozlišča 1 in 5, medtem ko ostane količina prepeljanega blaga iz vozlišča 2 do vozlišča 5 in 6 nespremenjena.

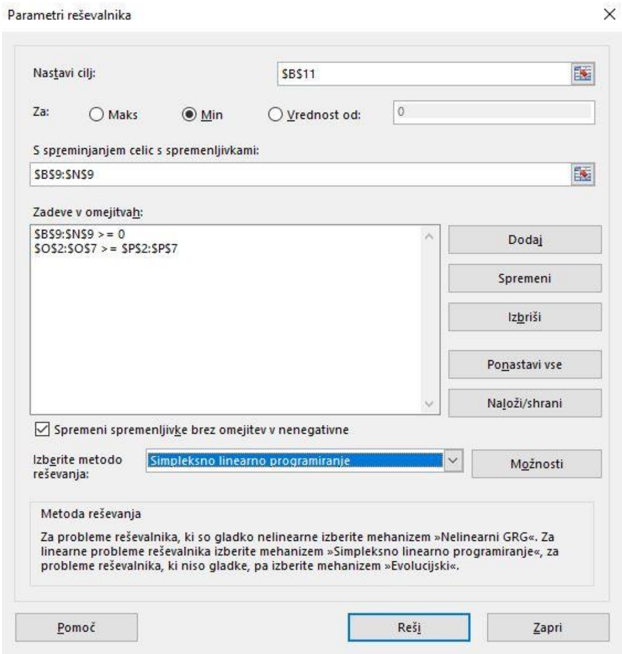
### 1.3.4 Reševanje modela vozlišča in lokalnih povezav s pomočjo programa Excel

Poglejmo še reševanje primera modela vozlišča in lokalnih povezav s programskim orodjem Excel. Pripravimo incidenčno matriko problema. Pripravimo enačbe za omejitve toka skozi vozlišča, definiramo namensko funkcijo problema in pripravimo tabelo za vpis rešitve problema.

Slika 10: Podatki modela v orodju Excel

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Vozlišče/Povezava | x13 | x14 | x15 | x16 | x23 | x24 | x25 | x26 | x34 | x35 | x36 | x45 | x46 | | b |
| 2 | 1 | -1 | -1 | -1 | -1 | | | | | | | | | | =SUMPRODUCT(B2:N2;$B$9:$N$9) | -90 |
| 3 | 2 | | | | | -1 | -1 | -1 | -1 | | | | | | =SUMPRODUCT(B3:N3;$B$9:$N$9) | -180 |
| 4 | 3 | 1 | | | | 1 | | | | -1 | -1 | -1 | | | =SUMPRODUCT(B4:N4;$B$9:$N$9) | 0 |
| 5 | 4 | | 1 | | | | 1 | | | 1 | | | -1 | -1 | =SUMPRODUCT(B5:N5;$B$9:$N$9) | 0 |
| 6 | 5 | | | 1 | | | | 1 | | | 1 | | 1 | | =SUMPRODUCT(B6:N6;$B$9:$N$9) | 135 |
| 7 | 6 | | | | 1 | | | | 1 | | | 1 | | 1 | =SUMPRODUCT(B7:N7;$B$9:$N$9) | 135 |
| 8 | Cena enote blaga | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | | |
| 9 | Vrednost spremenljivke | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | |
| 11 | Vrednost namenske funkcije | =SUMPRODUCT(B8:N8;B9:N9) | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | |

Slika 11: Vnos parametrov v Reševalnik




Kot cilj nastavimo celico B11, ki predstavlja namensko funkcijo in označimo, da želimo poiskati minimum te funkcije ob pogojih, ki so definirani v celicah O2:O7 in P2:P7. Pogoje definiramo v »zadeve v omejitvah«. Kot levo stran omejitev označimo obseg O2:O7, kjer vsaka celica predstavlja seštevek vhodnih in izhodnih povezav vsakega vozlišča. Na primer, v celici O2 je zapisana vrednost, ki je ekvivalentna matematičnemu zapisu $-x_{13} - x_{14} - x_{15} - x_{16}$. Nato označimo ustrezen neenačaj, ki je v tem primeru ≥ ter dodamo desno stran omejitvenih neenačb, ki jo preberemo iz obsega P2:P7. Tako se npr. omejitev za vozlišče 1, glasi: $-x_{13} - x_{14} - x_{15} - x_{16} \geq -90$. V reševalniku izberemo metodo reševanja, simpleksno linearno programiranje.

Slika 12: Rešitev Hub and spoke modela v Excelu

| | A | | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Vozlišče/Povezava | | x13 | x14 | x15 | x16 | x23 | x24 | x25 | x26 | x34 | x35 | x36 | x45 | x46 | | b |
| 2 | | 1 | -1 | -1 | -1 | -1 | | | | | | | | | | -90 | **-90** |
| 3 | | 2 | | | | | -1 | -1 | -1 | -1 | | | | | | -180 | **-180** |
| 4 | | 3 | 1 | | | | 1 | | | | -1 | -1 | -1 | | | 0 | **0** |
| 5 | | 4 | | 1 | | | | 1 | | | 1 | | | -1 | -1 | 0 | **0** |
| 6 | | 5 | | | 1 | | | | 1 | | | 1 | | 1 | | 135 | **135** |
| 7 | | 6 | | | | 1 | | | | 1 | | | 1 | | 1 | 135 | **135** |
| 8 | Cena enote blaga | | **10** | **7** | **23** | **26** | **12** | **9** | **21** | **28** | **14** | **21** | **23** | **7** | **12** | | |
| 9 | Vrednost spremenljivke | | 0 | 90 | 0 | 0 | 0 | 180 | 0 | 0 | 0 | 0 | 0 | 135 | 135 | | |
| 10 | | | | | | | | | | | | | | | | | |
| 11 | Vrednost namenske funkcije | | | | 4815 | | | | | | | | | | | | |

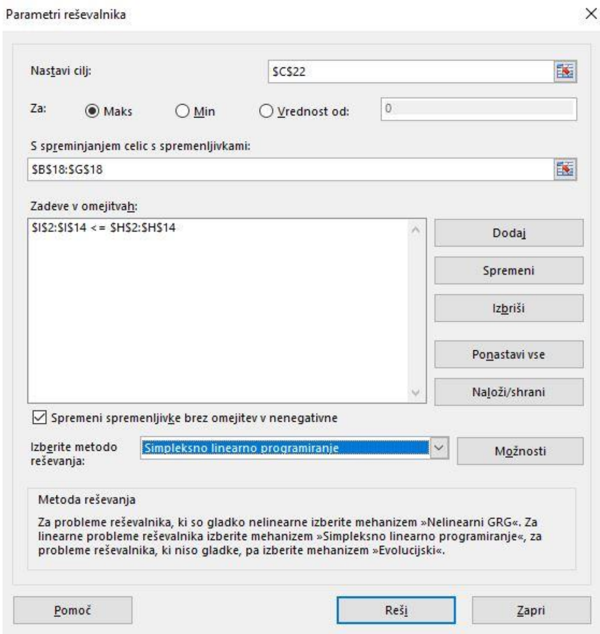
Vrednosti odločitvenih spremenljivk so izpisane v rumenem razdelku, vrednost namenske funkcije ob takšnih vrednostih pa v celici D11. V vozlišče 4 se iz vozlišča 1 transportira 90 enot materiala in iz vozlišča 2 180 enot materiala. Iz vozlišča 4 se v vozlišče 5 transportira 135 enot materiala in v vozlišče 6 135 enot materiala. Ob tem bodo transportni stroški 4.815 €.

Poglejmo še reševanje dualnega programa modela vozlišča in lokalnih povezav v Excelu. Uvedemo dualne spremenljivke za vsako vozlišče. Omejitve zapišemo po načelu izčrpavanja vrednosti, kar pomeni, da je dualna cena naslednjega vozlišča manjša ali enaka seštevku dualne cene predhodnega vozlišča in transportnih stroškov enote transportiranega blaga po tej povezavi. V tem primeru želimo maksimirati senčno vrednost transportiranega materiala znotraj omrežja.

Slika 13: Dualni program modela vozlišča in lokalnih povezav s programskim orodjem Excel

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | u1 | u2 | u3 | u4 | u5 | u6 | Cena | Omejitve |
| 2 | x13 | -1 | | 1 | | | | **10** | =SUMPRODUCT(B2:G2;$B$18:$G$18) |
| 3 | x14 | -1 | | | 1 | | | **7** | =SUMPRODUCT(B3:G3;$B$18:$G$18) |
| 4 | x15 | -1 | | | | 1 | | **23** | =SUMPRODUCT(B4:G4;$B$18:$G$18) |
| 5 | x16 | -1 | | | | | 1 | **26** | =SUMPRODUCT(B5:G5;$B$18:$G$18) |
| 6 | x23 | | -1 | 1 | | | | **12** | =SUMPRODUCT(B6:G6;$B$18:$G$18) |
| 7 | x24 | | -1 | | 1 | | | **9** | =SUMPRODUCT(B7:G7;$B$18:$G$18) |
| 8 | x25 | | -1 | | | 1 | | **21** | =SUMPRODUCT(B8:G8;$B$18:$G$18) |
| 9 | x26 | | -1 | | | | 1 | **28** | =SUMPRODUCT(B9:G9;$B$18:$G$18) |
| 10 | x34 | | | -1 | 1 | | | **14** | =SUMPRODUCT(B10:G10;$B$18:$G$18) |
| 11 | x35 | | | -1 | | 1 | | **21** | =SUMPRODUCT(B11:G11;$B$18:$G$18) |
| 12 | x36 | | | -1 | | | 1 | **23** | =SUMPRODUCT(B12:G12;$B$18:$G$18) |
| 13 | x45 | | | | -1 | 1 | | **7** | =SUMPRODUCT(B13:G13;$B$18:$G$18) |
| 14 | x46 | | | | -1 | | 1 | **12** | =SUMPRODUCT(B14:G14;$B$18:$G$18) |
| 15 | b | **-90** | **-180** | **0** | **0** | **135** | **135** | | |
| 16 | | | | | | | | | |
| 17 | | u1 | u2 | u3 | u4 | u5 | u6 | | |
| 18 | Vrednosti odločitvenih spremenljivk | | | | | | | | |
| 19 | | | | | | | | | |
| 20 | | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | Vrednost namenske funkcije | =SUMPRODUCT(B15:G15;B18:G18) | | | | | | | |

Slika 14: Vnos parametrov v Reševalnik




Rešitev so vrednosti dualnih spremenljivk in vrednost namenske funkcije, ki je enaka kot vrednost namenske funkcije primarnega linearnega programa.

Slika 15: Rešitev dualnega linearnega programa modela vozlišča in lokalnih povezav

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | u1 | u2 | u3 | u4 | u5 | u6 | Cena | Omejitve |
| 2 | x13 | -1 | | 1 | | | | **10** | -2 |
| 3 | x14 | -1 | | | 1 | | | **7** | 7 |
| 4 | x15 | -1 | | | | 1 | | **23** | 14 |
| 5 | x16 | -1 | | | | | 1 | **26** | 19 |
| 6 | x23 | | -1 | 1 | | | | **12** | 0 |
| 7 | x24 | | -1 | | 1 | | | **9** | 9 |
| 8 | x25 | | -1 | | | 1 | | **21** | 16 |
| 9 | x26 | | -1 | | | | 1 | **28** | 21 |
| 10 | x34 | | | -1 | 1 | | | **14** | 9 |
| 11 | x35 | | | -1 | | 1 | | **21** | 16 |
| 12 | x36 | | | -1 | | | 1 | **23** | 21 |
| 13 | x45 | | | | -1 | 1 | | **7** | 7 |
| 14 | x46 | | | | -1 | | 1 | **12** | 12 |
| 15 | b | **-90** | **-180** | **0** | **0** | **135** | **135** | | |
| 16 | | | | | | | | | |
| 17 | | u1 | u2 | u3 | u4 | u5 | u6 | | |
| 18 | Vrednosti odločitvenih spremenljivk | 2 | 0 | 0 | 9 | 16 | 21 | | |
| 19 | | | | | | | | | |
| 20 | | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | Vrednost namenske funkcije | | 4815 | | | | | | |
| 23 | | | | | | | | | |

Maksimalna senčna vrednost blaga v omrežju je enaka 4.815 €, kar je enako stroškom transporta danih enot preko omrežja.

## 1.4 Problem najkrajše poti[19]

Na zelo podoben način rešujemo tudi probleme najkrajše poti. V tem primeru želimo poiskati najkrajšo pot od vozlišča $s$ do vozlišča $t$. V ta namen definiramo odločitveno spremenljivko, ki ravno tako predstavlja količino blaga, ki ga transportiramo po povezavi:

$$x_{ij} = tok\ od\ vozlišča\ i\ do\ vozlišča\ j$$

Ker želimo poiskati najkrajšo pot med vozlišči v omrežju, potrebujemo funkcijo razdalje. To zapišemo kot vsoto produktov dolžine poti ($c_{ij}$) in količine blaga, ki bo prepeljana po poti $c_{ij} \cdot x_{ij}$:

$$\min f(x_{ij}) = \sum_{za\ vse\ poti\ (i,j)} c_{ij} x_{ij}$$

Pri problemu najkrajše poti predpostavimo, da pošljemo skozi omrežje le 1 enoto materiala ali storitve. Predpostavimo, da imamo na izvornem vozlišču (ponudba) -1 enoto in na izhodnem vozlišču (povpraševanje) +1 enoto, vsa ostala pretovorna vozlišča v omrežju pa nimajo niti povpraševanja niti ponudbe (povpraševanje = 0; ponudba = 0).

Odločitvene spremenljivke lahko v optimalni rešitvi zavzamejo le binarni vrednosti (0 ali 1), za kar je potrebno uvesti dodatne omejitve, ki bodo zagotovile binarnost odločitvenih spremenljivk.

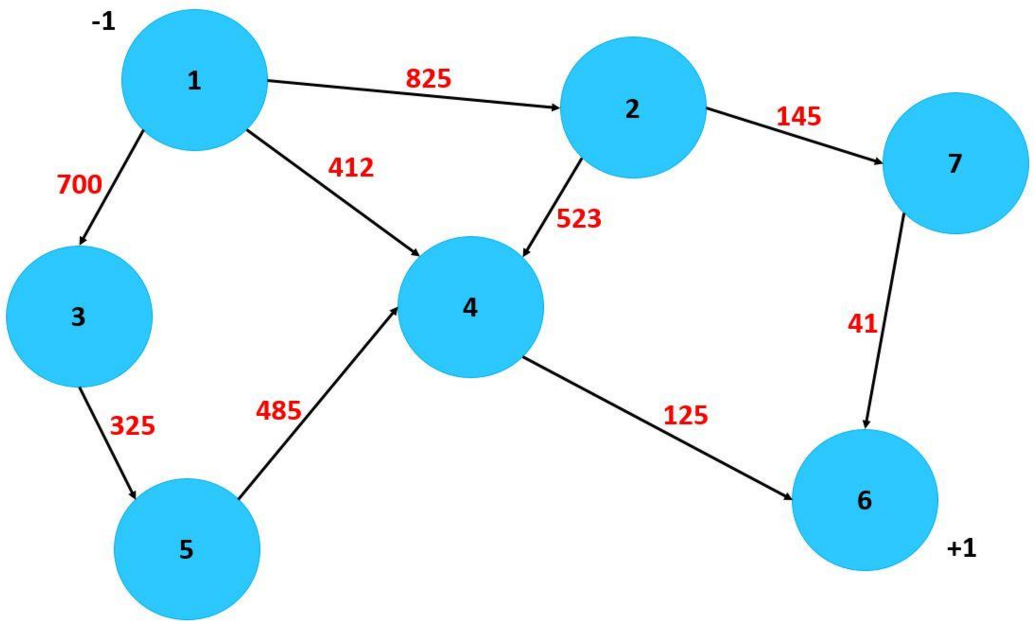
**Primer 2:**

Oseba planira potovanje z letalom. Potovanje bo začela na letališču, ki ga predstavlja vozlišče 1, njen končni cilj pa je letališče, ki ga predstavlja vozlišče 6. Letališče ne omogoča direktnega leta, temveč z različnimi prestopi. Preden

[19] Shortest path problem

rezervira let želi ugotoviti, kateri let bi predstavljal najkrajšo pot. Na sliki 16 so predstavljena letališča in razdalje med njimi.

Slika 16: Problem najkrajše poti




Oseba v vozlišču 1 predstavlja ponudbo, v vozlišču 6 pa predstavlja povpraševanje. Vozlišče 1 je tako izvor, kjer je ponudba prirejena na 1 enoto. Vozlišče 6 je ponor s povpraševanjem po 1 enoti. Ker imajo vsa ostala pretovorna vozlišča ponudbo in povpraševanje enako 0, bo zagotovo ta 1 enota iz vozlišča 1 pripotovala do vozlišča 6. Želimo minimirati prepotovano pot.

**Odločitvene spremenljivke:**

$$x_{ij} = \check{s}tevilo\ prepotovanih\ enot\ po\ povezavah$$

V primeru problema najkrajše poti lahko odločitvena spremenljivka $x_{ij}$ zavzame binarne vrednosti (0, 1), saj je samo 1 enota nekega materiala na vhodu v omrežje. Tista odločitvena spremenljivka, ki predstavlja pot, po kateri se bo ta enota transportirala, bo zavzela vrednost 1, ostale odločitvene spremenljivke, ki predstavljajo pot, po kateri enota materiala ne bo potovala, zavzamejo vrednost 0.

Na osnovi omrežja na sliki najprej zapišimo incidenčno matriko, ki bo osnova za linearni program.

| | $x_{12}$ | $x_{13}$ | $x_{14}$ | $x_{24}$ | $x_{27}$ | $x_{35}$ | $x_{46}$ | $x_{54}$ | $x_{67}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | | | | | | | -1 |
| 2 | 1 | | | -1 | -1 | | | | | 0 |
| 3 | | 1 | | | | -1 | | | | 0 |
| 4 | | | 1 | 1 | | | -1 | 1 | | 0 |
| 5 | | | | | | 1 | | -1 | | 0 |
| 6 | | | | | | | 1 | | -1 | 0 |
| 7 | | | | | 1 | | | | 1 | 1 |
| $d$ | 825 | 700 | 412 | 523 | 145 | 325 | 125 | 485 | 41 | |

Ker želimo poiskati najkrajšo pot med vozlišči v omrežju, bomo minimirali funkcijo razdalje.

**Minimum namenske funkcije:**

$$\min f(x_{ij}) = 700x_{13} + 412x_{14} + 825x_{12} + 325x_{35} + 485x_{54} + 523x_{42} + 145x_{27} + 125x_{46} + 41x_{76}$$

**Omejitve:**

Omejitve ravnovesja pretoka v vozliščih so sledeče:

Vozlišče 1: $-x_{13} - x_{14} - x_{12} = -1$

Vozlišče 2: $x_{12} + x_{42} - x_{27} = 0$

Vozlišče 3: $x_{13} - x_{35} = 0$

Vozlišče 4: $x_{14} + x_{54} - x_{42} - x_{46} = 0$

Vozlišče 5: $x_{35} - x_{54} = 0$

Vozlišče 6: $x_{46} + x_{76} = 1$

Vozlišče 7: $x_{27} - x_{76} = 0$

Ter pogoj za birnarnost števil: $x_{ij} \epsilon \{0,1\}$

#### 1.4.1.1 Reševanje problema najkrajše poti s programom Excel

Priprava podatkov in zapis incidenčne matrike v Excelu:

Slika 17: Priprava podatkov v Excelu

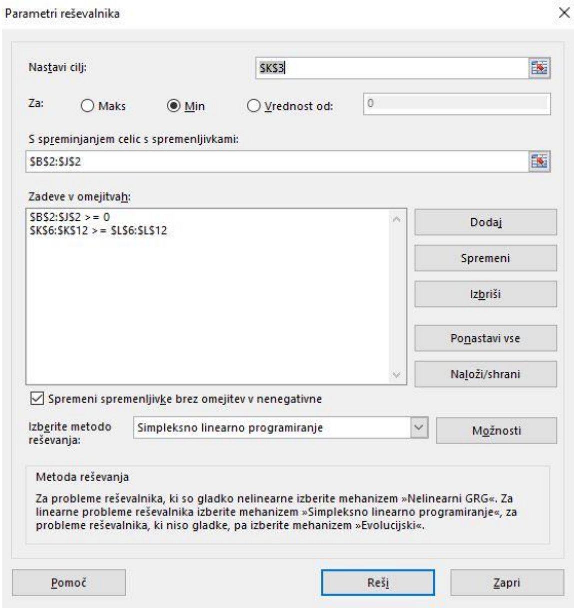
| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Povezave | x13 | x14 | x12 | x35 | x54 | x42 | x27 | x46 | x76 | | |
| 2 | Vrednost spremenljivk | | | | | | | | | | Minimalna razdalja | |
| 3 | Razdalja | 700 | 412 | 825 | 325 | 485 | 523 | 145 | 125 | 41 | =SUMPRODUCT(B2:J2;B3:J3) | |
| 4 | | | | | | | | | | | | |
| 5 | Incidenčna matrika | | | | | | | | | | | b |
| 6 | Vozlišče 1 | -1 | -1 | -1 | | | | | | | =SUMPRODUCT(B6:J6;$B$2:$J$2) | -1 |
| 7 | Vozlišče 2 | | | 1 | | | 1 | -1 | | | =SUMPRODUCT(B7:J7;$B$2:$J$2) | 0 |
| 8 | Vozlišče 3 | 1 | | | -1 | | | | | | =SUMPRODUCT(B8:J8;$B$2:$J$2) | 0 |
| 9 | Vozlišče 4 | | 1 | | | 1 | -1 | | -1 | | =SUMPRODUCT(B9:J9;$B$2:$J$2) | 0 |
| 10 | Vozlišče 5 | | | | 1 | -1 | | | | | =SUMPRODUCT(B10:J10;$B$2:$J$2) | 0 |
| 11 | Vozlišče 6 | | | | | | | | 1 | 1 | =SUMPRODUCT(B11:J11;$B$2:$J$2) | 1 |
| 12 | Vozlišče 7 | | | | | | | 1 | | -1 | =SUMPRODUCT(B12:J12;$B$2:$J$2) | 0 |
| 13 | | | | | | | | | | | | |

V celicah v obsegu B3:J3 so zapisane razdalje med vozlišči. V celicah v obsegu B2:J2 so navedene odločitvene spremenljivke. Namenska funkcija je zapisana v celici K3 in predstavlja vsoto produktov razdalj med vozlišči in korespondenčnih spremenljivk na tej povezavi. V celicah v obsegu L6:L12 so zapisane količine v povpraševanju in ponudbi in predstavljajo desne strani omejitvenih neenačb.

Celice v obsegu K6:K12 so leve strani omejitvenih neenačb. Polja v obsegu B6:J12 predstavljajo incidenčno matriko.

V vrstici 2 je pripravljena tabela za izpis vrednosti odločitvenih spremenljivk. Te lahko zavzamejo vrednost samo 0 ali 1. V primeru, da odločitvena spremenljivka zavzame vrednost 1, bo ta povezava del rešitve problema, v nasprotnem primeru pa povezava ne bo del rešitve problema.

Slika 18: Parametri Reševalnika




Kot cilj vnesemo celico K3, v kateri je zapisana namenska funkcija. Želimo poiskati minimum namenske funkcije.

V naslednje prosto mesto vnesemo obseg celic, kjer so zapisane odločitvene spremenljivke ter dodamo omejitve.

Slika 19: Rešitev problema najkrajše poti

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Povezave | x13 | x14 | x12 | x35 | x54 | x42 | x27 | x46 | x76 | |
| 2 | Vrednost spremenljivk | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Minimalna razdalja |
| 3 | Razdalja | 700 | 412 | 825 | 325 | 485 | 523 | 145 | 125 | 41 | 537 |

Najkrajša pot od vozlišča 1 do vozlišča 6 je preko vozlišča 4 in znaša 537 enot. V rešitvi se pojavita odločitveni spremenljivki $x_{14}$ in $x_{16}$. Obe zavzameta vrednost 1, medtem ko so vse ostale odločitvene spremenljivke problema enake 0. Če to rešitev vstavimo v namensko funkcijo, dobimo:

$$f(x_{ij}) = 700 \cdot 0 + 412 \cdot 1 + 825 \cdot 0 + 325 \cdot 0 + 485 \cdot 0 + 523 \cdot 0 + 145 \cdot 0 + 125 \cdot 1 + 41 \cdot 0 = 537$$

Najkrajša pot v omrežju od vozlišča 1 do vozlišča 6 znaša 537 enot.

#### 1.4.1.2 Reševanje problema najkrajše poti s programom LINGO

Prikažimo še rešitev problema najkrajše poti s programom LINGO:

```
!Namenska funkcija;
min=700*x13+412*x14+825*x12+325*x35+485*x54+523*x42+145*x2
7+125*x46+41*x76;
!Omejitve;
[Vozlisce_1] -x13 - x14 - x12 = -1;
[Vozlisce_2] x12 + x42 - x27 = 0;
[Vozlisce_3] x13 - x35 = 0;
[Vozlisce_4] x14 + x54 - x42 - x46 = 0;
[Vozlisce_5] x35 - x54 = 0;
[Vozlisce_6] x46 + x76 = 1;
[Vozlisce_7] x27 - x76 = 0;
```

Zapisali smo namensko funkcijo, ki jo želimo minimirati ob pogojih ravnovesja pretoka skozi vsako vozlišče. Pretok je omejen z vrednostjo 0 ali 1.

Poročilo o rešitvi problema najkrajše poti:

```
Global optimal solution found.
  Objective value:                              537.0000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.03

  Model Class:                                        LP

                            Variable           Value        Reduced Cost
                                 X13        0.000000            0.000000
                                 X14        1.000000            0.000000
                                 X12        0.000000            474.0000
                                 X35        0.000000            0.000000
                                 X54        0.000000            1098.000
                                 X42        0.000000            584.0000
                                 X27        0.000000            0.000000
                                 X46        1.000000            0.000000
                                 X76        0.000000            0.000000

                                 Row    Slack or Surplus      Dual Price
                                   1        537.0000           -1.000000
                         VOZLISCE_1        0.000000            0.000000
                         VOZLISCE_2        0.000000           -351.0000
                         VOZLISCE_3        0.000000           -700.0000
                         VOZLISCE_4        0.000000           -412.0000
                         VOZLISCE_5        0.000000           -1025.000
                         VOZLISCE_6        0.000000           -537.0000
                         VOZLISCE_7        0.000000           -496.0000
```

Rešitev je enaka kot pri Excelu, najkrajša pot od vozlišča 1 do vozlišča 6 je preko vozišča 4 in znaša 537 enot.

## 1.5 Problem pretovora

Problem pretovora lahko razumemo kot več fazni transportni problem, kjer je tok materiala prekinjen vsaj enkrat na poti od izvorne točke do ponorne točke.

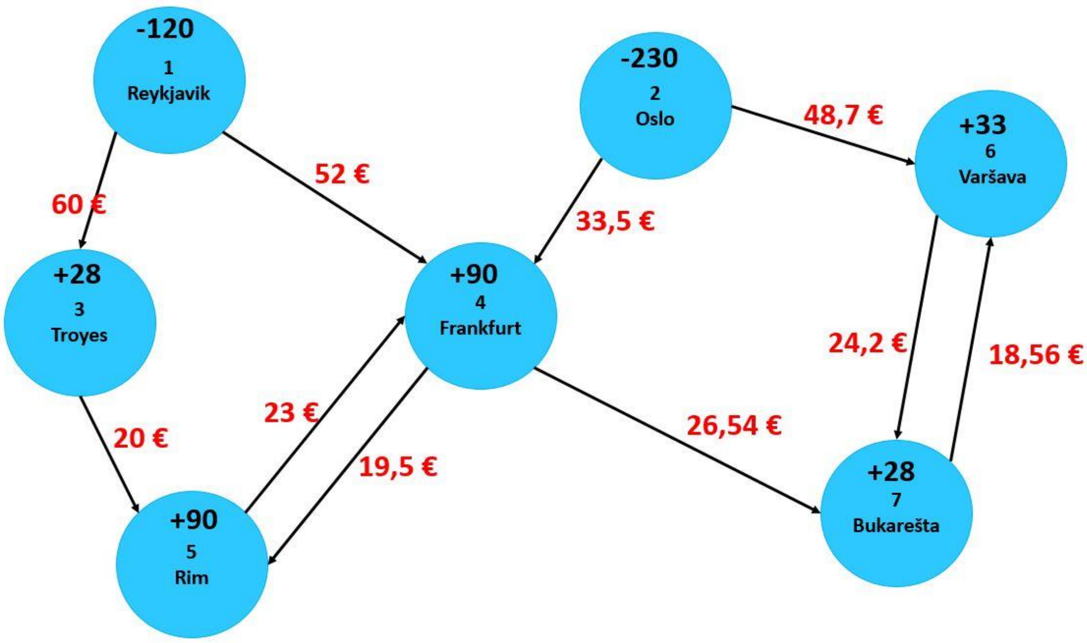
Velikokrat se srečamo s konkretnimi problemi, ki zahtevajo, da se material ali tovor transportira od ponudnika do končne točke preko ene ali več vmesnih postaj, tako da se doseže minimalne možne stroške. Problemi s takšnimi karakteristikami predstavljajo problem pretovora.

**<u>Primer 3:</u>**

PRIMER PODJETJA IKEA

Podjetje Ikea želi dostaviti »Spring – bed« postelje iz Reykjavika in Osla v ostala mesta, podana v spodnjem omrežju, po najnižjih stroških. Kapacitete poti, povpraševanje in ponudba ter cena prevoza 1 postelje so podane na sliki 20.

Slika 20: Model problema pretovora primer podjetja Ikea



Z rdečo barvo so podane cene 1 transportirane enote. Vsako vozlišče ima prav tako podane količine povpraševanja in ponudbe. Mesti Reykjavik in Oslo sta izvorni vozlišči, od koder se postelje transportirajo v ostala mesta. Reykjavik ima v ponudbi 120 postelj, Oslo pa 230 postelj. Povpraševanja ostalih mest, v katera je potrebno transportirati postelje, so predstavljena na sliki 128.

Pri problemu pretovora vedno nastopajo identični produkti. Vse postelje, ki jih podjetje razvaža so enake. Tok in odločitvena spremenljivka sta torej izražena v enotah istega produkta.

**Odločitvene spremenljivke:**

$x_{ij} =$

$\check{s}tevilo\ poslanih\ postelj\ od\ mesta\ i\ do\ mesta\ j\ za\ vsak\ par\ medsebojno\ povezanih\ mest$

Na podlagi znanih podatkov sestavimo incidenčno matriko.

| | $x_{13}$ | $x_{14}$ | $x_{24}$ | $x_{26}$ | $x_{35}$ | $x_{45}$ | $x_{47}$ | $x_{54}$ | $x_{67}$ | $x_{76}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | | | | | | | | | -120 |
| 2 | | | -1 | -1 | | | | | | | -230 |
| 3 | 1 | | | | -1 | | | | | | 28 |
| 4 | | 1 | 1 | | | -1 | -1 | 1 | | | 90 |
| 5 | | | | | 1 | 1 | | -1 | | | 90 |
| 6 | | | | 1 | | | | | -1 | 1 | 33 |
| 7 | | | | | | | 1 | | 1 | -1 | 28 |
| $c_{ij}$ | 60 | 52 | 33,5 | 48,7 | 20 | 19,5 | 26,54 | 23 | 24,2 | 18,56 | |

Pri zapisu omejitev linearnega programa najprej poglejmo skupno vsoto postelj, ki so na voljo in kolikšno je skupno povpraševanje ter ponudba v omrežju. Ponudba je v danem primeru višja od povpraševanja $(350 > 269)$. Velja torej izraz $[\sum vhodnih\ tokov\ v\ vozli\check{s}\check{c}e\ i] - [\sum izhodnih\ tokov\ iz\ vozli\check{s}\check{c}a\ i] \geq b_i$, ki nam bo v pomoč pri pisanju omejitev.

**Minimum namenske funkcije:**

$$\min f(x_{ij}) = 60x_{13} + 52x_{14} + 33{,}5x_{24} + 48{,}7x_{26} + 20x_{35} + 19{,}5x_{45} + +26{,}54x_{47} + 23x_{54} + 24{,}2x_{67} + 18{,}56x_{76}$$

**Omejitve:**

Omejitve, ki se nanašajo na ravnovesni tok vsakega vozlišča, ob upoštevanju povpraševanja in ponudbe, lahko zapišemo v dveh različnih oblikah.

Vozlišče 1:

$$x_{13} + x_{14} \leq 120$$

Količina postelj, ki jih podjetje izvozi iz vozlišča 1, je lahko manjša ali kvečjemu enaka količini, ki jo imajo na razpolago v vozlišču 1. To omejitev lahko zapišemo tudi v drugi obliki. Ker je skupna ponudba v omrežju večja od povpraševanja to naredimo v obliki

$[\sum vhodnih\ tokov\ v\ vozlišče\ i] - [\sum izhodnih\ tokov\ iz\ vozlišča\ i] \geq b_i$:

$$0 - x_{13} - x_{14} \geq -120$$

kjer 0 predstavlja tok v vozlišče 1 - $x_{01}$ (saj v vozlišče 1 ne transportiramo postelj) in spremenljivko v zapisu lahko opustimo, ostali dve spremenljivki pa predstavljata tok iz vozlišča 1. Ponudba v vozlišču 1 je enaka 120 enotam. V nadaljevanju zapišemo omejitve za vsako vozlišče iz modela in sicer na način, predstavljen v drugi obliki.

| | |
|---|---|
| Vozlišče 2: | $0 - x_{24} - x_{26} \geq -230$ |
| Vozlišče 3: | $x_{13} - x_{35} \geq 28$ |
| Vozlišče 4: | $x_{14} + x_{24} + x_{54} - x_{45} - x_{47} \geq 90$ |
| Vozlišče 5: | $x_{35} + x_{45} - x_{54} \geq 90$ |
| Vozlišče 6: | $x_{26} + x_{76} - x_{67} \geq 33$ |
| Vozlišče 7: | $x_{47} + x_{67} - x_{76} \geq 28$ |

Ter še pogoj nenegativnosti odločitvenih spremenljivk:

$$x_{ij} \geq 0; i = 1, 2, 3, 4, 5, 6, 7 \text{ in } j = 2, 3, 4, 5, 6, 7$$

Tako smo zastavljen problem prikazali kot problem linearnega programa, ki ga lahko rešimo s programskim orodjem Excel ali LINGO.

### 1.5.1 Reševanje problema pretovora s programom Excel

Zapišemo podatke v primerni obliki na prazen delovni list v Excelu:

Slika 21: Priprava podatkov

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Povezave | x13 | x14 | x35 | x54 | x45 | x24 | x47 | x26 | x67 | x76 | | |
| 2 | Število prepeljanih postelj | | | | | | | | | | | Skupna cena | |
| 3 | Cena [€] | 60 | 52 | 20 | 23 | 19,5 | 33,5 | 26,54 | 48,7 | 24,2 | 18,56 | =SUMPRODUCT(B2:K2;B3:K3) | |
| 4 | | | | | | | | | | | | | |
| 5 | Incidenčna matrika | | | | | | | | | | | | b |
| 6 | Vozlišče 1 | -1 | -1 | | | | | | | | | =SUMPRODUCT(B6:K6;$B$2:$K$2) | -120 |
| 7 | Vozlišče 2 | | | | | | -1 | | -1 | | | =SUMPRODUCT(B7:K7;$B$2:$K$2) | -230 |
| 8 | Vozlišče 3 | 1 | | -1 | | | | | | | | =SUMPRODUCT(B8:K8;$B$2:$K$2) | 28 |
| 9 | Vozlišče 4 | | 1 | | 1 | -1 | 1 | -1 | | | | =SUMPRODUCT(B9:K9;$B$2:$K$2) | 90 |
| 10 | Vozlišče 5 | | | 1 | -1 | 1 | | | | | | =SUMPRODUCT(B10:K10;$B$2:$K$2) | 90 |
| 11 | Vozlišče 6 | | | | | | | | 1 | -1 | 1 | =SUMPRODUCT(B11:K11;$B$2:$K$2) | 33 |
| 12 | Vozlišče 7 | | | | | | | 1 | | 1 | -1 | =SUMPRODUCT(B12:K12;$B$2:$K$2) | 28 |
| 13 | | | | | | | | | | | | | |

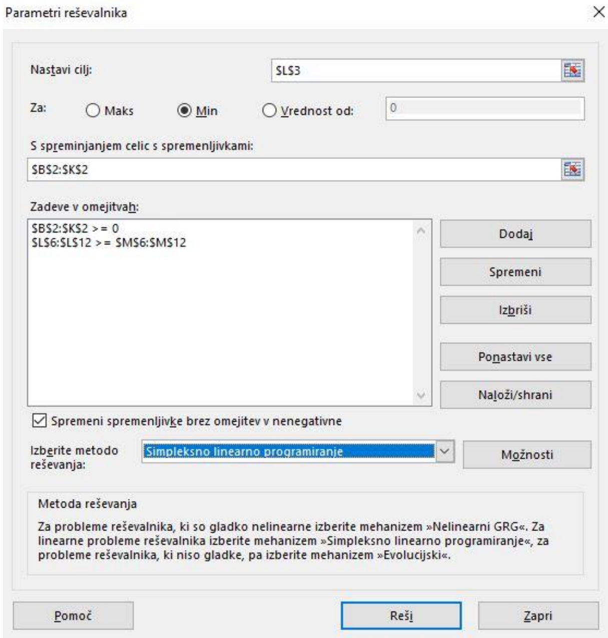
V prvo vrstico zapišemo vse odločitvene spremenljivke, v vrstico 3 zapišemo cene transporta po povezavah med mesti. Polja za število prepeljanih postelj so prazna in jih bo zapolnil program z rešitvijo vrednosti odločitvenih spremenljivk, tako da bo končna skupna cena transporta najnižja.

V celici L3 je zapisana namenska funkcija. V ta namen smo uporabili funkcijo *SUMPRODUCT*, ki jo omogoča Excel.

V celicah L6:L12 in M6:M12 so zapisane omejitve sistema. Tudi tukaj uporabimo funkcijo SUMPRODUCT.

Tako pripravljene podatke vnesemo v Reševalnik:

Slika 22: Parametri reševalnika




Ko uredimo vse parametre v Excelovem reševalniku, program poda rešitev.

Slika 23: Rešitev problema pretovora – primer podjetja Ikea

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Povezave | x13 | x14 | x35 | x54 | x45 | x24 | x47 | x26 | x67 | x76 | |
| 2 | Število prepel | 28 | 11 | 0 | 0 | 90 | 197 | 28 | 33 | 0 | 0 | Skupna cena |
| 3 | Cena [€] | 60 | 52 | 20 | 23 | 19,5 | 33,5 | 27 | 48,7 | 24,2 | 18,6 | 12956,72 |

Za zagotovitev najnižjih stroškov transporta postelj pod pogojem, da zadostimo povpraševanju in ponudbi v določenih mestih, je optimalno, da Ikea transportira postelje na sledeč način:

- 28 postelj iz vozlišča 1 v vozlišče 3 (Reykjavik-Troyes);
- 11 postelj iz Reykjavika v Frankfurt;
- 90 postelj iz Frankfurta v Rim;
- 197 iz Osla v Frankfurt, 28 iz Frankfurta v Bukareštom;
- ter 33 postelj iz Osla v Varšavo.

Ob takšnem transportnem načrtu bodo skupni stroški prevoza postelj znašali 12.956,72 €.

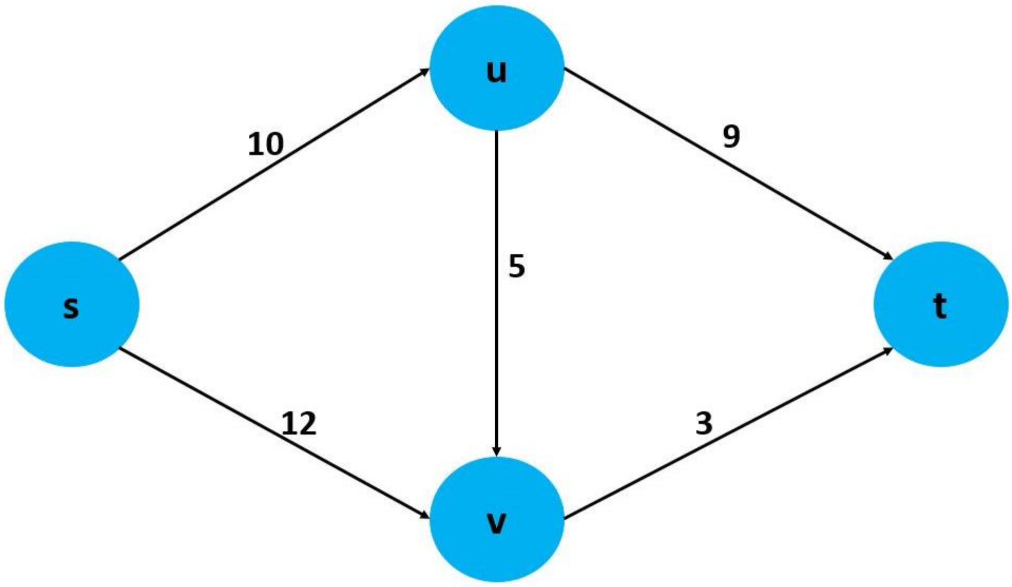
## 1.6 Problem maksimalnega pretoka

Do sedaj smo obravnavali primere, kjer smo iskali minimum namenske funkcije (najkrajši čas, najcenejša pot, najkrajša pot ipd.). V tem poglavju pa bomo pogledali primer s področja omrežnih pretokov, kjer namensko funkcijo maksimiramo.

Problem maksimalnega pretoka se po navadi pojavi tam, kjer obstajajo določene kapacitete oz. zmogljivosti ali omejitve na povezavah med vozlišči. Pri takšnih omrežnih pogojih želimo maksimirati pretok po teh povezavah, skladno z omejitvami povezav. Za reševanje teh problemov obstaja nabor algoritmov, vendar bomo v tem gradivu problem maksimalnega pretoka pretvorili na problem linearnega programiranja (omenimo le enega izmed algoritmov za reševanje tovrstnih problemov, ki je bil razvit leta 1962. To je Ford – Fulkersonov algoritem).

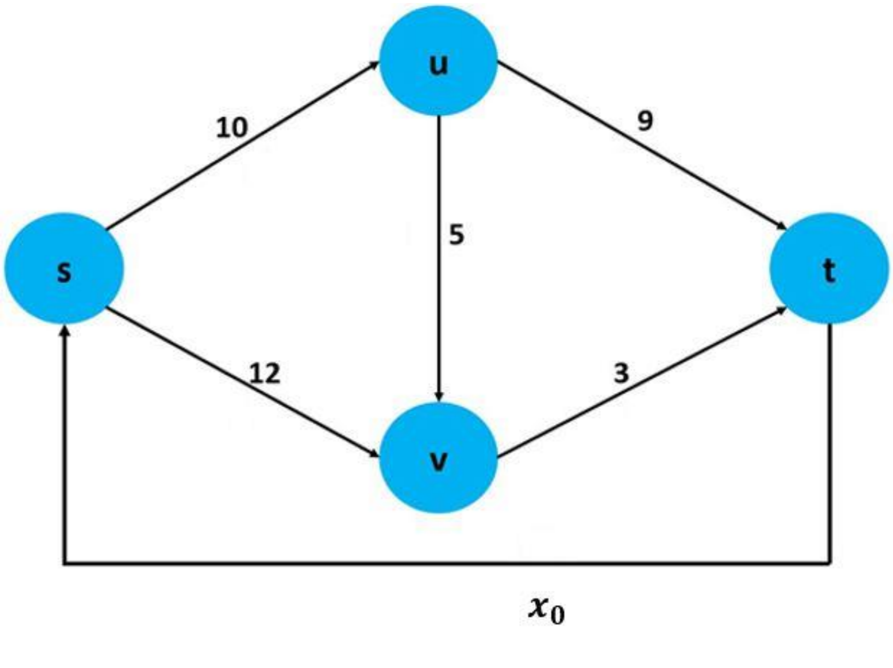
**Primer 4**:
Podjetje želi razposlati maksimalno količino vode na uro iz vozlišča $s$ do vozlišča $t$ na sliki 24. Na poti od vozlišča $s$ do vozlišča $t$ mora tok vode prečkati obe ali določeno vmesno postajo $u$ ali $v$. Povezave med vozlišči si predstavljajmo kot cevi različnih obsegov, kar omeji količino vode, ki lahko preteče po določeni povezavi. Na vsaki povezavi je zapisana kapaciteta pretoka (koliko litrov vode lahko preteče po posamezni cevi med dvema vozliščema v eni uri).

Slika 24: Maksimalni pretok skozi omrežje




Ker želimo ohraniti ravnovesni tok med izvornim vozliščem $s$ in ponornim vozliščem $t$, moramo kreirati navidezno povezavo, ki je usmerjena od vozlišča $t$ do vozlišča $s$. Označimo to povezavo z $x_0$ (slika 25). Ker pretok preko te povezave dejansko ni voda, to povezavo imenujemo navidezna povezava. Kasneje bomo videli zakaj smo uvedli navidezno povezavo.

Slika 25: Ustvarjena navidezna povezava



Za zapis pretoka vzdolž povezav uvedimo novo spremenljivko $x$:

$$x_{ij} = količina\ vode, izražena\ v\ \mathrm{L},$$

$$ki\ preteče\ v\ eni\ uri\ vzdolž\ povezave\ od\ vozlišča\ i\ do\ vozlišča\ j$$

Sestavimo incidenčno matriko za dano omrežje. Od doslej predstavljenih incidenčnih matrik se matrika pri problemu maksimalnega pretoka razlikuje v tem, da v vozliščih nimamo povpraševanja oz. ponudbe, saj nas zanima le, kolikšna je največja količina vode, ki jo lahko distribuiramo po omrežju glede na kapacitete.

| | $x_{su}$ | $x_{sv}$ | $x_{uv}$ | $x_{ut}$ | $x_{vt}$ | $x_0$ |
|---|---|---|---|---|---|---|
| $s$ | -1 | -1 | | | | 1 |
| $u$ | 1 | | -1 | -1 | | |
| $v$ | | 1 | 1 | | -1 | |
| $t$ | | | | 1 | 1 | -1 |
| | 10 | 12 | 5 | 9 | 3 | |

Če z $x_0$ označimo pretok, ki preteče po navidezni povezavi, potem lahko na osnovi izreka o ohranitvi pretoka vidimo, da je

$$x_0 = vsota\ pretoka,$$

$$ki\ vstopi\ v\ izvorno\ vozlišče\ s\ in\ je\ enaka\ vsoti\ pretoka, ki\ izstopi\ iz\ izvornega\ vozlišča\ s$$

Formulirajmo linearni program, ki bo poiskal maksimalen pretok vode v eni uri po omrežju Podjetja z danimi kapacitetami.

Želimo poiskati maksimalno količino vode, ki jo lahko distribuiramo po danem omrežju, torej iščemo maksimum navidezne poznavezave (kajti ta je enaka količini vode, ki jo bomo iz vozlišča $s$ spustili).

$$max\ \mathrm{f} = x_0$$

Pri pretoku vode po posameznih povezavah moramo upoštevati, da so povezave omejene s količino vode, ki lahko teče po njih. Torej, da je tok po omrežju sploh izvedljiv, mora biti količina pretečene vode v mejah med

$$0 \leq pretok\ vode\ po\ povezavi \leq kapaciteta\ povezave$$

$$\left.\begin{array}{c} 0 \leq x_{su} \leq 10 \\ 0 \leq x_{sv} \leq 12 \\ 0 \leq x_{uv} \leq 5 \\ 0 \leq x_{ut} \leq 9 \\ 0 \leq x_{vt} \leq 3 \end{array}\right\} omejitve\ s\ kapaciteto$$

Poleg tega pa moramo upoštevati tudi izrek o ohranitvi pretoka, ki pravi, da mora biti količina vode, ki steče v vozlišče, enaka količini vode, ki iz vozlišča odteče.

Sedaj lahko vidimo zakaj smo uvedli umetno povezavo v omrežju med vozliščema $t$ in $s$. Omogoči nam zapis omejitve o ohranitvi pretoka tudi za vozlišči $s$ in $t$.[20]

$x_0 - x_{su} - x_{sv} = 0$ *Ravnotežje tokov v vozlišču* $s$

$x_{su} - x_{ut} - x_{uv} = 0$ *Ravnotežje tokov v vozlišču*

$x_{sv} + x_{uv} - x_{vt} = 0$ *Ravnotežje tokov v vozlišču* $v$

$x_{ut} + x_{vt} - x_0 = 0$ *Ravnotežje tokov v vozlišču* $t$

#### 1.6.1.1 Reševanje problema maksimalnega pretoka s programom Excel

Zapis linearnega programa za maksimalen pretok vode v eni uri od vozlišča $s$ do vozlišča $t$ bomo izvedli v Excelu, s pomočjo katerega bomo določili tudi vrednosti odločitvenih spremenljivk tako, da bo namenska funkcija zavzela maksimalno vrednost. V Excelu najprej zapišemo podatke v obliki:

[20] *Izrek o ohranitvi pretoka smo definirani tako, da velja samo za notranja vozlišča, brez izvornega vozlišča $s$ in ponornega vozlišča $t$, ki imata samo izhodni tok in vhodni tok v vozlišče. Ob uvedbi navidezne povezave, ki je usmerjena od ponora do izvora, začne tudi za ti dve vozlišči veljati izrek o ohranitvi pretoka.*

Slika 26: Priprava podatkov v Excelu

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Podatki omrežja s podanimi kapacitetami pozamezne povezave | | | | | |
| 2 | | | V vozlišče | | | |
| 3 | | | Vozlišče s | Vozlišče u | Vozlišče v | Vozlišče t |
| 4 | | Vozlišče s | 0 | 10 | 12 | 0 |
| 5 | | Vozlišče u | 0 | 0 | 5 | 9 |
| 6 | | Vozlišče v | 0 | 0 | 0 | 3 |
| 7 | Iz vozlišča | Vozlišče t | **1000** | 0 | 0 | 0 |
| 8 | | | | | | |

kjer so podane kapacitete vsake povezave med dvema vozliščema. Kapaciteta povezave od vozlišča $s$ do vozlišča $u$ je enaka $10\ \mathrm{L/h}$, v vozlišče $v$ pa $12\ \mathrm{L/h}$. Kapaciteta povezave iz vozlišča $u$ do vozlišča $v$ je enaka $5\ \mathrm{L/h}$ ter do vozlišča $t$ $9\ \mathrm{L/h}$. Kapaciteta povezave, ki vodi iz vozlišča $v$ do vozlišča $t$ je enaka $3\ \mathrm{L/h}$. Navidezni povezavi, ki vodi iz končnega vozlišča $t$ do začetnega vozlišča $s$, priredimo neko veliko število, npr. $1000\ \mathrm{L/h}$.

V nadaljevanju kreiramo tabelo za izpis vrednosti odločitvenih spremenljivk (slika 27).

Celice, ki predstavljajo odločitvene spremenljivke pustimo prazne, saj jih bo program zapolnil šele, ko bo rešil problem. Vsaka izmed celic v obsegu C13:F16 predstavlja odločitveno spremenljivko določene povezave med dvema vozliščema. V stolpcu G je zapisana vsota vseh povezav, ki vodijo v določeno vozlišče. Na primer vsota v celici G13 predstavlja količino vode, ki bo v 1 uri odtekla iz vozlišča $s$.

V vrstici 17 so zapisane vsote količin vode, ki v 1 uri pritečejo v določeno vozlišče po posamičnih povezavah. Takšen zapis bo v nadaljevanju koristil pri zapisu omejitev problema, po katerem mora biti pritok v vozlišče enak odtoku iz vozlišča.

Maksimalni pretok predstavlja rešitev namenske funkcije pri danih omejitvah. V konkretnem primeru predstavlja pretok po fiktivni povezavi od vozlišča $t$ do vozlišča $s$.

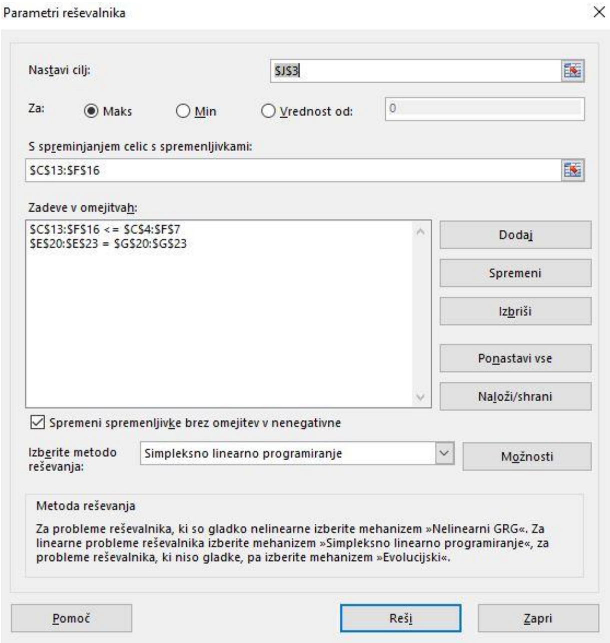
Pripravimo tabelo, kjer definiramo pogoj, ki se nanaša na ravnovesje pretoka skozi vozlišča (vhodni tok v vozlišče mora biti enak izhodnemu toku iz vozlišča).

Slika 27: Priprava podatkov za reševanje v Excelu

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Podatki omrežja s podanimi kapacitetami pozamezne povezave | | | | | | | | |
| 2 | | | | | V vozlišče | | | | | |
| 3 | | | Vozlišče s | Vozlišče u | Vozlišče v | Vozlišče t | | | Maksimalni pretok | =C16 |
| 4 | | Vozlišče s | 0 | 10 | 12 | 0 | | | | |
| 5 | | Vozlišče u | 0 | 0 | 5 | 9 | | | | |
| 6 | | Vozlišče v | 0 | 0 | 0 | 3 | | | | |
| 7 | Iz vozlišča | Vozlišče t | **1000** | 0 | 0 | 0 | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | Tabela rešitev LP | | | | | | | | | |
| 11 | | | | | V vozlišče | | ∑ iz vozlišča | | | |
| 12 | | | Vozlišče s | Vozlišče u | Vozlišče v | Vozlišče t | | | | |
| 13 | | Vozlišče s | 0 | 0 | 0 | 0 | =SUM(C13:F13) | | | |
| 14 | Iz vozlišča | Vozlišče u | 0 | 0 | 0 | 0 | =SUM(C14:F14) | | | |
| 15 | | Vozlišče v | 0 | 0 | 0 | 0 | =SUM(C15:F15) | | | |
| 16 | | Vozlišče t | **0** | 0 | 0 | 0 | =SUM(C16:F16) | | | |
| 17 | ∑ v vozlišče | | =SUM(C13:C16) | =SUM(D13:D16) | =SUM(E13:E16) | =SUM(F13:F16) | | | | |
| 18 | | | | | | | | | | |
| 19 | | Povezava | ∑ v vozlišče | ∑ iz vozlišča | Razlika med ∑ v vozlišče in ∑ iz vozlišča | | Omejitev kapacitet | | | |
| 20 | | Vozlišče s | =C17 | =G13 | =C20-D20 | = | 0 | | | |
| 21 | | Vozlišče u | =D17 | =G14 | =C21-D21 | = | 0 | | | |
| 22 | | Vozlišče v | =E17 | =G15 | =C22-D22 | = | 0 | | | |
| 23 | | Vozlišče t | =F17 | =G16 | =C23-D23 | = | 0 | | | |

Tako zastavljen problem rešimo s pomočjo orodja Reševalnik. Pod omejitve definiramo, da odločitvene spremenljivke ne morejo biti večje od danih kapacitet sistema (označimo celotno tabelo s spremenljivkami in dodamo pogoj, da mora biti manjša ali kvečjemu enaka tabeli s kapacitetami). Kot metodo reševanja izberemo simpleksno linearno programiranje.

Slika 28: Parametri reševalnika




Excel izpiše vrednosti spremenljivk v tabelo, kjer so definirane odločitvene spremenljivke sistema.

Slika 29: Rešitev

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Podatki omrežja s podanimi kapacitetami pozamezne povezave | | | | | | | | | |
| 2 | | | | V vozlišče | | | | | | |
| 3 | | | Vozlišče s | Vozlišče u | Vozlišče v | Vozlišče t | | | Maksimalni pretok | 12 |
| 4 | | Vozlišče s | 0 | 10 | 12 | 0 | | | | |
| 5 | | Vozlišče u | 0 | 0 | 5 | 9 | | | | |
| 6 | | Vozlišče v | 0 | 0 | 0 | 3 | | | | |
| 7 | Iz vozlišča | Vozlišče t | **1000** | 0 | 0 | 0 | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | Tabela rešitev LP | | | | | | | | | |
| 11 | | | | V vozlišče | | | ∑ iz vozlišča | | | |
| 12 | | | Vozlišče s | Vozlišče u | Vozlišče v | Vozlišče t | | | | |
| 13 | Iz vozlišča | Vozlišče s | 0 | 9 | 3 | 0 | 12 | | | |
| 14 | | Vozlišče u | 0 | 0 | 0 | 9 | 9 | | | |
| 15 | | Vozlišče v | 0 | 0 | 0 | 3 | 3 | | | |
| 16 | | Vozlišče t | **12** | 0 | 0 | 0 | 12 | | | |
| 17 | ∑ v vozlišče | | 12 | 9 | 3 | 12 | | | | |
| 18 | | | | | | | | | | |
| 19 | | Povezava | ∑ v vozlišče | ∑ iz vozlišča | Razlika med ∑ v vozlišče in | | Omejitev kapacitet | | | |
| 20 | | Vozlišče s | 12 | 12 | 0 | = | 0 | | | |
| 21 | | Vozlišče u | 9 | 9 | 0 | = | 0 | | | |
| 22 | | Vozlišče v | 3 | 3 | 0 | = | 0 | | | |
| 23 | | Vozlišče t | 12 | 12 | 0 | = | 0 | | | |
| 24 | | | | | | | | | | |

Maksimalni pretok skozi omrežje je 12, torej je maksimalni pretok vode, ki lahko preteče od vozlišča $s$ do vozlišča $t$ v eni uri enak $12\ \mathrm{L}$. Iz poročila o odgovorih, ki ga lahko poljubno izberemo kot prikaz rezultata, preberemo vrednosti pretoka posamezne povezave:

Poročilo o odgovorih pretoka vode Podjetja:

Celica s cilji[21] (Maks)

| Celica | Ime | Izvirna vrednost | Končna vrednost |
|---|---|---|---|
| $J$3 | Maksimalni pretok | 0 | 12 |

Celice s spremenljivkami[22]

| Celica | Ime | Izvirna vrednost | Končna vrednost | Celo število |
|---|---|---|---|---|
| $C$13 | Vozlišče s Vozlišče s | 0 | 0 | Contin |
| $D$13 | Vozlišče s Vozlišče u | 0 | 9 | Contin |
| $E$13 | Vozlišče s Vozlišče v | 0 | 3 | Contin |
| $F$13 | Vozlišče s Vozlišče t | 0 | 0 | Contin |
| $C$14 | Vozlišče u Vozlišče s | 0 | 0 | Contin |
| $D$14 | Vozlišče u Vozlišče u | 0 | 0 | Contin |
| $E$14 | Vozlišče u Vozlišče v | 0 | 0 | Contin |
| $F$14 | Vozlišče u Vozlišče t | 0 | 9 | Contin |
| $C$15 | Vozlišče v Vozlišče s | 0 | 0 | Contin |
| $D$15 | Vozlišče v Vozlišče u | 0 | 0 | Contin |
| $E$15 | Vozlišče v Vozlišče v | 0 | 0 | Contin |
| $F$15 | Vozlišče v Vozlišče t | 0 | 3 | Contin |
| $C$16 | Vozlišče t Vozlišče s | 0 | 12 | Contin |
| $D$16 | Vozlišče t Vozlišče u | 0 | 0 | Contin |
| $E$16 | Vozlišče t Vozlišče v | 0 | 0 | Contin |
| $F$16 | Vozlišče t Vozlišče t | 0 | 0 | Contin |

Vrednosti povezav pretoka vode v $\mathrm{L/uro}$ Podjetja so sledeče:

$$x_{su} = 9, \qquad x_{sv} = 3, \qquad x_{vt} = 3, \qquad x_{ut} = 9, \qquad x_0 = 12$$

Na vseh ostalih povezavah je pretok vode enak 0.

[21] Objective Cells
[22] Variable Cells

#### 1.6.1.2 Reševanje problema maksimalnega pretoka s programom LINGO

Prikažimo še postopek reševanja problema pretoka vode Podjetja s programskim orodjem LINGO:

```
!Namenska funkcija;
max=x0;

!Omejitve kapacitet povezav;
xsu <= 10;
xsv <= 12;
xuv <= 5;
xut <= 9;
xvt <= 3;

!Ravnotežje tokov;
x0 - xsu – xsv = 0;
xsu – xut – xuv = 0;
xsv + xuv – xvt = 0;
xut + xvt – x0 = 0;
```

Poročilo o rešitvi maksimalnega pretoka vode:

```
  Global optimal solution found.
  Objective value:                              12.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             2
  Elapsed runtime seconds:                          0.07

  Model Class:                                        LP


                          Variable           Value        Reduced Cost
                                X0        12.00000            0.000000
                               XSU        9.000000            0.000000
                               XSV        3.000000            0.000000
                               XUV        0.000000            0.000000
                               XUT        9.000000            0.000000
                               XVT        3.000000            0.000000

                               Row    Slack or Surplus      Dual Price
                                 1        12.00000            1.000000
                                 2        1.000000            0.000000
                                 3        9.000000            0.000000
                                 4        5.000000            0.000000
                                 5        0.000000            1.000000
                                 6        0.000000            1.000000
                                 7        0.000000            1.000000
                                 8        0.000000            1.000000
                                 9        0.000000           -1.000000
                                10        0.000000            0.000000
```

Maksimalni pretok skozi omrežje v eni uri je $12 \text{ L}$ vode. Maksimalen pretok se izvrši preko povezav (rešitve pod spremenljivko »FLOW«):

$$x_{su} = 9$$

$$x_{sv} = 3$$
$$x_{vt} = 3$$
$$x_{ut} = 9$$

**Primer 5:**

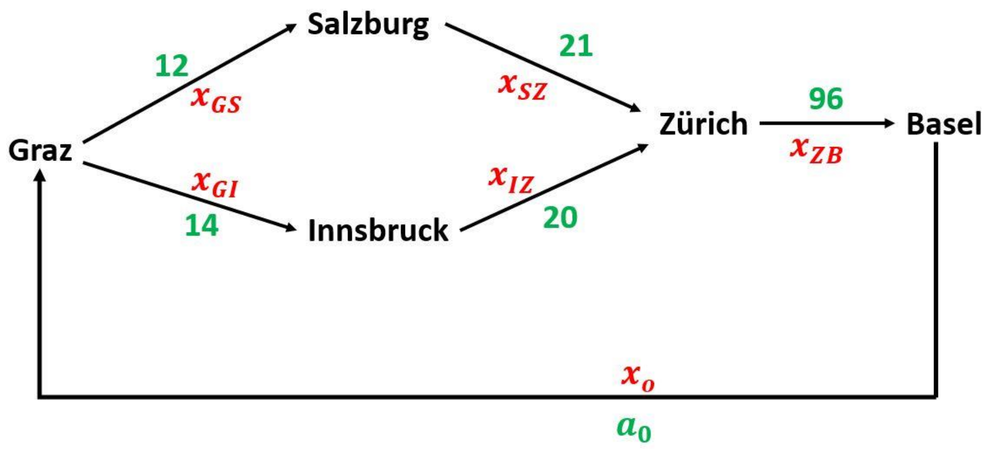
Analizirajmo železniški promet izbranih povezav. Poglejmo, kolikšno je lahko maksimalno število dnevnih povezav na relaciji Graz - Basel. Prestopne povezave so v Salzburgu ali Innsbrucku ter v Zürichu.

Dnevne omejitve voženj vlakov so podane v tabeli:

| Relacije | Maksimalno število voženj na dan |
|---|---|
| Graz - Salzburg | 12 |
| Graz - Innsbruck | 14 |
| Salzburg - Zürich | 21 |
| Innsbruck - Zürich | 20 |
| Zürich - Basel | 96 |

Pripadajoče omrežje z navidezno povezavo v smeri Basel-Graz, predstavljeno z grafom:

Slika 30: Omrežje povezav vlakov z dnevnimi frekvencami




Uvedemo odločitveno spremenljivko $x_{ij}$, ki predstavlja pretok od povezave $i$ do povezave $j$. Povezava $x_{GS}$ predstavlja povezavo Graz – Salzburg, $x_{GI}$ je pretok na povezavi Graz – Innsbruck, $x_{sz}$ je pretok na povezavi Salzburg – Zürich, $x_{IZ}$ je pretok na povezavi Innsbruck – Zürich in $x_{ZB}$, ki predstavlja pretok na povezavi Zürich – Basel.

Pretok po navidezni povezavi naj bo označen z $x_0$ in predstavlja maksimalno število vlakov na dan, ki lahko peljejo iz Gradca do Basla preko prestopnih postaj Salzburg ali Innsbruck in Zürich. Želimo določiti maksimalno število vlakov, ki lahko peljejo od Gradca do Basla v enem dnevu.

Zapišimo namensko funkcijo in omejitve linearnega programa.

$$\max f(x_i) = x_0$$

pri omejitvah: $x_0 = x_{gs} + x_{gi}$

$$x_{gs} = x_{sz}$$

$$x_{gi} = x_{iz}$$

$$x_{sz} + x_{iz} = x_{zb}$$

$$x_{zb} = x_0$$

Pogoji pretoka, vezani na omejitev s kapaciteto povezave so:

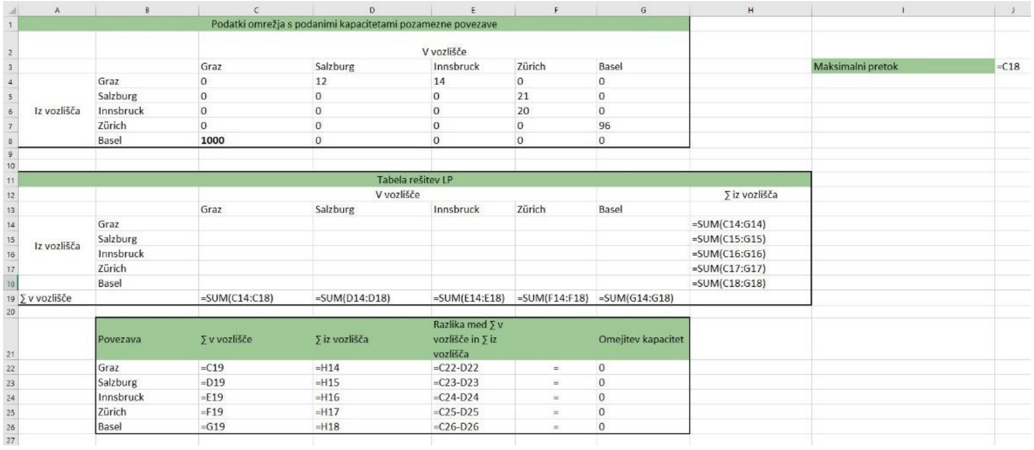
$$x_{gs} \leq 12, \quad x_{gi} \leq 14, \quad x_{iz} \leq 20, \quad x_{sz} \leq 21, \quad x_{zb} \leq 96$$

Ter pogoji nenegativnosti za pretok po posamezni povezavi:

$$x_{gs} \geq 0, \quad x_{gi} \geq 14, \quad x_{iz} \geq 0, \quad x_{sz} \geq 21, \quad x_{zb} \geq 96$$

Problem maksimalnega pretoka v železniškem prometu smo rešili s pomočjo programskega orodja Excel. Nastavitev problema v programu je predstavljena na sliki 31.

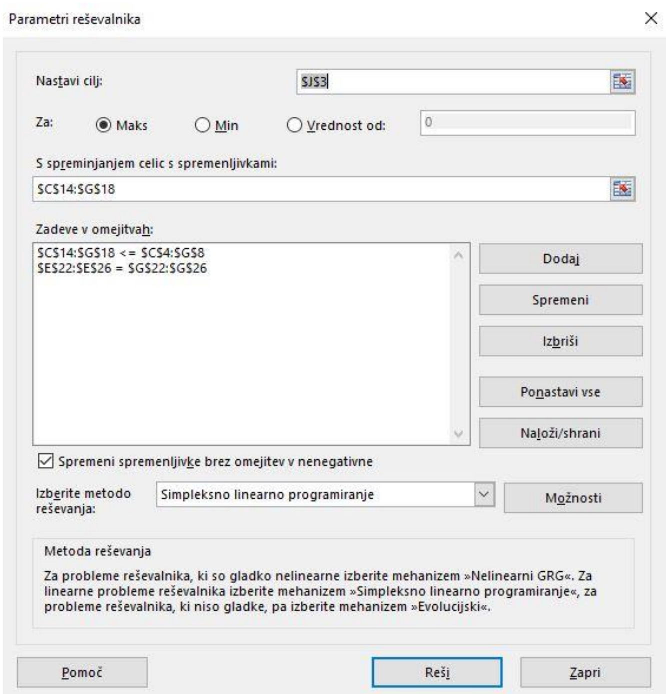
Slika 31: Dnevni pretok vlakov na relaciji Graz – Basel



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Podatki omrežja s podanimi kapacitetami pozamezne povezave | | | | | | | | | |
| 2 | | | | | V vozlišče | | | | | |
| 3 | | | Graz | Salzburg | Innsbruck | Zürich | Basel | | Maksimalni pretok | =C18 |
| 4 | | Graz | 0 | 12 | 14 | 0 | 0 | | | |
| 5 | | Salzburg | 0 | 0 | 0 | 21 | 0 | | | |
| 6 | Iz vozlišča | Innsbruck | 0 | 0 | 0 | 20 | 0 | | | |
| 7 | | Zürich | 0 | 0 | 0 | 0 | 96 | | | |
| 8 | | Basel | **1000** | 0 | 0 | 0 | 0 | | | |
| 9 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| 11 | Tabela rešitev LP | | | | | | | | | |
| 12 | | | | V vozlišče | | | | ∑ iz vozlišča | | |
| 13 | | | Graz | Salzburg | Innsbruck | Zürich | Basel | | | |
| 14 | | Graz | | | | | | =SUM(C14:G14) | | |
| 15 | | Salzburg | | | | | | =SUM(C15:G15) | | |
| 16 | Iz vozlišča | Innsbruck | | | | | | =SUM(C16:G16) | | |
| 17 | | Zürich | | | | | | =SUM(C17:G17) | | |
| 18 | | Basel | | | | | | =SUM(C18:G18) | | |
| 19 | ∑ v vozlišče | | =SUM(C14:C18) | =SUM(D14:D18) | =SUM(E14:E18) | =SUM(F14:F18) | =SUM(G14:G18) | | | |
| 20 | | | | | | | | | | |
| 21 | | Povezava | ∑ v vozlišče | ∑ iz vozlišča | Razlika med ∑ v vozlišče in ∑ iz vozlišča | | Omejitev kapacitet | | | |
| 22 | | Graz | =C19 | =H14 | =C22-D22 | = | 0 | | | |
| 23 | | Salzburg | =D19 | =H15 | =C23-D23 | = | 0 | | | |
| 24 | | Innsbruck | =E19 | =H16 | =C24-D24 | = | 0 | | | |
| 25 | | Zürich | =F19 | =H17 | =C25-D25 | = | 0 | | | |
| 26 | | Basel | =G19 | =H18 | =C26-D26 | = | 0 | | | |
| 27 | | | | | | | | | | |

V Reševalniku nastavimo pravilno celico za maksimalni pretok, ki je cilj optimizacije. V polje *S spreminjanjem celic* s spremenljivkami vnesemo matriko odločitvenih spremenljivk, ki je pripravljena za izpis rešitve, dodamo omejitvi glede kapacitet povezav in pogoja, da mora biti razlika med vhodnim tokom v vozlišče in izhodnim tokom enaka 0. Izberemo simpleksno linearno programiranje.

Slika 32: Nastavitev reševalnika




Rešitev problema maksimalnega števila vlakov, ki peljejo na relaciji Graz – Basel v enem dnevu:

Celica s cilji (Maks)

| Celica | Ime | Izvirna vrednost | Končna vrednost |
|---|---|---|---|
| $J$3 | Maksimalni pretok | 0 | 26 |

Celice s spremenljivkami

| Celica | Ime | Izvirna vrednost | Končna vrednost | Celo število |
|---|---|---|---|---|
| $C$14 | Graz Graz | 0 | 0 | Contin |
| $D$14 | Graz Salzburg | 0 | 12 | Contin |
| $E$14 | Graz Innsbruck | 0 | 14 | Contin |
| $F$14 | Graz Zürich | 0 | 0 | Contin |
| $G$14 | Graz Basel | 0 | 0 | Contin |
| $C$15 | Salzburg Graz | 0 | 0 | Contin |
| $D$15 | Salzburg Salzburg | 0 | 0 | Contin |
| $E$15 | Salzburg Innsbruck | 0 | 0 | Contin |
| $F$15 | Salzburg Zürich | 0 | 12 | Contin |
| $G$15 | Salzburg Basel | 0 | 0 | Contin |
| $C$16 | Innsbruck Graz | 0 | 0 | Contin |
| $D$16 | Innsbruck Salzburg | 0 | 0 | Contin |
| $E$16 | Innsbruck Innsbruck | 0 | 0 | Contin |
| $F$16 | Innsbruck Zürich | 0 | 14 | Contin |
| $G$16 | Innsbruck Basel | 0 | 0 | Contin |
| $C$17 | Zürich Graz | 0 | 0 | Contin |
| $D$17 | Zürich Salzburg | 0 | 0 | Contin |
| $E$17 | Zürich Innsbruck | 0 | 0 | Contin |
| $F$17 | Zürich Zürich | 0 | 0 | Contin |
| $G$17 | Zürich Basel | 0 | 26 | Contin |
| $C$18 | Basel Graz | 0 | 26 | Contin |
| $D$18 | Basel Salzburg | 0 | 0 | Contin |
| $E$18 | Basel Innsbruck | 0 | 0 | Contin |
| $F$18 | Basel Zürich | 0 | 0 | Contin |
| $G$18 | Basel Basel | 0 | 0 | Contin |

Maksimalno število vlakov, ki lahko v enem dnevu peljejo na relaciji Graz – Basel je 26. 12 vlakov lahko pelje iz Gradca do Salzburga, 14 pa jih lahko pelje iz Gradca do Innsbrucka. Na povezavi Innsbruck – Zürich lahko pelje maksimalno 14 vlakov na dan, ter 12 vlakov na povezavi Salzburg – Zürich. Iz Züricha lahko prispe v Basel 26 vlakov na dan.

Prikažimo še sintakso za reševanje problema maksimalnega dnevnega pretoka vlakov na relaciji Graz – Basel v programu LINGO:

```
!Namenska funkcija;
max=x0;

!Omejitve kapacitet povezav;
xgs <= 12;
xgi <= 14;
xiz <= 20;
xsz <= 21;
xzb <= 96;

!Ravnotežje tokov;
x0 = xgs + xgi;
xgs = xsz;
xgi = xiz;
xzb = xsz + xiz;
xzb=x0;
```

Poročilo o rešitvi programa LINGO:

```
   Global optimal solution found.
 Objective value:                              26.00000
 Infeasibilities:                              0.000000
 Total solver iterations:                             0
 Elapsed runtime seconds:                          0.04

 Model Class:                                        LP


                            Variable           Value        Reduced Cost
                                  X0        26.00000            0.000000
                                 XGS        12.00000            0.000000
                                 XGI        14.00000            0.000000
                                 XIZ        14.00000            0.000000
                                 XSZ        12.00000            0.000000
                                 XZB        26.00000            0.000000

                                 Row    Slack or Surplus      Dual Price
                                   1        26.00000            1.000000
                                   2        0.000000            1.000000
                                   3        0.000000            1.000000
                                   4        6.000000            0.000000
                                   5        9.000000            0.000000
                                   6        70.00000            0.000000
                                   7        0.000000            0.000000
                                   8        0.000000           -1.000000
                                   9        0.000000           -1.000000
                                  10        0.000000            1.000000
                                  11        0.000000           -1.000000
```

Rešitve programa LINGO so enake kot rešitve v programu Excel.

**Primer 6:**

Pretok blaga poteka od začetnega vozlišča do končnega vozlišča preko določenih povezav. V slednjem primeru moramo upoštevati tudi preference in kapacitete individualnih povezav med vozlišči znotraj omrežja in ne samo skupnega pretoka od vstopnega do končnega vozlišča. Podatki o omrežju so predstavljeni v tabeli:

| Izvor/Ponor | Vozlišče 4 | Vozlišče 5 | Vozišče 6 | Ponudba |
|---|---|---|---|---|
| Vozlišče 1 | 60 | 80 | 60 | 200 |
| Vozlišče 2 | 50 | 85 | 105 | 240 |
| Vozlišče 3 | 120 | 120 | 40 | 280 |
| Povpraševanje | 230 | 285 | 205 | 720 |

Celotna ponudba v vozlišču 1 je 200 enot. Od vseh enot, ki so na voljo, je treba 60 enot prepeljati v vozlišče 4, 80 v vozlišče 5 in 60 v vozlišče 6. V vozlišču 2 čaka na transport 240 enot, od tega jih mora 50 prispeti v vozlišče 4, 85 v vozlišče 5 in 105 v vozlišče 6. 280 enot blaga čaka na transport v izvornem vozlišču 3, iz katerega je treba 120 enot prepeljati v vozlišče 4, 120 enot v vozlišče 5 in 40 enot v vozlišče 6. Na drugi strani pa je treba s transportom zadovoljiti potrebe v ponornih vozliščih (povpraševanje) – vozlišče 4 potrebuje vsega skupaj 230 enot, vozlišče 5 285 enot in vozlišče 6 205 enot.

Da bi lahko razrešili takšen problem, moramo zastaviti tri linearne programe – za vsako izmed izvornih vozlišč posebej. V spodnji tabeli predstavljamo podatke za vsako vozlišče:

| | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $x_{23}$ | $x_{24}$ | $x_{25}$ | $x_{26}$ | $x_{34}$ | $x_{35}$ | $x_{36}$ | $x_{45}$ | $x_{46}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\geq -200$ |
| 2 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | $\geq 0$ |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 | $\geq 0$ |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 | $\geq 60$ |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | $\geq 80$ |
| 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | $\geq 60$ |
| $c_{ij}$ | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | |

V naslednji tabeli so prikazani podatki za izdelavo linearnega programa z vidika izvornega vozlišča 2. Incidenčna matrika in cena prevoza 1 enote blaga na posamični povezavi ostanejo enake, uvedemo le nove odločitvene spremenljivke. Namesto z $x$, tokrat število enot, prepeljanih po določeni povezavi, označimo z $y$.

| | $y_{13}$ | $y_{14}$ | $y_{15}$ | $y_{16}$ | $y_{23}$ | $y_{24}$ | $y_{25}$ | $y_{26}$ | $y_{34}$ | $y_{35}$ | $y_{36}$ | $y_{45}$ | $y_{46}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\geq 0$ |
| 2 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | $\geq -240$ |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 | $\geq 0$ |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 | $\geq 50$ |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | $\geq 85$ |
| 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | $\geq 105$ |
| $c_{ij}$ | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | |

V naslednji tabeli so prikazani podatki za sestavo linearnega programa s strani izvornega vozlišča 3. Incidenčna matrika in cene prevoza 1 enote blaga po posamičnem vozlišču ostajajo enake, postavimo le dodatno oznako za neznanke, tokrat $z$.

| | $z_{13}$ | $z_{14}$ | $z_{15}$ | $z_{16}$ | $z_{23}$ | $z_{24}$ | $z_{25}$ | $z_{26}$ | $z_{34}$ | $z_{35}$ | $z_{36}$ | $z_{45}$ | $z_{46}$ | b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\geq 0$ |
| | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | $\geq 0$ |
| | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | 0 | $\geq -280$ |
| | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | -1 | -1 | $\geq 120$ |
| | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | $\geq 120$ |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | $\geq 40$ |
| $c_{ij}$ | 10 | 7 | 23 | 26 | 12 | 9 | 21 | 28 | 14 | 21 | 23 | 7 | 12 | |

Lahko bi zapisali linearni program za vsako izmed zapisanih tabel posebej, lahko pa združimo kar vse tri linearne programe v enega.

Cilj problema je zmanjšati globalne stroške prevoza (torej minimirati vsoto vseh treh posamičnih linearnih programov). V glavnem programu nastopajo tri različne spremenljivke ($x, y$ in $z$). Program ima skupaj 18 omejitev – 6 omejitev za vozlišče 1, 6 omejitev za vozlišče 2 in 6 omejitev za vozlišče 3.

Optimalna rešitev združenega programa mora biti enaka rešitvam posameznih treh programov. Pri reševanju problema si pomagajmo s programskim orodjem LINGO.

Najprej zapišemo linearni program za vozlišče 1:

**Minimum namenske funkcije**

$$\min f(x_{ij}) = 10x_{13} + 7x_{14} + 23x_{15} + 26x_{16} + 12x_{23} + 9x_{24} + 21x_{25} + 28x_{26} + 14x_{34} + 21x_{35} + 23x_{36} + 7x_{45} + 12x_{46}$$

pri omejitvah: $-x_{13} - x_{14} - x_{15} - x_{16} \geq -200$

$$-x_{23} - x_{24} - x_{25} - x_{26} \geq 0$$
$$x_{13} + x_{23} - x_{34} - x_{35} - x_{36} \geq 0$$
$$x_{14} + x_{24} + x_{34} - x_{45} - x_{46} \geq 60$$
$$x_{15} + x_{25} + x_{35} + x_{45} \geq 80$$
$$x_{16} + x_{26} + x_{36} + x_{46} \geq 60$$

pogoji nenegativnosti: $x_{13}, x_{14}, x_{15}, x_{16}, x_{23}, x_{24}, x_{25}, x_{26}, x_{34}, x_{35}, x_{36}, x_{45}, x_{46} \geq 0$

Linearni program za izvorno vozlišče 2:

**Minimum namenske funkcije:**

$$\min f(y_{ij}) = 10y_{13} + 7y_{14} + 23y_{15} + 26y_{16} + 12y_{23} + 9y_{24} + 21y_{25} + 28y_{26} + 14y_{34} + 21y_{35} + 23y_{36} + 7y_{45} + 12y_{46}$$

pri omejitvah: $-y_{13} - y_{14} - y_{15} - y_{16} \geq 0$

$$-y_{23} - y_{24} - y_{25} - y_{26} \geq -240$$
$$y_{13} + y_{23} - y_{34} - y_{35} - y_{36} \geq 0$$
$$y_{14} + y_{24} + y_{34} - y_{45} - y_{46} \geq 50$$
$$y_{15} + y_{25} + y_{35} + y_{45} \geq 85$$
$$y_{16} + y_{26} + y_{36} + y_{46} \geq 105$$

pogoji nenegativnosti: $y_{13}, y_{14}, y_{15}, y_{16}, y_{23}, y_{24}, y_{25}, y_{26}, y_{34}, y_{35}, y_{36}, y_{45}, y_{46} \geq 0$

Ter še linearni program za izvorno vozlišče 3:

**Minimum namenske funkcije:**

$$\min f(z_{ij}) = 10z_{13} + 7z_{14} + 23z_{15} + 26z_{16} + 12z_{23} + 9z_{24} + 21z_{25} + 28z_{26} + 14z_{34} + 21z_{35} + 23z_{36} + 7z_{45} + 12z_{46}$$

pri omejitvah:

$$-z_{13} - z_{14} - z_{15} - z_{16} \geq 0$$
$$-z_{23} - z_{24} - z_{25} - z_{26} \geq 0$$
$$z_{13} + z_{23} - z_{34} - z_{35} - z_{36} \geq -280$$
$$z_{14} + z_{24} + z_{34} - z_{45} - z_{46} \geq 120$$
$$z_{15} + z_{25} + z_{35} + z_{45} \geq 120$$
$$z_{16} + z_{26} + z_{36} + z_{46} \geq 40$$

pogoji nenegativnosti: $z_{13}, z_{14}, z_{15}, z_{16}, z_{23}, z_{24}, z_{25}, z_{26}, z_{34}, z_{35}, z_{36}, z_{45}, z_{46} \geq 0$

Vse tri linearne programe lahko združimo v enega. Dobimo naslednji linearni program:

**Minimum namenske funkcije:**

$$\min f(x_{ij}, y_{ij}, z_{ij}) = (10x_{13} + 7x_{14} + 23x_{15} + 26x_{16} + 12x_{23} + 9x_{24} + 21x_{25} + 28x_{26} + 14x_{34} + 21x_{35} + 23x_{36} + 7x_{45} + 12x_{46}) + (10y_{13} + 7y_{14} + 23y_{15} + 26y_{16} + 12y_{23} + 9y_{24} + 21y_{25} + 28y_{26} + 14y_{34} + 21y_{35} + 23y_{36} + 7y_{45} + 12y_{46}) + 10z_{13} + 7z_{14} + 23z_{15} + 26z_{16} + 12z_{23} + 9z_{24} + 21z_{25} + 28z_{26} + 14z_{34} + 21z_{35} + 23z_{36} + 7z_{45} + 12z_{46}$$

pri omejitvah:

$$-x_{13} - x_{14} - x_{15} - x_{16} \geq -200$$
$$-x_{23} - x_{24} - x_{25} - x_{26} \geq 0$$
$$x_{13} + x_{23} - x_{34} - x_{35} - x_{36} \geq 0$$
$$x_{14} + x_{24} + x_{34} - x_{45} - x_{46} \geq 60$$
$$x_{15} + x_{25} + x_{35} + x_{45} \geq 80$$
$$x_{16} + x_{26} + x_{36} + x_{46} \geq 60$$

$$-y_{13} - y_{14} - y_{15} - y_{16} \geq 0$$
$$-y_{23} - y_{24} - y_{25} - y_{26} \geq -240$$
$$y_{13} + y_{23} - y_{34} - y_{35} - y_{36} \geq 0$$
$$y_{14} + y_{24} + y_{34} - y_{45} - y_{46} \geq 50$$
$$y_{15} + y_{25} + y_{35} + y_{45} \geq 85$$
$$y_{16} + y_{26} + y_{36} + y_{46} \geq 105$$
$$-z_{13} - z_{14} - z_{15} - z_{16} \geq 0$$
$$-z_{23} - z_{24} - z_{25} - z_{26} \geq 0$$
$$z_{13} + z_{23} - z_{34} - z_{35} - z_{36} \geq -280$$
$$z_{14} + z_{24} + z_{34} - z_{45} - z_{46} \geq 120$$
$$z_{15} + z_{25} + z_{35} + z_{45} \geq 120$$
$$z_{16} + z_{26} + z_{36} + z_{46} \geq 40$$

### Pogoji nenegativnosti:

$$x_{13}, x_{14}, x_{15}, x_{16}, x_{23}, x_{24}, x_{25}, x_{26}, x_{34}, x_{35}, x_{36}, x_{45}, x_{46}, y_{13}, y_{14}, y_{15}, y_{16}, y_{23}, y_{24}, y_{25}, y_{26}, y_{34}, y_{35}, y_{36}, y_{45}, y_{46}, z_{13}, z_{14}, z_{15}, z_{16}, z_{23}, z_{24}, z_{25}, z_{26}, z_{34}, z_{35}, z_{36}, z_{45}, z_{46} \geq 0$$

Tako zapisan linearni program zapišemo v LINGO sintaksi:

```
!Namenska funkcija;
min = 10*x13 + 7*x14 + 23*x15 + 26*x16 + 12*x23 + 9*x24 +
21*x25 + 28*x26 +14*x34 + 21*x35 + 23*x36 + 7*x45 + 12*x46
+ 10*y13 + 7*y14 + 23*y15 + 26*y16 + 12*y23 + 9*y24 + 21*y25
+ 28*y26 + 14*y34 + 21*y35 + 23*y36 + 7*y45 + 12*y46 +
10*z13 + 7*z14 + 23*z15 + 26*z16 + 12*z23 + 9*z24 + 21*z25
+ 28*z26 + 14*z34 + 21*z35 + 23*z36 +7*z45 + 12*z46;
!Omejitve;
-x13 -x14 - x15 - x16 >= -200;
-x23 - x24 - x25 - x26 >= 0;
x13 + x23 - x34 - x35 - x36 >= 0;
x14 + x24 + x34 - x45 - x46 >= 60;
x15 + x25 + x35 + x45 >= 80;
```

```
x16 + x26 + x36 + x46 >= 60;
- y13 -y14 - y15 - y16 >= 0;
-y23 - y24 - y25 - y26 >= -240;
y13 + y23 - y34 - y35 - y36 >= 0;
y14 + y24 + y34 - y45 - y46 >= 50;
y15 + y25 + y35 + y45 >= 85;
y16 + y26 + y36 + y46 >= 105;
- z13 -z14 - z15 - z16 >= 0;
-z23 - z24 - z25 - z26 >= 0;
z13 + z23 - z34 - z35 - z36 >= -280;
z14 + z24 + z34 - z45 - z46 >= 120;
z15 + z25 + z35 + z45 >= 120;
z16 + z26 + z36 + z46 >= 40;
```

Poročilo o rešitvi:

```
Global optimal solution found.
Objective value:                              11815.00
Infeasibilities:                              0.000000
Total solver iterations:                            14
Elapsed runtime seconds:                          0.04

Model Class:                                        LP


                              Variable           Value        Reduced Cost
                                   X13        0.000000            0.000000
                                   X14        200.0000            0.000000
                                   X15        0.000000            9.000000
                                   X16        0.000000            7.000000
                                   X23        0.000000            2.000000
                                   X24        0.000000            2.000000
                                   X25        0.000000            7.000000
                                   X26        0.000000            9.000000
                                   X34        0.000000            17.00000
                                   X35        0.000000            17.00000
                                   X36        0.000000            14.00000
                                   X45        80.00000            0.000000
                                   X46        60.00000            0.000000
                                   Y13        0.000000            0.000000
                                   Y14        0.000000            0.000000
                                   Y15        0.000000            9.000000
                                   Y16        0.000000            7.000000
                                   Y23        0.000000            0.000000
                                   Y24        240.0000            0.000000
                                   Y25        0.000000            5.000000
                                   Y26        0.000000            7.000000
                                   Y34        0.000000            17.00000
                                   Y35        0.000000            17.00000
                                   Y36        0.000000            14.00000
```

| | | |
|---|---|---|
| Y45 | 85.00000 | 0.000000 |
| Y46 | 105.0000 | 0.000000 |
| Z13 | 0.000000 | 17.00000 |
| Z14 | 0.000000 | 0.000000 |
| Z15 | 0.000000 | 9.000000 |
| Z16 | 0.000000 | 10.00000 |
| Z23 | 0.000000 | 17.00000 |
| Z24 | 0.000000 | 0.000000 |
| Z25 | 0.000000 | 5.000000 |
| Z26 | 0.000000 | 10.00000 |
| Z34 | 240.0000 | 0.000000 |
| Z35 | 0.000000 | 0.000000 |
| Z36 | 40.00000 | 0.000000 |
| Z45 | 120.0000 | 0.000000 |
| Z46 | 0.000000 | 3.000000 |

Rešitev:

- iz vozlišča 1 prepeljemo vseh 200 enot najprej v vozlišče 4, od tam pa 80 enot v vozlišče 5 in 60 enot v vozlišče 6. V vozlišču 4 ostane 60 enot blaga;
- iz vozlišča 2 prepeljemo vseh 240 enot najprej v vozlišče 4, nato pa iz vozlišča 4 85 enot naprej v vozlišče 5 in 105 enot v vozlišče 6. V vozlišču 4 bo ostalo 50 enot blaga;
- iz vozlišča 3 dostavimo 240 enot blaga najprej v vozlišče 4, od tam pa 120 enot blaga v vozlišče 5. V vozlišču 4 bo tako ostalo potrebnih 120 enot blaga. 40 enot blaga, ki ostanejo v vozlišču 3 pa odpeljemo neposredno v vozlišče 6.

Tako smo iz izvornih vozlišč odpeljali vso blago, ki so ga ta imela na voljo, hkrati pa smo zadovoljili vse potrebe vozlišč in upoštevali omejitve glede smeri potovanja blaga. Minimalni stroški prevoza pri tem znašajo 11.815 €.

#### 1.6.1.3 Omejitve zmogljivosti vozlišč ali povezav

V resničnem svetu se ves čas srečujemo z največjimi možnimi omejitvami prometa po cestah, železnicah, transporterjih in drugih transportnih sredstvih. Takšne omejitve lahko zapišemo s povezavami obravnavanega modela (največji promet, ki ga lahko pretovorimo skozi določeno povezavo). Omejitve pa se lahko pojavijo tudi zaradi kapacitet skladišč, distribucijskih centrov itd. Tovrstne omejitve lahko preslikamo na vozlišča modela.

Nekatere kapacitete in omejitve so določene s fizičnimi omejitvami (velikost in število razpoložljivih transportnih enot, zmogljivost nalaganja na transportno sredstvo, število razpoložljivih transportnih sredstev), druge omejitve pa lahko povzročijo okoljski vidiki (največje število tovornjakov, ki se lahko peljejo čez nek odsek ceste čez dan, omejitve hrupa, največje dovoljeno onesnaženje itd.).
Da bi ponazorili, kako tovrstne omejitve vplivajo na pretok blaga skozi omrežje, poglejmo naslednji primer.

**Primer 7:**

Kot osnovo vzemimo primer 6. Predpostavimo, da imamo sedaj še dodatni omejitvi z okoljskega vidika glede količine prometa, ki lahko steče skozi vozlišči 3 in 4. Predpostavimo, da lahko le 280 enot blaga prepeljemo skozi vozlišče 3 in le 295 enot blaga lahko zapusti vozlišče 4. V linearni program za primer 6 dodamo še dve novi omejitvi:

$$x_{34} + x_{35} + x_{36} + y_{34} + y_{35} + y_{36} + z_{34} + z_{35} + z_{36} \leq 280$$
$$x_{45} + x_{46} + y_{45} + y_{46} + z_{45} + z_{46} \geq 295$$

Če se spomnimo primera 6, kjer še nista upoštevani dodatni omejitvi, je bilo v vozlišču 3 skupno 280 enot blaga, ki so čakale, da zapustijo vozlišče. Ko dodamo zgornjo omejitev povzročimo, da v vozlišču 3 ni več prostora za sprejem novega blaga.

Če pogledamo vozlišče 4, je optimalna rešitev primera 6 pokazala, da vozlišče 4 zapusti 285 enot blaga, ki se transportirajo v vozlišče 5 in 165 enot blaga, ki se prepeljejo v vozlišče 6, kar močno preseže na novo postavljeno omejitev.

V novem linearnem programu ni več samo 18 omejitev, temveč se pridružita še 2 dodatni omejitvi, ki združujeta vsa tri izvorna vozlišča. Ti dve omejitvi prevzameta vlogo združevanja:

```
!Namenska funkcija;
min = 10*x13 + 7*x14 + 23*x15 + 26*x16 + 12*x23 + 9*x24 +
21*x25 + 28*x26 +14*x34 + 21*x35 + 23*x36 + 7*x45 + 12*x46
+
10*y13 + 7*y14 + 23*y15 + 26*y16 + 12*y23 + 9*y24 + 21*y25
+ 28*y26 + 14*y34 + 21*y35 + 23*y36 + 7*y45 + 12*y46 +
10*z13 + 7*z14 + 23*z15 + 26*z16 + 12*z23 + 9*z24 + 21*z25
+ 28*z26 + 14*z34 + 21*z35 + 23*z36 +7*z45 + 12*z46;
!Omejitve;
-x13 -x14 - x15 - x16 >= -200;
-x23 - x24 - x25 - x26 >= 0;
x13 + x23 - x34 - x35 - x36 >= 0;
x14 + x24 + x34 - x45 - x46 >= 60;
x15 + x25 + x35 + x45 >= 80;
x16 + x26 + x36 + x46 >= 60;
- y13 -y14 - y15 - y16 >= 0;
-y23 - y24 - y25 - y26 >= -240;
y13 + y23 - y34 - y35 - y36 >= 0;
y14 + y24 + y34 - y45 - y46 >= 50;
y15 + y25 + y35 + y45 >= 85;
y16 + y26 + y36 + y46 >= 105;
- z13 -z14 - z15 - z16 >= 0;
-z23 - z24 - z25 - z26 >= 0;
z13 + z23 - z34 - z35 - z36 >= -280;
z14 + z24 + z34 - z45 - z46 >= 120;
z15 + z25 + z35 + z45 >= 120;
z16 + z26 + z36 + z46 >= 40;
!Dodatni omejitvi;
x34 + x35 + x36 +y34 +y35 +y36 + z34 + z35 + z36 <= 280;
x45 + x46 + y45 + y46 + z45 + z46 <= 295;
```

Poročilo o rešitvi v LINGO:

```
Global optimal solution found.
Objective value:                              11990.00
Infeasibilities:                              0.000000
Total solver iterations:                            16
Elapsed runtime seconds:                          0.04

Model Class:                                        LP


                         Variable           Value        Reduced Cost
                              X13        0.000000            0.000000
                              X14        200.0000            0.000000
                              X15        0.000000            4.000000
                              X16        0.000000            2.000000
                              X23        0.000000            2.000000
                              X24        0.000000            2.000000
                              X25        0.000000            2.000000
                              X26        0.000000            4.000000
                              X34        0.000000            17.00000
                              X35        0.000000            12.00000
                              X36        0.000000            9.000000
                              X45        80.00000            0.000000
                              X46        60.00000            0.000000
                              Y13        0.000000            0.000000
                              Y14        0.000000            0.000000
                              Y15        0.000000            4.000000
                              Y16        0.000000            2.000000
                              Y23        0.000000            0.000000
                              Y24        205.0000            0.000000
                              Y25        35.00000            0.000000
                              Y26        0.000000            2.000000
                              Y34        0.000000            17.00000
                              Y35        0.000000            12.00000
                              Y36        0.000000            9.000000
                              Y45        50.00000            0.000000
                              Y46        105.0000            0.000000
                              Z13        0.000000            17.00000
                              Z14        0.000000            0.000000
                              Z15        0.000000            9.000000
                              Z16        0.000000            10.00000
                              Z23        0.000000            17.00000
                              Z24        0.000000            0.000000
                              Z25        0.000000            5.000000
                              Z26        0.000000            10.00000
                              Z34        120.0000            0.000000
                              Z35        120.0000            0.000000
                              Z36        40.00000            0.000000
                              Z45        0.000000            5.000000
                              Z46        0.000000            8.000000
```

Rešitev optimalne distribucije blaga se je tokrat prilagodila dvema naknadno postavljenima pogojema:

- rešitev glede vozlišča 1 je ostala enaka napram rešitvi primera 6. 200 enot blaga najprej prepeljemo v vozlišče 4, od tam pa 80 enot v vozlišče 5 in 60 v vozlišče 6. Ostane 60 enot blaga v vozlišču 4;

- iz vozlišča 2 odpeljemo 205 enot v vozlišče 4 in nato iz njega 50 enot v vozlišče 5 in 105 enot v vozlišče 6. V vozlišču 4 ostane 50 enot blaga. Iz vozlišča 2 odpeljemo še 35 enot neposredno v vozlišče 5 in tako izpraznimo zaloge v vozlišču 2, zapolnimo pa povpraševanje vozlišča 5;
- iz vozlišča 3 peljemo 120 enot blaga v vozlišče 4 in s tem zapolnimo povpraševanje v tem vozlišču, 120 enot iz vozlišča 3 peljemo neposredno v vozlišče 5 in 40 enot blaga neposredno v vozlišče 6;
- iz vozlišča 4 tako odpeljemo le 130 enot blaga v vozlišče 5 in 165 enot v vozlišče 6. Skupaj vozlišče zapusti 295 enot blaga, kar je natanko toliko kot smo določili v omejitvi. Spomnimo, da je brez postavljene omejitve to vozlišče zapustilo 450 enot blaga.

Vrednost namenske funkcije se je zaradi dodatno postavljenih pogojev povišala in znaša 11.990 €. Da bi bolje razumeli problem, poglejmo še rešitve dualnega problema. Spomnimo, da so rešitve dualnega problema enake senčnim cenam vsake omejitve v osnovnem primeru – senčne cene prikazujemo v spodnjem poročilu.

Poročilo o rešitvi dualnega linearnega programa:

```
Global optimal solution found.
  Objective value:                              11990.00
  Infeasibilities:                              0.000000
  Total solver iterations:                            16
  Elapsed runtime seconds:                          0.04

  Model Class:                                        LP


                                    Row    Slack or Surplus      Dual Price
                                      1        11990.00          -1.000000
                                      2        0.000000           0.000000
                                      3        0.000000           0.000000
                                      4        0.000000          -10.00000
                                      5        0.000000          -7.000000
                                      6        0.000000          -19.00000
                                      7        0.000000          -24.00000
                                      8        0.000000          -2.000000
                                      9        0.000000           0.000000
                                     10        0.000000          -12.00000
                                     11        0.000000          -9.000000
                                     12        0.000000          -21.00000
                                     13        0.000000          -26.00000
                                     14        0.000000          -7.000000
                                     15        0.000000          -5.000000
                                     16        0.000000           0.000000
                                     17        0.000000          -14.00000
                                     18        0.000000          -21.00000
```

```
                    19        0.000000          -23.00000
                    20        0.000000           0.000000
                    21        0.000000           5.000000
```

Prva tabela spodaj prikazuje optimalno distribucijo blaga po povezavah v omrežju, tabela na naslednji strani pa senčne cene transporta blaga posameznih vozlišč. V tabeli na naslednji strani so v stolpcu $x$ prikazane senčne cene transporta, ki izhaja iz izvornega vozlišča 1, v stolpcu $y$ so senčne cene transporta, ki izhaja iz izvornega vozlišča 2 in v stolpcu z so podane senčne cene transporta, ki izhaja iz izvornega vozlišča 3.

| povezava | $x$ | $y$ | $z$ |
|---|---|---|---|
| 1-3 | 0 | 0 | 0 |
| 1-4 | 200 | 0 | 0 |
| 1-5 | 0 | 0 | 0 |
| 1-6 | 0 | 0 | 0 |
| 2-3 | 0 | 0 | 0 |
| 2-4 | 0 | 205 | 0 |
| 2-5 | 0 | 35 | 0 |
| 2-6 | 0 | 0 | 0 |
| 3-4 | 0 | 0 | 120 |
| 3-5 | 0 | 0 | 120 |
| 3-6 | 0 | 0 | 40 |
| 4-5 | 80 | 50 | 0 |
| 4-6 | 60 | 105 | 0 |

| vozlišče | $x$ | $y$ | $z$ |
|---|---|---|---|
| 1 | 0 | 2 | 7 |
| 2 | 0 | 0 | 5 |
| 3 | 10 | 12 | 14 |
| 4 | 7 | 9 | 21 |
| 5 | 19 | 21 | 23 |
| 6 | 24 | 26 | 5 |

Seveda tudi dvema dodatnima omejitvama pripadata senčni ceni:

- senčna cena omejitve s kapaciteto v vozlišču 3 je 0;
- senčna cena omejitve s kapaciteto v vozlišču 4 je 5.

Za interpretacijo prikazanih rezultatov, si oglejmo izčrpavanje senčnih cen vzdolž vsake povezave v omrežju. Tabela spodaj prikazuje naraščanje senčne cene transporta, ki se pojavi vzdolž vsake povezave in je izračunana z razliko med senčno ceno ponornega vozlišča in senčno ceno izvornega vozlišča. V tabeli so prikazane senčne cene transportirane enote vzdolž vsake povezave. Izračunane so kot vsota cene transporta ene enote vzdolž povezave in senčne cene omejitve s kapaciteto določenega vozlišča.

| povezava | $x$ | $y$ | $z$ |
|---|---|---|---|
| 1-3 | 10 – 0 = 10 | 12 – 2 =10 | 14 – 7 = 7 |
| 1-4 | 7 – 0 = 7 | 9 – 2 = 7 | 21 – 7 = 14 |
| 1-5 | 19 – 0 = 19 | 21 – 2 =19 | 23 – 7 = 16 |
| 1-6 | 24 – 0 = 24 | 26 – 2 = 24 | 5 -7 = -12 |
| 2-3 | 10 – 0 = 10 | 12 – 0 = 12 | 14 – 5 = 9 |
| 2-4 | 7 – 0 = 7 | 9 – 0 = 9 | 21 – 5 = 16 |
| 2-5 | 19 – 0 = 19 | 21 – 0 = 21 | 23 – 5 = 18 |
| 2-6 | 24 – 0 = 24 | 26 – 0 = 26 | 5 – 5 = 0 |
| 3-4 | 7 – 10 = -3 | 9 – 12 = -3 | 21 – 14 = 7 |
| 3-5 | 19 – 10 = 9 | 21 – 12 = 9 | 23 – 14 = 9 |
| 3-6 | 24 – 10 = 14 | 26 – 12 = 14 | 5 – 23 = - 18 |
| 4-5 | 19 – 7 = 12 | 21 – 9 = 12 | 23 – 21 = 2 |
| 4-6 | 24 – 7 = 17 | 26 – 9 = 17 | 5 – 21 = -16 |

Pri povezavah, ki so obarvane je senčna cena vozlišča enaka vsoti senčne cene predhodnega vozlišča in cene transporta ene enote po tej povezavi. Vzdolž teh povezav, ki so obarvane, hipotetični pošiljatelj doseže točko preloma[23], kjer so potencialni stroški transporta enaki potencialni vrednosti transporta.

[23] break even point

| povezava | $x$ | $y$ | $z$ |
|---|---|---|---|
| 1-3 | 10 | 10 | 10 |
| 1-4 | 7 | 7 | 7 |
| 1-5 | 22 | 22 | 22 |
| 1-6 | 25 | 25 | 25 |
| 2-3 | 12 | 12 | 12 |
| 2-4 | 9 | 9 | 9 |
| 2-5 | 20 | 20 | 20 |
| 2-6 | 27 | 27 | 27 |
| 3-4 | 14 + 0 = 14 | 14 + 0 = 14 | 14 + 0 = 14 |
| 3-5 | 20 + 0 = 20 | 20 + 0 = 20 | 20 + 0 = 20 |
| 3-6 | 22 + 0 = 22 | 22 + 0 = 22 | 22 + 0 = 22 |
| 4-5 | 6 + 5 = 11 | 6 + 5 = 11 | 6 + 5 = 11 |
| 4-6 | 11 + 5 = 16 | 11 + 5 = 16 | 11 + 5 = 16 |

Če podrobneje pogledamo celice, ki niso obarvane opazimo, da je povečanje cene zaradi senčnih cen manjše od stroškov prevoza na enoto in spremenjenih stroškov zaradi kapacitivnih pogojev, ki smo jih dodali. Če bi želeli poslati pošiljko po teh povezavah, bi torej imeli izgubo na pošiljko enote blaga.

### 1.6.2 Problem kombiniranja ujemajočih se parov pretvorjen na problem maksimalnega pretoka.

**Primer 8:**

Primer povzet po Kuosmanen, Cherchye & Sipilaäinen (b. d.)

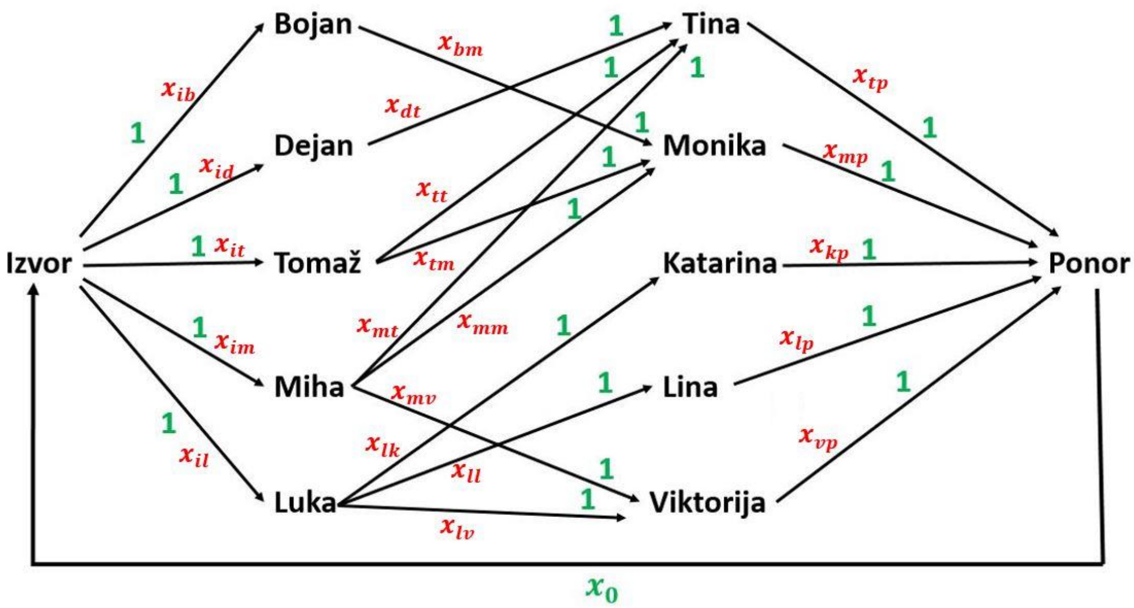
V podjetju je 10 sodelavcev. V spodnji tabeli so označene kompatibilnosti med njimi, določene na osnovi znanj in sposobnosti, ki jih premore vsakdo od njih. Z namenom doseganja čim višje produktivnosti in zadovoljstva zaposlenih na delu, jih mora delodajalec kombinirati tako, da bo čim bolj zadoščeno podatkom iz tabele. Narišite omrežje, ki bo predstavljeni problem pretvorilo na problem maksimalnega pretoka (maksimalno število kompatibilnih ujemanj je enako maksimalnemu pretoku).

| | Tina | Monika | Katarina | Lina | Viktorija |
|---|---|---|---|---|---|
| Bojan | / | C | / | / | / |
| Dejan | C | / | / | / | / |
| Tomaž | C | C | / | / | / |
| Miha | C | C | / | / | C |
| Luka | / | / | C | C | C |

Za zapis povezav uvedimo odločitveno spremenljivko $x$.

$$x_{ij} = povezava\ med\ vozlišči\ grafa$$

Slika 33: Omrežje parov




Povezavam priredimo vrednosti 1, saj lahko medsebojno povežemo le dva zaposlena hkrati. Maksimalni pretok je maksimalno število kompatibilnih parov, ki jih delodajalec uspe medsebojno povezati.

Z $x_0$ označimo vsoto kompatibilnih parov med sodelavci. Na osnovi slike 141 zapišemo naslednji linearni program:

$$\max f = x_0$$

Omejitve s kapaciteto:

$$0 \leq x_{ib} \leq 1$$
$$0 \leq x_{id} \leq 1$$
$$0 \leq x_{it} \leq 1$$
$$0 \leq x_{im} \leq 1$$
$$0 \leq x_{il} \leq 1$$
$$0 \leq x_{bm} \leq 1$$
$$0 \leq x_{dt} \leq 1$$
$$0 \leq x_{tt} \leq 1$$
$$0 \leq x_{tm} \leq 1$$
$$0 \leq x_{mt} \leq 1$$
$$0 \leq x_{mm} \leq 1$$
$$0 \leq x_{mv} \leq 1$$
$$0 \leq x_{lk} \leq 1$$
$$0 \leq x_{ll} \leq 1$$
$$0 \leq x_{lv} \leq 1$$
$$0 \leq x_{tp} \leq 1$$
$$0 \leq x_{mp} \leq 1$$
$$0 \leq x_{kp} \leq 1$$
$$0 \leq x_{lp} \leq 1$$
$$0 \leq x_{vp} \leq 1$$

Omejitve glede ravnotežja tokov:

$$x_o = x_{ib} + x_{id} + x_{it} + x_{im} + x_{il} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Izvor)$$
$$x_{ib} = x_{bm} \ (Ravnotežje\ tokov\ v\ vozlišču\ Bojan)$$
$$x_{id} = x_{dt} \ (Ravnotežje\ tokov\ v\ vozlišču\ Dejan)$$
$$x_{it} = x_{tt} + x_{tm} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Tomaž)$$
$$x_{im} = x_{mt} + x_{mm} + x_{mv} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Miha)$$
$$x_{il} = +x_{lk} + x_{ll} + x_{lv} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Luka)$$
$$x_{dt} + x_{tt} + x_{mt} = x_{tp} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Tina)$$
$$x_{bm} + x_{tm} + x_{mm} = x_{mp} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Monika)$$
$$x_{lk} = x_{kp} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Katarina)$$

$$x_{ll} = x_{lp} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Lina)$$

$$x_{mv} + x_{lv} = x_{vp} \quad (Ravnotežje\ tokov\ v\ vozlišču\ Viktorija)$$

$$x_{tp} + x_{mp} + x_{kp} + x_{lp} + x_{vp} = x_o \quad (Ravnotežje\ tokov\ v\ vozlišču\ Ponor)$$

#### 1.6.2.1 Reševanje problema kombiniranja s programom LINGO

Problem bomo rešili s programskim orodjem LINGO:

```
!Namenska funkcija;
max=x0;

!Omejitve kapacitet povezav;
xib <= 1;  xid <= 1;  xit <= 1;  xim <= 1;  xil <= 1;  xbm <= 1;
xdt <= 1;  xtt <= 1;  xtm <= 1;  xmt <= 1;  xmm <= 1;  xmv <= 1;
xlk <= 1;  xll <= 1;  xlv <= 1;  xtp <= 1;  xmp <= 1;  xkp <= 1;
xlp <= 1;  xvp <=1;

!Ravnotežje tokov;
x0 = xib + xid + xit + xim + xil;
xib = xbm;
xid = xdt;
xit = xtt + xtm;
xim = xmt + xmm + xmv;
xil = xlk + xll + xlv;
xtp = xdt + xtt + xmt;
xmp = xbm + xtm + xmm;
xkp = xlk;
xlp = xll;
xvp = xmv + xlv;
xo = xtp + xmp + xkp + xlp + xvp;
```

**Rešitev**

Poročilo o rešitvi ujemajočih parov v LINGO:

```
  Global optimal solution found.
  Objective value:                              4.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             5
  Elapsed runtime seconds:                          0.05

  Model Class:                                        LP


                                Variable           Value        Reduced Cost
                                      X0        4.000000            0.000000
                                     XIB        1.000000            0.000000
                                     XID        0.000000            0.000000
                                     XIT        1.000000            0.000000
                                     XIM        1.000000            0.000000
                                     XIL        1.000000            0.000000
                                     XBM        1.000000            0.000000
                                     XDT        0.000000            0.000000
                                     XTT        1.000000            0.000000
                                     XTM        0.000000            0.000000
```

| | | |
|---|---|---|
| XMT | 0.000000 | 0.000000 |
| XMM | 0.000000 | 0.000000 |
| XMV | 1.000000 | 0.000000 |
| XLK | 0.000000 | 0.000000 |
| XLL | 1.000000 | 0.000000 |
| XLV | 0.000000 | 0.000000 |
| XTP | 1.000000 | 0.000000 |
| XMP | 1.000000 | 0.000000 |
| XKP | 0.000000 | 0.000000 |
| XLP | 1.000000 | 0.000000 |
| XVP | 1.000000 | 0.000000 |
| XO | 4.000000 | 0.000000 |

Maksimalno število kompatibilnih parov je enako 4. Če se Bojan poveže z Moniko, Tomaž z Tino, Miha z Viktorijo ter Luka z Lino, bodo v podjetju maksimalno izkoristili kompetence delavcev.

## 1.7 Problem asignacije[24]

Problem dodeljevanja oziroma problem asignacije je poseben primer transportnega problema, kjer je cilj minimirati čase ali stroške pri dokončanju določenih nalog z določenim številom izvajalcev. Problem je uporaben predvsem za namene planiranja proizvodnje, poteka storitev in podobno. Gre predvsem za dodeljevanje določenih nalog strojem, dodeljevanje področja prodaje trgovskim potnikom itd.

Pri problemu dodeljevanja obstajajo določene predpostavke:

- število del, ki jih je potrebno opraviti, je enako številu oseb ali strojev, ki lahko opravljajo ta dela;
- vsakemu stroju ali osebi je dodeljeno natanko eno delo;
- vsak stroj ali oseba je sposobna samostojno opravljati katerokoli delo, ki ga je treba opraviti;
- namen problema je specifično določen (ali želimo maksimirati dobiček ali minimizirati stroške dela).

Predpostavimo, da ima podjetje $n$ delavcev, ki so na razpolago za opravljanje dela. Obenem ima podjetje enako število raznovrstnih del, ki jih je treba opraviti. En

[24] Assignment Problem

delavec lahko opravlja samo eno delo. Namen problema je minimirati skupne stroške opravljenega dela. Stroškovna tabela je podana spodaj.

Delo

| Delavci | $j_1$ | $j_2$ | ... | $j_n$ | $a$ |
|---|---|---|---|---|---|
| $i_1$ | $x_{11}$ <br> $c_{11}$ | $x_{12}$ <br> $c_{12}$ | ... | $x_{1n}$ <br> $c_{1n}$ | 1 |
| $i_2$ | $x_{21}$ <br> $c_{21}$ | $x_{22}$ <br> $c_{22}$ | ... | $x_{2n}$ <br> $c_{2n}$ | 1 |
| ... | ... | ... | ... | ... | ... |
| $i_n$ | $x_{n1}$ <br> $c_{n1}$ | $x_{n2}$ <br> $c_{n2}$ | ... | $x_{nn}$ <br> $c_{nn}$ | 1 |
| $b_j$ | 1 | 1 | ... | 1 | |

Pri matematičnem zapisu problema dodeljevanja definiramo odločitveno spremenljivko kot:

$$x_{ij} = \begin{cases} 1; \; \textit{če je delo dodeljeno delavcu } i \\ 0; \; \textit{sicer} \end{cases}$$

$i = 1, 2, \ldots, n$ in $j = 1, 2, \ldots, n$

V zgornji tabeli predstavlja parameter $c_{ij}$ strošek izvedbe, $j$ - tega dela s strani, $i$-tega delavca.

**Minimum namenske funkcije:**

$$\min f(x_{ij}) = c_{11}x_{11} + c_{12}x_{12} + \cdots + c_{nn}x_{nn}$$

**Omejitve:**

$$x_{i1} + x_{i2} + \ldots\ldots\ldots + x_{in} = 1 \qquad i = 1,2,\ldots,n$$

$$x_{1j} + x_{2j} + \ldots\ldots\ldots + x_{nj} = 1 \qquad j = 1, 2, \ldots, n$$

$$x_{ij} \in \{1,0\}$$

Poglejmo problem dodeljevanja na konkretnem primeru, ki ga bomo rešili na dva načina. Najprej poglejmo problem dodeljevanja kot problem pretoka.

**Primer 9:**

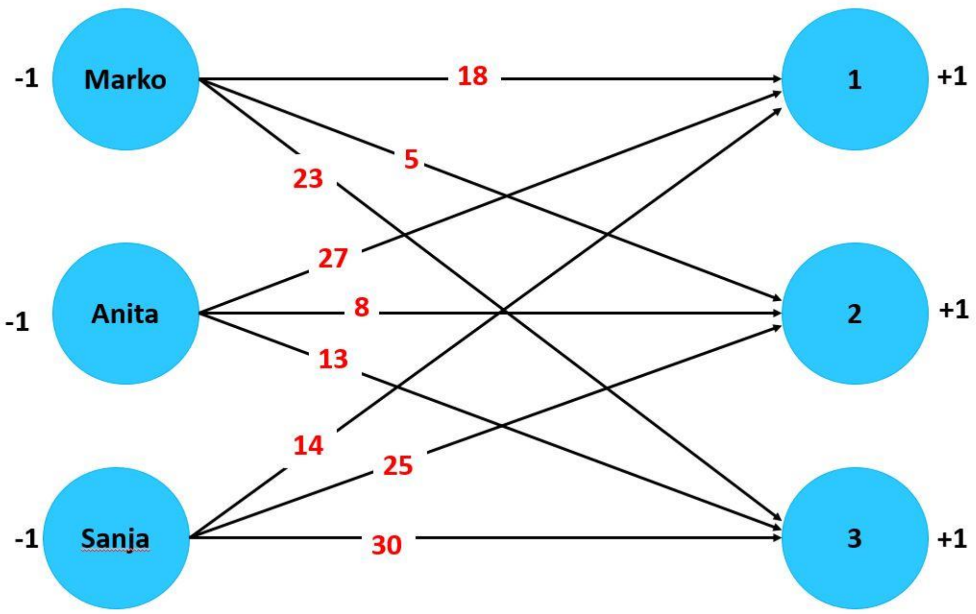
Na voljo so trije delavci in tri naloge, ki morajo biti opravljene. Delavcem mora biti delo dodeljeno tako, da bo delo najhitreje opravljeno. V spodnji tabeli so predstavljeni podatki o času, ki ga določen delavec potrebuje, da opravi določeno delo. Poiskati je treba tako kombinacijo delavcev in dela, da bo ta čas najkrajši.

| | Zaporedno št. dela | | |
|---|---|---|---|
| Delavec | **1** | **2** | **3** |
| Marko | 18 | 5 | 23 |
| Anita | 27 | 8 | 13 |
| Sanja | 14 | 25 | 30 |

Če želimo problem rešiti na podoben način kot omrežne probleme, najprej narišemo omrežje. V tem primeru dodelimo kapacitete vozlišč na izvorna vozlišča in ponorna vozlišča (ponudba in povpraševanje), ki sta v tem primeru enaki 1 enoti (eno delo za enega delavca ter en delavec za eno delo). Vsak delavec namreč potrebuje natanko 1 nalogo ter 1 naloga potrebuje natanko 1 delavca, da jo opravi.

Povpraševanje na vozlišču predstavimo s pozitivnim predznakom, ponudbo pa z negativnim. Tako je omrežje za dani primer predstavljeno na sliki 34.

Slika 34: Problem dodeljevanja dela delavcu




Marko potrebuje 18 ur, da opravi delo pod zaporedno številko 1; 5 ur, da opravi delo pod zaporedno številko 2 in 23 ur, da dokonča delo pod številko 3. Anita potrebuje 27 ur za delo št. 1; 8 ur za delo pod zaporedno številko 2 in 13 ur za delo pod številom 3. Sanja opravi delo pod zaporednim številom 1 v 14 urah, delo pod številko 2 v 25 urah ter delo pod številko 3 v 30 urah. Poskusimo dodeliti enega delavca, Marka, Anito ter Sanjo tako, da bodo dela 1, 2 in 3 najhitreje opravljena. Pri tem upoštevajmo delavčeve časovne omejitve.

Namenska funkcija bo minimirala čas, ki ga delavci potrebujejo, da bodo opravljena vsa dela:

$$\min f(x_{ij}) = 18x_{M1} + 5x_{M2} + 23x_{M3} + 27x_{A1} + 8x_{A2} + 13x_{A3} + 14x_{S1} + 25x_{S2} + 30x_{S3}$$

Omejitve problema so določene za vsako vozlišče in so odvisne od povpraševanja in ponudbe. V danem primeru je 6 vozlišč, od tega so 3 vozlišča izvorna ter 3 vozlišča končna. Preverimo, ali je povpraševanje enako ponudbi. V tem primeru je enako, zato lahko omejitve zapišemo po principu: $[tok\ v\ i] - [tok\ iz\ i] = b_i$.

**Omejitve** za vsako vozlišče posebej:

Vozlišče Marko: $0 - x_{M1} - x_{M2} - x_{M3} = -1$ ali $x_{M1} + x_{M2} + x_{M3} = 1$

Vozlišče Anita: $0 - x_{A1} - x_{A2} - x_{A3} = -1$ ali $x_{A1} + x_{A2} + x_{A3} = 1$

Vozlišče Sanja: $0 - x_{S1} - x_{S2} - x_{S3} = -1$ ali $x_{S1} + x_{S2} + x_{S3} = 1$

Vozlišče 1: $x_{M1} + x_{A1} + x_{S1} = 1$

Vozlišče 2: $x_{M2} + x_{A2} + x_{S2} = 1$

Vozlišče 3: $x_{M3} + x_{A3} + x_{S3} = 1$

**Pogoj nenegativnosti:** $x_{ij} \geq 0$

### 1.7.1 Reševanje problema asignacije s programom Excel

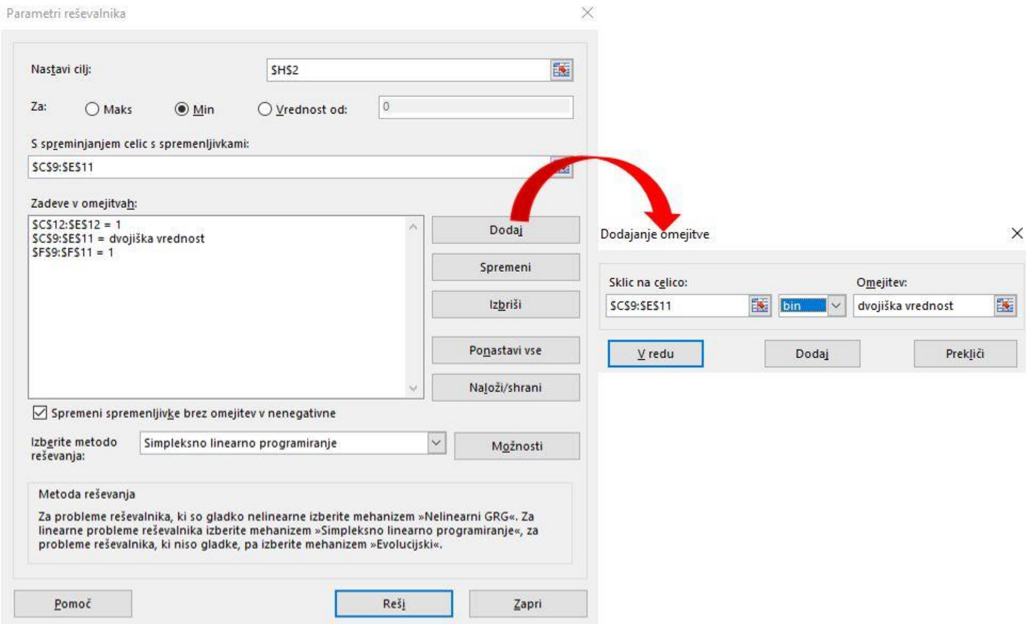
V delovni zvezek programa Excel najprej vnesemo podatke o času, ki ga posamezni delavci porabijo za posamezno delo. Pripravimo še eno identično tabelo, kjer pustimo prazne celice za izpis optimalne rešitve problema. Vsak delavec lahko opravi le eno delo in vsako delo lahko opravi le 1 delavec, zato pripravimo formule za izračun posameznih vrstic in stolpcev. Namenska funkcija je produkt matrike odločitvenih spremenljivk in matrike za delo porabljenega časa.

Slika 35: Priprava podatkov v programu Excel

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Zaporedno št. dela | | | | | |
| 2 | | | 1 | 2 | 3 | | Namenska funkcija | =SUMPRODUCT(C3:E5;C9:E11) |
| 3 | | Marko | 18 | 5 | 23 | | | |
| 4 | | Anita | 27 | 8 | 13 | | | |
| 5 | | Sanja | 14 | 25 | 30 | | | |
| 6 | | | | | | | | |
| 7 | | | Zaporedno št. dela | | | | | |
| 8 | | | 1 | 2 | 3 | Skupaj | | |
| 9 | | Marko | | | | =SUM(C9:E9) | | |
| 10 | | Anita | | | | =SUM(C10:E10) | | |
| 11 | | Sanja | | | | =SUM(C11:E11) | | |
| 12 | | Skupaj | =SUM(C9:C11) | =SUM(D9:D11) | =SUM(E9:E11) | | | |
| 13 | | | | | | | | |

Vnesemo še vse potrebne parametre v Reševalnik. Namensko funkcijo minimiziramo. Vsota po posameznih vrsticah in stolpcih mora biti enaka 1. Odločitvene spremenljivke lahko zasedejo le vrednosti 0 ali 1, kar upoštevamo pri vnašanju omejitev. Uporabimo Simpleksno linearno programiranje kot metodo reševanja problema.

Slika 36: Priprava parametrov v Reševalniku




Rešitev, ki jo izpiše Excel, predstavljamo na sliki 37. V tabeli z odločitvenimi spremenljivkami so podane vrednosti vsake odločitvene spremenljivke tako da je vrednost namenske funkcije minimalna. Delavci porabijo skupaj 32 ur dela, da opravijo vse dodeljene naloge. Pri tem Snja opravi delo 1, Marko delo 2 in Anita delo 3.

Slika 37: Rešitev problema asignacije

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Zaporedno št. dela | | | | | |
| 2 | | | 1 | 2 | 3 | | Namenska funkcija | 32 |
| 3 | | Marko | 18 | 5 | 23 | | | |
| 4 | | Anita | 27 | 8 | 13 | | | |
| 5 | | Sanja | 14 | 25 | 30 | | | |
| 6 | | | | | | | | |
| 7 | | | Zaporedno št. dela | | | | | |
| 8 | | | 1 | 2 | 3 | Skupaj | | |
| 9 | | Marko | 0 | 1 | 0 | 1 | | |
| 10 | | Anita | 0 | 0 | 1 | 1 | | |
| 11 | | Sanja | 1 | 0 | 0 | 1 | | |
| 12 | | Skupaj | 1 | 1 | 1 | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | |

### 1.7.2 Uvoz podatkov iz Excela v LINGO s funkcijo OLE

Velikokrat se zgodi, da imamo določene podatke zapisane v Excelovem delovnem zvezku. Za lažjo in hitrejšo uporabo teh podatkov v LINGO programu lahko uporabimo funkcijo OLE.

OLE funkcija mogoča povezavo med Excelom in LINGOM. S to funkcijo program LINGO bere podatke, shranjene v programu Excel in sicer za namene reševanja problema v LINGO, hkrati pa omogoča tudi izvoz podatkov iz programa LINGO nazaj v Excel. Uporabnost funkcije si poglejmo na prejšnjem primeru. Najprej definirajmo odločitvene spremenljivke ter ostale koeficiente v programu LINGO.

Sintaksa je sledeča:

```
MODEL:
SETS:
Delo/P1,P2,P3/:DEM;
Delavci/Marko,Anita,Sanja/:CAP;
LINKS(Delavci,Delo):ure,dodeli;
ENDSETS
MIN=@SUM(LINKS:ure*dodeli);
@FOR(Delavci(J):
@SUM(Delo(I):dodeli(I,J))=DEM(J));
@FOR(Delo(I):
@SUM(Delavci(J):dodeli(I,J))=CAP(I));
DATA:
CAP,DEM,ure=@OLE('\\fs1\FolderRedirectShare$\ivana.radic1\
Desktop\uvozvlingo.xls', 'Caps','povp','cena');
ENDDATA
END
```

Vozlišča so definirana kot Delavci in Delo. Z *LINKS* funkcijo ustvarimo povezave med vozlišči Delavci in Delo. Koeficienti, shranjeni pod spremenljivko *ure* so določeni v tabeli in predstavljajo ure, ki jih določen delavec porabi, da opravi določeno delo. Spremenljivka »*dodeli*« pa predstavlja odločitvene spremenljivke.

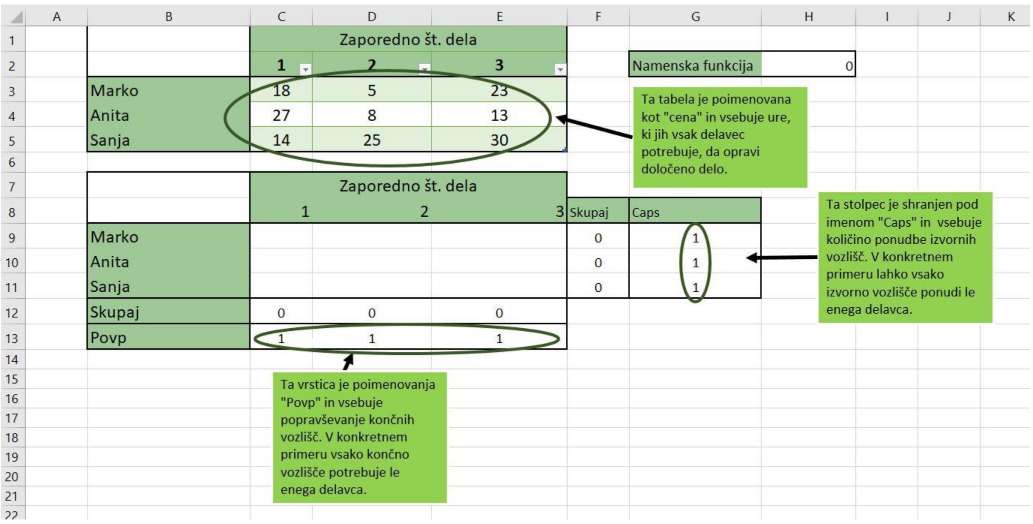
Določimo omejitve sistema in sicer, da je enemu delavcu lahko dodeljeno samo eno delo. V programu to zapišemo kot vsoto vseh prihodov v vozlišča. Vsota vhodov v posameznem vozlišču *Delo* mora biti natanko 1 in vsota vseh izhodov iz vozlišč *Delavci* mora biti prav tako natanko 1.

Funkcija @OLE predstavlja vmesnik med Excelom in Lingom in je oblike @OLE(). Na prvem mestu v funkciji določimo lokacijo Excelove datoteke, na naslednjih pozicijah pa definiramo, katere podatke naj prebere iz datoteke.

V tem primeru smo podatke v Excelu poimenovali »*Caps*«, »*povp*« in »*cena*«. *Caps* so vrednosti izvornih vozlišč (vozlišča Delavci), *povp* predstavlja vrednosti na končnih vozliščih (vozlišča Delavci).

Poglejmo si zapis teh podatkov v Excelu:

Slika 38: Zapis podatkov v Excelu




Vrstica

***CAP,DEM,ure=@OLE('\\fs1\FolderRedirectShare$\ivana.radic1\Desktop\uvozvlingo.xls', 'Caps','povp','cena');***

poda ukaz, da LINGO odpre Excelov dokument in iz njega prebere podatke za »*Caps*«, »*povp*« in »*cena*«, ki so shranjeni pod temi imeni v Excelu.

Poročilo o rešitvi v LINGO:

```
Global optimal solution found.
  Objective value:                              32.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             3
  Elapsed runtime seconds:                          0.83

  Model Class:                                        LP


                                Variable        Value      Reduced Cost
                                 DEM(P1)    1.00000000         0.000000
                                 DEM(P2)    1.00000000         0.000000
                                 DEM(P3)    1.00000000         0.000000
                              CAP(MARKO)    1.00000000         0.000000
                              CAP(ANITA)    1.00000000         0.000000
                              CAP(SANJA)    1.00000000         0.000000
                          URE(MARKO, P1)    18.0000000         0.000000
                          URE(MARKO, P2)    5.00000000         0.000000
                          URE(MARKO, P3)    23.0000000         0.000000
                          URE(ANITA, P1)    27.0000000         0.000000
                          URE(ANITA, P2)    8.00000000         0.000000
                          URE(ANITA, P3)    13.0000000         0.000000
                          URE(SANJA, P1)    14.0000000         0.000000
                          URE(SANJA, P2)    25.0000000         0.000000
                          URE(SANJA, P3)    30.0000000         0.000000
                       DODELI(MARKO, P1)    0.00000000         18.00000
                       DODELI(MARKO, P2)    1.00000000         0.000000
                       DODELI(MARKO, P3)    0.00000000         10.00000
                       DODELI(ANITA, P1)    0.00000000         27.00000
                       DODELI(ANITA, P2)    0.00000000         3.000000
                       DODELI(ANITA, P3)    1.00000000         0.000000
                       DODELI(SANJA, P1)    1.00000000         0.000000
                       DODELI(SANJA, P2)    0.00000000         6.000000
                       DODELI(SANJA, P3)    0.00000000         3.000000


                                     Row    Slack or Surplus    Dual Price
                                       1      32.00000         -1.000000
                                       2      0.000000          0.000000
                                       3      0.000000         -5.000000
                                       4      0.000000         -13.00000
                                       5      0.000000           0.00000
                                       6      0.000000           0.00000
                                       7      0.000000         -14.00000
```

Dobimo enak rezultat kot pri reševanju problema s programom Excel. Vrednost namenske funkcije je 32 ur. Pod spremenljivkami »DODELI (IME DELAVCA, OPRAVILO)« preberemo rešitev, kjer vidimo da je program dodelil Marku delo P2, Aniti delo P3 ter Sanji delo P2.

# 2 ANALIZA OVOJNICE PODATKOV - DEA

Analiza ovojnice podatkov[25] ali krajše DEA, je metoda merjenja učinkovitosti, ki so jo leta 1978 predstavili Charnes, Cooper in Rhodes. DEA je nestatistična in neparametrična metoda za ugotavljanje relativnih učinkovitosti posameznih odločitvenih enot, ki jih vključimo v analizo. Te enote poimenujemo posamezne odločitvene enote ali krajše DMU[26].

Analiza se največkrat uporablja za primerjavo med institucijami, zato lahko kot primere takih DMU navedemo: poslovalnice neke banke, policijske postaje, bolnice, davčni uradi, zapori, vojaške baze, šole, univerze in podobno.

Posebna prednost DEA analize je, da se lahko uporabi tudi v organizacijah, kjer dobiček ni edino merilo učinkovitosti. V letih od predstavitve metode do danes je bilo opravljenega veliko teoretičnega in praktičnega dela, da je metoda dobila svojo današnjo obliko. O njej je bilo napisanih veliko učbenikov in knjig, npr. Ramanathan (2003) in Beasley (b. d.). V tem poglavju smo se pri razlagi pojmov oprli na definicije in oznake, ki jih v svojih zapiskih navaja Beasley (b. d.).

Metoda deluje na konceptu ovojnice okrog izbranih enot DMU. Ovojnica se oblikuje na podlagi najboljših DMU, mere učinkovitosti ostalih pa se nato izračunajo relativno glede na oblikovano ovojnico (Ramanathan, 2003).

Relativno učinkovitost DMU izračunamo na podlagi količnika med vhodnimi (inputi) in izhodnimi podatki (outputi). Osnovna predpostavka metode je: če nek DMU, ki ga poimenujemo A izvede $X_2(A)$ enot outputov z $X_1(A)$ inputi, potem moramo tudi ostale DMU-je v vzorcu primerjati na isti način. Torej koliko outputov $X_2(B)$ izvede podjetje z $X_1(B)$ inputi. Ta predpostavka omogoča, da lahko enote DMU v vzorcu učinkovito primerjamo med seboj (DEA, b. d.).

[25] Data Envelopment Analysis
[26] decision-making units

V osnovi poznamo dve razdelitvi DEA modelov:

- Modeli, naravnani na vhodne podatke (v literaturi večkrat zasledimo izraz BCC[27])
    - ti modeli preverjajo, ali lahko preučevana DMU zmanjša svoje trenutne vhodne resurse in še vedno proizvaja isto število izhodov;
- modeli, naravnani na izhodne podatke (v literaturi večkrat zasledimo izraz CCR[28])
    - ti modeli preverjajo, ali lahko preučevana DMU poveča svoj trenutni izhod ob nespremenjenem številu vhodnih resursov.

Poglejmo si preprost izračun relativne učinkovitosti na podlagi štirih transportnih podjetij in samo dveh podatkov.

**Primer 10:**

SERVISNA PODJETJA I.

Med seboj bomo primerjali učinkovitost štirih servisnih podjetij (ta predstavljajo DMU- je v danem primeru). Za vsako od njih imamo po en izhodni podatek (število zasebnih popravil) in en vhodni podatek (število zaposlenih). Podatki so zbrani v spodnji tabeli.

| | Število popravil | Število zaposlenih |
|---|---|---|
| Medved | 125 | 18 |
| Granit | 44 | 16 |
| Slapar | 80 | 17 |
| Plamen | 23 | 11 |

Iz podatkov lahko razberemo, da podjetje Granit s 16 zaposlenimi opravi 44 popravil na mesec, medtem ko podjetje Slapar s samo enim zaposlenim več, opravi kar 80 popravil na mesec. Sklepamo lahko, da je podjetje Slapar bolj

[27] Banker, Charnes and Cooper model – poimenovan po avtorjih modela
[28] Charnes, Cooper, Rhodes model

učinkovito kot podjetje Granit. DEA analiza pa nam omogoča, da lahko to tudi računsko preverimo.

Prvi korak pri izračunu relativne učinkovitosti podjetij je, da izračunamo razmerje (količnik) vhodnega in izhodnega podatka za vsako podjetje posebej.

$$količnik = \frac{izhodni\ podatek}{vhodni\ podatek}$$

| | Izračun količnika | Število popravil na zaposlenega |
|---|---|---|
| Medved | $\frac{125}{18}$ | 6,9 |
| Granit | $\frac{44}{16}$ | 2,75 |
| Slapar | $\frac{80}{17}$ | 4,71 |
| Plamen | $\frac{23}{11}$ | 2,09 |

Ko izračunamo količnik, preverimo katera izmed DMU ima najvišji količnik, saj bo ta DMU služila kot osnova za nadaljnje izračune. V danem primeru ima najvišji količnik popravil glede na število zaposlenih podjetje Medved. Relativno učinkovitost ostalih podjetij izračunamo na podlagi ugotovljenega največjega količnika.

| | Relativni količnik | Relativna učinkovitost [%] |
|---|---|---|
| Medved | $\frac{6,9}{6,9} \cdot 100$ | 100 |
| Granit | $\frac{2,75}{6,9} \cdot 100$ | 39,86 |
| Slapar | $\frac{4,71}{6,9} \cdot 100$ | 68,26 |
| Plamen | $\frac{2,09}{6,9} \cdot 100$ | 30,29 |

Največjo relativno učinkovitost ima podjetje Medved, najmanjše pa podjetje Plamen. Če bi se želela podjetja Granit, Slapar in Plamen izboljšati v primerjavi s podjetjem Medved, bi morala spremeniti vhodno ali izhodno vrednost.

Spremembo na bolje lahko dosežejo z zmanjšanjem vhodnega podatka – zmanjšanjem števila zaposlenih, vendar ob predpostavki, da jim število zasebnih popravil ne upade. Lahko pa ob istem številu zaposlenih povečajo število popravil, ki jih opravijo na mesečni ravni.

Seveda pa takšna preprosta analiza največkrat ne pove veliko o učinkovitosti podjetja, saj podjetja ne opravljajo samo ene dejavnosti. V realnem sistemu so sistemi DMU-jev veliko bolj kompleksni in tudi analiza DEA postane kompleksnejša.

## 2.1 Grafični način reševanja analize ovojnice podatkov

**Primer 11:**

SERVISNA PODJETJA II.

Primer servisnih podjetij bomo razširili tako, da bomo dodali še eno dejavnost, s katero se ukvarjajo podjetja. Poleg zasebnih popravil sedaj opravljajo tudi popravila v tovarnah. Tako imamo sedaj en vhodni in dva izhodna podatka.

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

Kakor v prejšnjem primeru najprej izračunamo število popravil na zaposlenega v posameznem podjetju. Ločimo med številom zasebnih popravil in številom popravil v tovarnah.

| | Število zasebnih popravil na zaposlenega | Število popravil v tovarnah na zaposlenega |
|---|---|---|
| Medved | 6,9 | 2,78 |
| Granit | 2,75 | 1,25 |
| Slapar | 4,71 | 3,24 |
| Plamen | 2,09 | 1,09 |

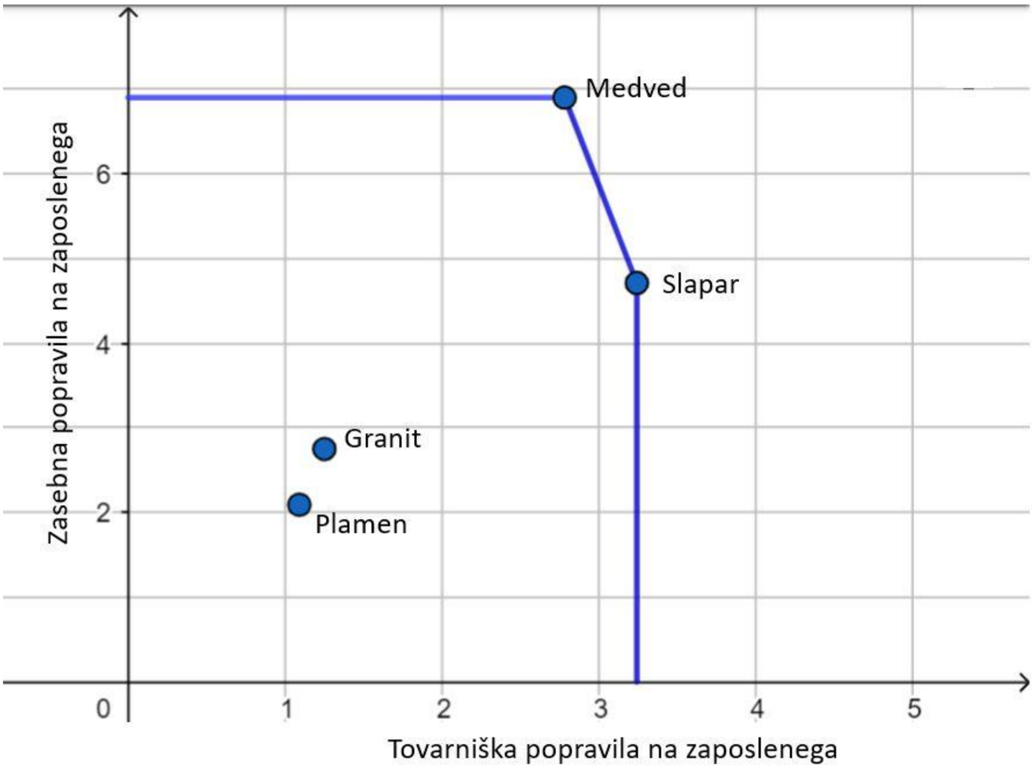
Podatkov o številu zasebnih popravil nismo spreminjali, najvišji količnik ima podjetje Medved, najnižji pa podjetje Plamen. Pri številu popravil v tovarnah pa ima najvišji količnik podjetje Slapar, najnižjega pa podjetje Plamen.

Ostali dve podjetji ne dosegata tako visokih količnikov, torej v primerjavi s podjetjema Medved in Slapar poslujeta slabše. Ker se glede na različna izhodna podatka uspešnost podjetij razlikuje, težko zaključimo, katero podjetje posluje najbolj učinkovito. Ne moremo z gotovostjo trditi, ali je pomembneje, da podjetje dosega visok količnik pri zasebnih popravilih ali pri tovarniških popravilih.

Dober način kako interpretiramo rezultate, kjer se količniki razlikujejo in imamo samo dva vhodna in en izhodni podatek, je grafična analiza količnikov.

Pripravimo si koordinatni sistem, kjer abscisna os predstavlja količnik tovarniških popravil na zaposlenega, ordinatna os pa količnik zasebnih popravil na zaposlenega.

Slika 39: Grafično reševanje DEA analize

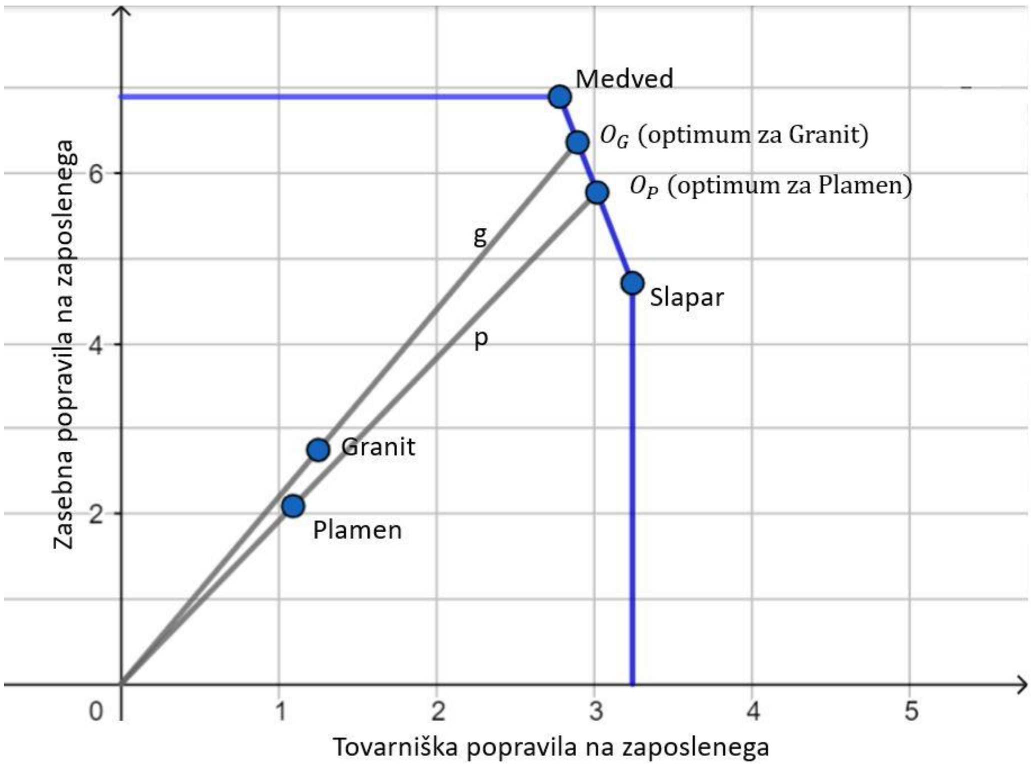



Položaja podjetij Medved in Slapar na zgornji sliki predstavljata nivo učinkovitosti, ki presega učinkovitost ostalih dveh podjetij. Modra črta na grafu tako predstavlja mejo učinkovitosti ali ovoj podatkov (ravno zaradi ovoja, ki nastane pri grafični metodi, je analiza dobila ime). Meja učinkovitosti je območje, ki naj bi ga podjetji Granit in Plamen poskušali doseči s ciljem biti učinkovitejši. Posamično relativno učinkovitost podjetij lahko preprosto izračunamo po formuli, ki smo jo uporabili že v primeru Servisna podjetja I. in dobimo naslednje rezultate:

| | Relativna učinkovitost zasebnih popravil [%] | Relativna učinkovitost tovarniških popravil [%] |
|---|---|---|
| Medved | 100 | 85,80 |
| Granit | 39,86 | 38,58 |
| Slapar | 68,26 | 100 |
| Plamen | 30,29 | 33,64 |

Ker na podlagi posameznih relativnih učinkovitosti težko sklepamo o tem, kako uspešna sta v primerjavi z ostalima dvema podjetjema, moramo izračunati numerično vrednost njune učinkovitosti.

Slika 40: Numerična učinkovitost podjetij Granit in Plamen




Raven učinkovitosti za podjetji Granit in Plamen dobimo tako, da primerjamo oddaljenost trenutne lege teh dveh podjetij na grafu do koordinatnega izhodišča in oddaljenost točke njune optimalne uspešnosti do koordinatnega izhodišča.

Točki njune optimalne uspešnosti sta na grafu predstavljeni s točkama $O_P$ (točka optimalnosti za podjetje Plamen) in $O_G$ (točka optimalnosti za podjetje Granit). Vidimo, da ležita ravno na ovoju podatkov, ki ga tvorita 100 % uspešni podjetji Medved in Slapar.

Najprej izračunamo raven učinkovitosti za podjetje Plamen. Trenutno oddaljenost podjetja od izhodišča izračunamo s formulo za razdaljo med dvema točkama v koordinatnem sistemu:

$$d(A,B) = |AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

kjer so $x_1, x_2, y_1$ in $y_2$ koordinate točk, med katerima računamo razdaljo.

Prva točka, ki jo upoštevamo v izračunu je koordinatno izhodišče s koordinatami $(0{,}0)$, druga točka pa trenutni položaj podjetja. V primeru podjetja Plamen je to točka s koordinatama $\left(\frac{109}{100}, \frac{209}{100}\right)$:

$$d_P = \sqrt{(2{,}09 - 0)^2 + (1{,}09 - 0)^2} = \sqrt{4{,}37 + 1{,}188} = \sqrt{5{,}558} = 2{,}36$$

Da lahko razdaljo podjetja od izhodišča primerjamo z njegovo razdaljo do ovoja podatkov, moramo najprej določiti, kje se nahaja točka optimalnosti za podjetje Plamen. Na grafu vidimo, da je točka presečišče premice $p$, ki poteka čez koordinatno izhodišče in točko, ki predstavlja trenutni položaj podjetja $\left(\frac{109}{100}, \frac{209}{100}\right)$ ter premico $o$, ki poteka čez točki trenutnih položajev podjetij Medved in Slapar.

Da lahko izračunamo presečišče premic, moramo najprej poiskati enačbo premice $o$. Spomnimo, da je osnovna enačba linearne funkcije $y = kx + n$, kjer sta $k$ in $n$ poljubni realni števili.

Postopek iskanja enačbe premice, ki poteka skozi dve točki, za kateri poznamo koordinate dobimo tako, da najprej izračunamo smerni koeficient premice $k$. Smerni koeficient izračunamo tako, da koordinati poznanih točk vstavimo v enačbo

$$k = \frac{y_2 - y_1}{x_2 - x_1}$$

Iščemo enačbo premice $o$, ki poteka skozi legi podjetij Medved in Slapar, zato upoštevamo koordinate za podjetje Medved $(\frac{278}{100}, \frac{697}{100})$ in za podjetje Slapar $(\frac{324}{100}, \frac{471}{100})$:

$$k = \frac{4{,}71 - 6{,}97}{3{,}24 - 2{,}78} = \frac{-2{,}26}{0{,}46} = -4{,}91$$

Število $n$ iz prej zapisane enačbe določimo s pomočjo enačbe $y = k(x - x_1) + y_1$, kjer za $k$ uporabimo predhodno izračunano vrednost smernega koeficienta ter koordinati ene izmed točk.

$$y = -4{,}91(x - 2{,}78) + 6{,}97$$
$$y = -4{,}91x + 20{,}62$$

Enačba premice, ki predstavlja ovoj podatkov je torej $y = -4{,}91x + 20{,}62$. Na enak način moramo sedaj določiti še enačbo premice $p$, ki poteka skozi koordinatno izhodišče $(0{,}0)$ in trenutno lego podjetja Plamen $\frac{109}{100}, \frac{209}{100})$:

$$k = \frac{2{,}79 - 0}{1{,}09 - 0} = \frac{2{,}09}{1{,}09} = 1{,}92$$

Premica $p$ poteka skozi koordinatno izhodišče, torej je njena začetna vrednost enaka 0. Njen funkcijski predpis je tako

$$y \; = \; 1{,}92x.$$

Točka $O_P$, ki predstavlja optimalno lego podjetja Plamen, leži na presečišču premic, $p$ in $o$, katerih enačbe smo pravkar določili. Koordinati presečišča izračunamo in dobimo optimalno lego podjetja Plamen, ki je točka $O_P\ (\frac{302}{100}, \frac{58}{10})$. Sedaj lahko izračunamo še njeno oddaljenost od koordinatnega izhodišča, na isti način, kot smo prej izračunali oddaljenost trenutne lege do izhodišča:

$$d_{OP} = \sqrt{(3{,}02 - 0)^2 + (5{,}80 - 0)^2} = \sqrt{9{,}12 + 33{,}64} = \sqrt{42{,}76} = 6{,}54$$

Ko imamo izračunani razdalji obeh točk s koordinatnim izhodiščem, lahko izračunamo njun količnik. Trenutna lega podjetja Plamen je od koordinatnega izhodišča oddaljena 2,36, optimalna lega pa 6,54.

$$\frac{d_P}{d_{OP}} = \frac{2{,}36}{6{,}54} = 0{,}3609$$
$$0{,}3609 \cdot 100 = 36{,}09\%$$

Trenutna raven uspešnosti podjetja Plamen je torej 36,09 % napram uspešnejšima podjetjema Medved in Slapar.

Po enakem postopku izračunamo trenutno raven podjetja Granit. Najprej izračunamo oddaljenost točke, ki predstavlja njegovo trenutno lego $(\frac{125}{100}, \frac{275}{100})$ od koordinatnega izhodišča:

$$d_G = \sqrt{(1{,}25 - 0)^2 + (2{,}75 - 0)^2} = \sqrt{1{,}56 + 7{,}56} = \sqrt{9{,}12} = 3{,}02$$

Da bi poiskali lego točke optimalne uspešnosti za podjetje Granit, moramo najprej določiti enačbo premice $g$, ki poteka skozi koordinatno izhodišče in trenutno lego podjetja Granit:

$$k = \frac{2{,}75 - 0}{1{,}25 - 0} = \frac{2{,}75}{1{,}25} = 2{,}2$$

Ker premica poteka skozi koordinatno izhodišče, je njen funkcijski predpis enak:

$$y = 2{,}2x$$

Poiščemo presečišče premic $g$ in $o$, kjer leži točka optimalne uspešnosti:

$$2{,}2x = -4{,}91x + 20{,}62$$
$$7{,}11x = 20{,}62$$
$$x = 2{,}9$$

Koordinato $y$ pa izračunamo tako, da izračunano koordinato $x = 2{,}9$ vstavimo v eno izmed enačb premic:

$$y = 2{,}2x$$
$$y = 2{,}2 \cdot 2{,}9$$
$$y = 6{,}38$$

Točka optimalne uspešnosti podjetja Granit leži na koordinatah $O_G(\frac{29}{10}, \frac{638}{100})$. Izračunati moramo še oddaljenost te točke do izhodišča:

$$d_{OG} = \sqrt{(2{,}9 - 0)^2 + (6{,}38 - 0)^2} = \sqrt{8{,}41 + 40{,}7} = \sqrt{49{,}11} = 7{,}01$$

Ko imamo izračunani razdalji obeh točk s koordinatnim izhodiščem, lahko izračunamo njun količnik. Trenutna lega podjetja Granit je od koordinatnega izhodišča oddaljena 3,02, optimalna lega pa 7,01.

$$\frac{d_G}{d_{OG}} = \frac{3{,}02}{7{,}01} = 0{,}4308$$
$$0{,}4308 \cdot 100 = 43{,}08\ \%$$

Trenutna raven uspešnosti podjetja Granit je torej 43,08 % glede na uspešnejši podjetji Medved in Slapar.

## 2.2 Analiza ovojnice podatkov in linearno programiranje

Primer, ki smo ga obravnavali, je obsegal le en vhodni in dva izhodna podatka, zato smo ga lahko rešili z grafično metodo. V kolikor pa se število vhodnih in/ali izhodnih podatkov poveča, grafično reševanje ni več mogoče. V nadaljevanju bomo predstavili kako DEA analizo razrešimo s pomočjo linearnega programa.

Ker je izhodno usmerjen pristop na splošno bolj primeren za oceno zmogljivosti in izkoriščenosti zmogljivosti, se bomo v nadaljevanju posvetili temu modelu. V izhodno usmerjenem modelu lahko razberemo, za koliko se spremeni relativna učinkovitost DMU, ki je ocenjen kot učinkovit, s sorazmernim povečanjem izhodov. Seveda pa pri tem razmerja vhodov ostanejo nespremenjena.

Rešitev izhodno usmerjenega modela je relativna učinkovitost posamezne DMU, vrednost učinkovitosti, ki jo takšen model poda, je na intervalu $[0{,}1]$. Pri tem 1 pomeni najvišjo učinkovitost DMU, manjše vrednosti pomenijo učinkovitost, ki je slabša od 100 %. Slabost izhodno usmerjene DEA analize je v tem, da lahko več podjetij postane 100 % učinkovitih.

### 2.2.1 Izhodno usmerjeni modeli

V nadaljevanju bomo predstavili analizo štirih DMU-jev s pomočjo izhodno usmerjenega modela, ki je lažji za razumevanje in interpretacijo. Še enkrat se bomo lotili problema Servisna podjetja II.

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

Linearni program, ki ga bomo predstavili v nadaljevanju, obravnava problem na nekoliko drugačen način kot smo ga spoznali pri grafičnem reševanju.

Pri DEA analizi, ki jo naredimo s pomočjo linearnega programa uvedemo uteži vhodnih in izhodnih podatkov (označimo jih z $W$). V danem primeru potrebujemo tri uteži, ki jih označimo z $W_Z$, $W_T$ in $W_{Zap}$.

Spremenljivka $W_Z$ naj predstavlja utež zasebnih popravil, $W_T$ utež tovarniških popravil, $W_{Zap}$ pa utež števila zaposlenih.

Definirati je treba učinkovitosti za vsak DMU posebej in sicer kot količnik med uteženo vsoto izhodnih podatkov in uteženo vsoto vhodnih podatkov. Numerično vrednost učinkovitosti pri tem normiramo z vrednostmi med 0 in 1. Pozorni moramo biti, da pri tem uteži izberemo tako, da maksimirajo učinkovitost danega podjetja.

Poglejmo, kako zapisati linearni program, s katerim izračunamo učinkovitost podjetja Granit. Cilj linearnega programa izhodno usmerjenega modela je maksimirati uteženo vsoto izhodov DMU, ki ga analiziramo:

$$\max\ U_g$$

Poglejmo na primeru podjetja Granit, kako zapisati celotno namensko funkcijo. Podjetje ima 44 zasebnih popravil in 20 popravil v tovarnah uteženo vsoto izhodnih podatkov zato zapišemo kot $44W_Z + 20W_T$.

$$\max U_g = 44W_Z + 20W_T$$

Za omejitve poleg utežene vsote izhodnih podatkov potrebujemo še uteženo vsoto vhodov. Zapisali smo namreč, da učinkovitost definiramo kot količnik med uteženo vsoto izhodov in vhodov. V danem primeru imamo en vhodni podatek. Poglejmo, kako sestaviti omejitev na primeru podjetja Granit, ki ima 16 zaposlenih. Utežena vsota izhodnih podatkov je torej $44W_Z + 20W_T$, utežena vsota vhodnih podatkov pa $16W_{Zap}$. Količnik med njima je torej $\frac{(44W_Z + 20W_T)}{16W_{Zap}}$.

Omejitve za ostale DMU-je napišemo na isti način:

$$U_M = \frac{(125W_Z + 50W_T)}{18W_{Zap}}$$

$$U_G = \frac{(44W_Z + 20W_T)}{16W_{Zap}}$$

$$U_S = \frac{(80W_Z + 55W_T)}{17W_{Zap}}$$

$$U_P = \frac{(23W_Z + 12W_T)}{11W_{Zap}}$$

$$0 \leq U_M \leq 1$$

$$0 \leq U_G \leq 1$$

$$0 \leq U_S \leq 1$$

$$0 \leq U_P \leq 1$$

$$W_Z \geq 0$$

$$W_T \geq 0$$

$$W_{Zap} \geq 0$$

kjer so $U_M$, $U_G$, $U_S$ in $U_P$ numerične vrednosti uspešnosti posameznih podjetij.

Predstavljeni problem optimizacije sodi med nelinearne probleme, ki so zahtevni za reševanje. Ker se v tem gradivu posvečamo linearnemu programiranju, bomo predstavljeni nelinearni primer preoblikovali v linearnega.

Zgoraj zapisane izraze v omejitvah oblikujemo v linearne izraze. Kot dodatno omejitev zapišemo, da mora biti utežena vsota vhodov DMU-ja, za katerega smo napisali linearni program, enaka 1. Dodatni pogoj postavimo zaradi lastnosti modela, ki opazuje, za koliko se spremeni učinkovitost podjetja, če spreminjamo izhodne parametre, medtem ko vhodni podatki ostajajo konstantni. S postavitvijo dodatne enačbe tudi poskrbimo, da dobimo linearno obliko programa.

Zapišemo torej:

$$16W_{Zap} = 1$$

Numerične vrednosti uspešnosti podjetij morajo biti izražene z vrednostmi med 0 in 1, zato zapišemo:

$$0 \leq \frac{(125W_Z + 50W_T)}{18W_{Zap}} \leq 1$$

$$0 \leq \frac{(44W_Z + 20W_T)}{16W_{Zap}} \leq 1$$

$$0 \leq \frac{(80W_Z + 55W_T)}{17W_{Zap}} \leq 1$$

$$0 \leq \frac{(23W_Z + 12W_T)}{11W_{Zap}} \leq 1$$

Da pridobimo linearne obliko omejitev, je potrebno vsako izmed omejitev pomnožiti z izrazi iz imenovalcev. V primeru prve omejitve, ki se nanaša na podjetje Medved, je postopek naslednji:

$$0 \leq \frac{(125W_Z + 50W_T)}{18W_{Zap}} \leq 1 \,/ \cdot 18W_{Zap}$$

$$0 \leq (125W_Z + 50W_T) \leq 18W_{Zap}$$

Če natančneje pogledamo desno stran enačbe $(125W_Z + 50W_T) \leq 18W_{Zap}$, vidimo, da je izraz $\left(125W_Z + 50W_T - 18W_{Zap}\right) \leq 0$.

Po preureditvi vseh omejitev so omejitve novega linearnega programa sledeče:

$$\left(125W_Z + 50W_T - 18W_{Zap}\right) \leq 0$$

$$\left(44W_Z + 20W_T - 16W_{Zap}\right) \leq 0.$$

$$\left(80W_Z + 55W_T - 17W_{Zap}\right) \leq 0.$$

$$\left(23W_Z + 12W_T - 11W_{Zap}\right) \leq 0.$$

Zapisati je treba še vse dodatne pogoje. Spomnimo, da uteži ne morejo biti negativno število in da moramo fiksirati uteženo vsoto vhodnih podatkov za opazovano podjetje.

$$W_Z \geq 0$$
$$W_T \geq 0$$
$$W_{Zap} \geq 0$$
$$16W_{Zap} = 1$$

### 2.2.2 Reševanje DEA analize na primeru izhodno usmerjenega modela s programom LINGO

Linearni program vnesemo v program LINGO:

```
!Namenska funkcija;
max = 44*WZ + 20*WT;
!Omejitve;
[Medved] 125*WZ + 50*WT - 18*WZap <= 0;
[Granit] 44*WZ + 20*WT - 16*WZap <= 0;
[Slapar] 80*WZ + 55*WT - 17*WZap <= 0;
[Plamen] 23*WZ + 12*WT - 11*WZap <= 0;
[Zasebna_popravila] WZ >= 0;
[Tovar_popravila] WT >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 16*WZap = 1;
```

Poročilo o rešitvi za DEA analizo (podjetje Granit):

```
Global optimal solution found.
Objective value:                              0.4317391
Infeasibilities:                               0.000000
Total solver iterations:                              2
Elapsed runtime seconds:                           0.05

Model Class:                                         LP


                              Variable           Value        Reduced Cost
                                    WZ    0.3043478E-02        0.000000
                                    WT    0.1489130E-01        0.000000
                                  WZAP    0.6250000E-01        0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        0.4317391        1.000000
                                MEDVED         0.000000       0.2852174
                                GRANIT        0.5682609        0.000000
                                SLAPAR         0.000000       0.1043478
                                PLAMEN        0.4388043        0.000000
                     ZASEBNA_POPRAVILA    0.3043478E-02        0.000000
                       TOVAR_POPRAVILA    0.1489130E-01        0.000000
                             ZAPOSLENI    0.6250000E-01        0.000000
                      DODATNA_OMEJITEV         0.000000       0.4317391
```

**Vrednost namenske funkcije:** v primeru podjetja Granit je vrednost namenske funkcije enaka 0,432. Tako kot pri grafičnem načinu vrednost pomnožimo s 100, da dobimo relativno učinkovitost podjetja v procentih

$$0{,}432 \cdot 100 = 43{,}2\ \%$$

Grafična rešitev se od rešitve LINGO razlikuje za 0,12 %, kar je posledica zaokroževanja števil pri računanju.

Poleg relativne učinkovitosti lahko iz poročila razberemo tudi vrednosti posameznih uteži. Vrednosti uteži vhodnih podatkov odražajo relativno povečanje učinkovitosti DMU-jev v primeru, da ustrezno vhodno vrednost zmanjšamo za eno enoto. V danem primeru imamo samo en vhodni podatek - število zaposlenih $W_{ZAP}$. Vrednosti uteži izhodnih podatkov pa odražajo spremembo učinkovitosti DMU-ja, če spremenimo ustrezno izhodno vrednost. V našem primeru imamo dva izhodna podatka – število popravil v tovarnah $W_T$ in število zasebnih popravil $W_Z$.

V tabeli **Vrednost** razberemo, da je vrednost $W_Z$ enaka 0,00304, kar pomeni, da če bi podjetje izvedlo eno zasebno popravilo več ob istem številu zaposlenih in istem številu tovarniških popravil, bi s tem svojo učinkovitost dvignilo za 0,304%.

Če pa bi opravilo eno tovarniško popravilo več, pa bi učinkovitost dvignilo za 1,489 %.

Cilj vsakega podjetja je, da se izboljšuje. Podatek o tem, za koliko mora izboljšati svojo učinkovitost, da bo njegovo delovanje dejansko učinkovito lahko izračunamo tako, da vrednost namenske funkcije odštejemo od 1:

$$1 - 0{,}432 = 0{,}568$$

Podjetje Granit mora svojo učinkovitost povečati za 56,8 %, da bo učinkovito. To lahko stori tako, da zmanjša število zaposlenih, ob predpostavki, da število izvedenih popravil ostane enako.

Če nas, namesto učinkovitosti podjetja Granit, zanima uspešnost podjetja Medved, v danem linearnem programu spremenimo namensko funkcijo in dodatno omejitev. Spremeniti je treba torej samo vrednosti vhodnih in izhodnih podatkov. Izraz za uspešnost podjetja Medved je naslednji:

Maksimum namenske funkcije je:

$$\max U_M = max(125W_Z + 50W_T).$$

Ostale omejitve ostanejo enake, nov linearni program za izračun relativne uspešnosti podjetja Medved je tako:

$$\max U_M = max(125W_Z + 50W_T).$$

$$\left(125W_Z + 50W_T - 18W_{Zap}\right) \leq 0$$

$$\left(44W_Z + 20W_T - 16W_{Zap}\right) \leq 0.$$

$$\left(80W_Z + 55W_T - 17W_{Zap}\right) \leq 0.$$

$$\left(23W_Z + 12W_T - 11W_{Zap}\right) \leq 0.$$

$$W_Z \geq 0$$

$$W_T \geq 0$$

$$W_{Zap} \geq 0$$

Dodatna omejitev pa je zdaj:

$$18W_{Zap} = 1$$

```
!Namenska funkcija;
max = 125*WZ + 50*WT;
!Omejitve;
[Medved] 125*WZ + 50*WT - 18*WZap <= 0;
[Granit] 44*WZ + 20*WT - 16*WZap <= 0;
[Slapar] 80*WZ + 55*WT - 17*WZap <= 0;
[Plamen] 23*WZ + 12*WT - 11*WZap <= 0;
[Zasebna_popravila] WZ >= 0;
[Tovar_popravila] WT >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 18*WZap = 1;
```

Poročilo o rešitvi za DEA analizo (Medved):

```
Global optimal solution found.
  Objective value:                              1.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             1
  Elapsed runtime seconds:                          0.05

  Model Class:                                        LP

                       Variable           Value        Reduced Cost
                             WZ       0.8000000E-02        0.000000
                             WT        0.000000            0.000000
                           WZAP       0.5555556E-01        0.000000

                            Row    Slack or Surplus      Dual Price
                              1        1.000000            1.000000
                         MEDVED        0.000000            1.000000
                         GRANIT       0.5368889            0.000000
                         SLAPAR       0.3044444            0.000000
                         PLAMEN       0.4271111            0.000000
              ZASEBNA_POPRAVILA       0.8000000E-02        0.000000
                TOVAR_POPRAVILA        0.000000            0.000000
                      ZAPOSLENI       0.5555556E-01        0.000000
               DODATNA_OMEJITEV        0.000000            1.000000
```

**Vrednost namenske funkcije** je tokrat 1. Numerična vrednost učinkovitosti podjetja Medved je tako 100 %, kar smo prej pokazali tudi z grafično metodo.

V tabeli **Vrednost** je vrednost $W_Z$ enaka 0,008, torej bi v primeru izvedbe enega zasebnega popravila več (ob istem številu zaposlenih in istem številu tovarniških popravil) podjetje svojo učinkovitost dvignilo za 0,8 %.

Ker je podjetje že 100% učinkovito, svoje učinkovitosti s povečanjem števila tovarniških popravil ne more dvigniti.

Poglejmo sedaj še LINGO kodo za podjetje Plamen:

```
!Namenska funkcija;
max = 23*WZ + 12*WT;
!Omejitve;
[Medved] 125*WZ + 50*WT - 18*WZap <= 0;
[Granit] 44*WZ + 20*WT - 16*WZap <= 0;
[Slapar] 80*WZ + 55*WT - 17*WZap <= 0;
[Plamen] 23*WZ + 12*WT - 11*WZap <= 0;
[Zasebna_popravila] WZ >= 0;
[Tovar_popravila] WT >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 11*WZap = 1;
```

Poročilo o rešitvi za DEA analizo (Plamen):

```
  Global optimal solution found.
  Objective value:                              0.3617391
  Infeasibilities:                               0.000000
  Total solver iterations:                              2
  Elapsed runtime seconds:                           0.05

  Model Class:                                         LP


                          Variable           Value        Reduced Cost
                                WZ      0.4426877E-02        0.000000
                                WT      0.2166008E-01        0.000000
                              WZAP      0.9090909E-01        0.000000

                               Row    Slack or Surplus      Dual Price
                                 1      0.3617391            1.000000
                            MEDVED       0.000000           0.1060870
                            GRANIT      0.8265613            0.000000
                            SLAPAR       0.000000           0.1217391
                            PLAMEN      0.6382609            0.000000
                 ZASEBNA_POPRAVILA      0.4426877E-02        0.000000
                   TOVAR_POPRAVILA      0.2166008E-01        0.000000
                         ZAPOSLENI      0.9090909E-01        0.000000
                  DODATNA_OMEJITEV       0.000000           0.3617391
```

**Vrednost namenske funkcije** je tokrat 0,3617. Numerična vrednost učinkovitosti podjetja Plamen je tako 36,17 %, kar smo prej pokazali tudi z grafično metodo.

V tabeli **Vrednost** je vrednost $W_Z$ enaka 0,00443, kar pomeni, da če bi podjetje izvedlo eno zasebno popravilo več ob istem številu zaposlenih in istem številu tovarniških popravil, bi s tem svojo učinkovitost dvignilo za 0,443 %, če pa bi opravilo eno tovarniško popravilo več, bi učinkovitost dvignilo za 9,09 %.

Podatek o tem, za koliko mora podjetje izboljšati svojo učinkovitost, da bo njegovo delovanje 100 % učinkovito lahko izračunamo tako, da vrednost namenske funkcije odštejemo od 1:

$$1 - 0{,}3617 = 0{,}6383$$

Podjetje Granit mora svojo učinkovitost povečati za 63,83 %, da bo 100 % učinkovito.

LINGO koda za izračun relativne učinkovitosti podjetja Slapar:

```
!Namenska funkcija;
max = 80*WZ + 55*WT;
!Omejitve;
[Medved] 125*WZ + 50*WT - 18*WZap <= 0;
[Granit] 44*WZ + 20*WT - 16*WZap <= 0;
[Slapar] 80*WZ + 55*WT - 17*WZap <= 0;
[Plamen] 23*WZ + 12*WT - 11*WZap <= 0;
[Zasebna_popravila] WZ >= 0;
[Tovar_popravila] WT >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 17*WZap = 1;
```

Poročilo o rešitvi za DEA analizo (Slapar):

```
Global optimal solution found.
Objective value:                              1.000000
Infeasibilities:                              0.000000
Total solver iterations:                             1
Elapsed runtime seconds:                          0.05

Model Class:                                        LP

                      Variable           Value        Reduced Cost
                            WZ        0.000000            0.000000
                            WT       0.1818182E-01        0.000000
                          WZAP       0.5882353E-01        0.000000

                           Row    Slack or Surplus      Dual Price
                             1        1.000000            1.000000
                        MEDVED       0.1497326            0.000000
                        GRANIT       0.5775401            0.000000
                        SLAPAR        0.000000            1.000000
                        PLAMEN       0.4288770            0.000000
             ZASEBNA_POPRAVILA        0.000000            0.000000
               TOVAR_POPRAVILA       0.1818182E-01        0.000000
                     ZAPOSLENI       0.5882353E-01        0.000000
            DODATNA_OMEJITEV          0.000000            1.000000
```

**Vrednost namenske funkcije** je tokrat 1. Numerična vrednost učinkovitosti podjetja Slapar je 100 %, kar smo prej pokazali tudi z grafično metodo.

Ker je podjetje na področju zasebnih popravil že 100% učinkovito, svoje učinkovitosti s povečanjem števila teh popravil ne more dvigniti.

V tabeli **Vrednost** je vrednost $W_T$ enaka 0,01818, kar pomeni, da če bi podjetje izvedlo eno zasebno popravilo več ob istem številu zaposlenih in istem številu tovarniških popravil, bi s tem svojo učinkovitost dvignilo za 1,818 %.

### 2.2.3 Reševanje analize ovojnice podatkov na primeru izhodno usmerjenega modela s programom Excel

V tem poglavju bomo spoznali še, kako rešimo primer DEA izhodno usmerjene analize v programu Excel. Za primer bomo vzeli primer Servisna podjetja II.

V programu Excel oblikujemo tabelo s podatki o številu popravil v posameznih podjetjih in številu zaposlenih v njih. Poleg tabele s podatki pripravimo še tabelo za uteži, vendar pustimo celice prazne.

V pripravljeno celico za namensko funkcijo vnesemo formulo za podjetje Granit, saj bomo najprej analizirali to podjetje. Spomnimo, da smo pri uvedbi linearnega programa zapisali, da je namenska funkcija utežena vsota izhodnih podatkov.

Slika 41: Priprava podatkov za DEA analizo v programu Excel

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih | | | |
| 3 | | Medved | 125 | 50 | 18 | | Uteži | |
| 4 | | Granit | 44 | 20 | 16 | | wz | |
| 5 | | Slapar | 80 | 55 | 17 | | wt | |
| 6 | | Plamen | 23 | 12 | 11 | | wzap | |
| 7 | | | | | | | | |
| 8 | | | Namenska funkcija | =C4*H4+D4*H5 | | | | |
| 9 | | | | | | | | |

Pripravimo celice za omejitve. Te bomo tokrat definirali nekoliko drugače kot v programu LINGO. Spomnimo, da smo omejitev podjetja Granit najprej razvili do takšne oblike:

$$0 \leq (125W_Z + 50W_T) \leq 18W_{Zap}$$

Omejitev lahko torej izrazimo tudi kot:

$$(125W_Z + 50W_T) \leq 18W_{Zap}$$

Pogoj, da mora biti izraz večji od 0 pa bo izpolnjen pod pogojem, da morajo biti vse vrednosti uteži večje od 0. Na levo stran neenačbe v Excelu za vsa podjetja vnesemo uteženo vsoto izhodnih podatkov. Na desno stran pa za vsa podjetja vnesemo uteženo vsoto vhodnih podatkov.

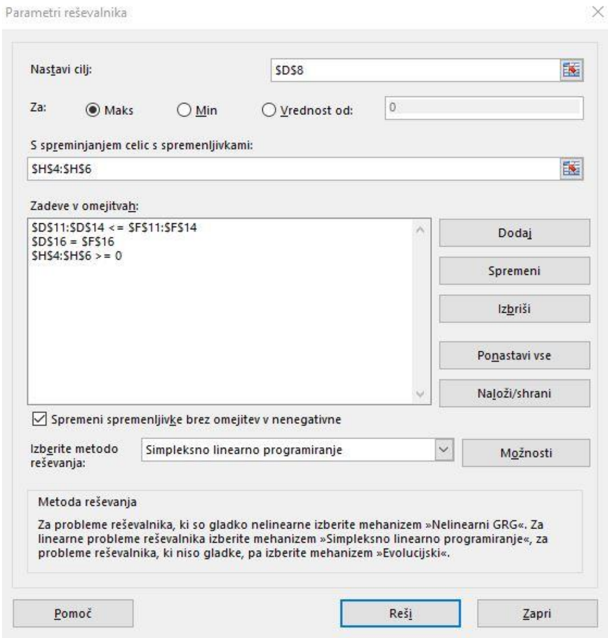
Potrebujemo le še dodatno omejitev. Utežena vsota vhodnih podatkov opazovanega podjetja mora biti enaka 1. Najprej bomo reševali primer za podjetje Granit.

Slika 42: Vnos omejitev za DEA analizo

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih | | | |
| 3 | | Medved | 125 | 50 | 18 | | Uteži | |
| 4 | | Granit | 44 | 20 | 16 | | wz | |
| 5 | | Slapar | 80 | 55 | 17 | | wt | |
| 6 | | Plamen | 23 | 12 | 11 | | wzap | |
| 7 | | | | | | | | |
| 8 | | | Namenska funkcija | =C4*H4+D4*H5 | | | | |
| 9 | | | | | | | | |
| 10 | | | Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov | | |
| 11 | | | Medved | =C3*$H$4+D3*$H$5 | <= | =E3*$H$6 | | |
| 12 | | | Granit | =C4*$H$4+D4*$H$5 | <= | =E4*$H$6 | | |
| 13 | | | Slapar | =C5*$H$4+D5*$H$5 | <= | =E5*$H$6 | | |
| 14 | | | Plamen | =C6*$H$4+D6*$H$5 | <= | =E6*$H$6 | | |
| 15 | | | | | | | | |
| 16 | | | Dodatna omejitev | =E4*H6 | = | 1 | | |

Vnesemo potrebne parametre v Reševalnik. Namensko funkcijo maksimiramo. V omejitve dodamo omejitev, da je utežena vsota izhodnih podatkov lahko manjša ali enaka uteženi vsoti vhodnih podatkov, dodatno omejitev, ki smo jo definirali in pogoj, da morajo biti odločitvene spremenljivke večje ali enake 0. Za metodo reševanja izberemo Simpleksno linearno programiranje.

Slika 43: Parametri Reševalnika za reševanje DEA




Slika 44: Rešitev za podjetje Granit

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih | | | |
| 3 | | Medved | 125 | 50 | 18 | | Uteži | |
| 4 | | Granit | 44 | 20 | 16 | | wz | 0,00304 |
| 5 | | Slapar | 80 | 55 | 17 | | wt | 0,01489 |
| 6 | | Plamen | 23 | 12 | 11 | | wzap | 0,0625 |
| 7 | | | | | | | | |
| 8 | | | Namenska funkcija | 0,43173913 | | | | |
| 9 | | | | | | | | |
| 10 | | | Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov | | |
| 11 | | | Medved | 1,125 | <= | 1,125 | | |
| 12 | | | Granit | 0,43173913 | <= | 1 | | |
| 13 | | | Slapar | 1,0625 | <= | 1,0625 | | |
| 14 | | | Plamen | 0,248695652 | <= | 0,6875 | | |
| 15 | | | | | | | | |
| 16 | | | Dodatna omejitev | 1 | = | 1 | | |
| 17 | | | | | | | | |

Program Excel je izpisal iste rezultate kot prej program LINGO. Zanima nas vrednost **Namenske funkcije**, ki nam v tem primeru pove, da podjetje Granit dosega 43,17 % relativno učinkovitost. V tabeli uteži pa lahko preberemo še, za koliko se bo spremenila učinkovitost podjetja, če se spremeni število izhodov za 1, ob predpostavki, da število zaposlenih ostane enako.

Če želimo sedaj preveriti učinkovitost podjetja Medved, moramo spremeniti namensko funkcijo in dodatno omejitev, vse ostalo ostane enako. Tudi v reševalniku ni potrebno ničesar spreminjati, le na novo ga zaženemo, ko spremenimo potrebno v delovnem listu. Enako naredimo tudi za preostali dve podjetji.

Slika 45: Sprememba podatkov za računanje učinkovitosti podjetja Medved

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | |
| wt | |
| wzap | |

| Namenska funkcija | =C3*H4+D3*H5 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | =C3*$H$4+D3*$H$5 | <= | =E3*$H$6 |
| Granit | =C4*$H$4+D4*$H$5 | <= | =E4*$H$6 |
| Slapar | =C5*$H$4+D5*$H$5 | <= | =E5*$H$6 |
| Plamen | =C6*$H$4+D6*$H$5 | <= | =E6*$H$6 |
| Dodatna omejitev | =E3*H6 | = | 1 |

Slika 46: Rešitev za podjetje Medved

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | 0,008 |
| wt | 0 |
| wzap | 0,055556 |

| Namenska funkcija | 1 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | 1 | <= | 1 |
| Granit | 0,352 | <= | 0,888888889 |
| Slapar | 0,64 | <= | 0,944444444 |
| Plamen | 0,184 | <= | 0,611111111 |
| Dodatna omejitev | 1 | = | 1 |

Zanima nas vrednost **Namenske funkcije**, ki je v tem primeru 1, torej podjetje Granit dosega 100 % učinkovitost. V tabeli uteži lahko preberemo še, za koliko se bo spremenila učinkovitost podjetja, če se spremeni število izhodov za 1, ob predpostavki, da število zaposlenih ostane enako.

Slika 47: Sprememba podatkov za računanje učinkovitosti podjetja Slapar

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | |
| wt | |
| wzap | |

| Namenska funkcija | =C5*H4+D5*H5 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | =C3*$H$4+D3*$H$5 | <= | =E3*$H$6 |
| Granit | =C4*$H$4+D4*$H$5 | <= | =E4*$H$6 |
| Slapar | =C5*$H$4+D5*$H$5 | <= | =E5*$H$6 |
| Plamen | =C6*$H$4+D6*$H$5 | <= | =E6*$H$6 |
| Dodatna omejitev | =E5*H6 | = | 1 |

Slika 48: Rešitev za podjetje Slapar

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | 0,00286 |
| wt | 0,01402 |
| wzap | 0,05882 |

| Namenska funkcija | 1 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | 1,058823529 | <= | 1,058823529 |
| Granit | 0,406342711 | <= | 0,941176471 |
| Slapar | 1 | <= | 1 |
| Plamen | 0,234066496 | <= | 0,647058824 |
| Dodatna omejitev | 1 | = | 1 |

Tudi podjetje Slapar dosega 100 % učinkovitost. Tabela uteži predstavlja vrednosti za katere se bo spremenila učinkovitost podjetja, če spremeni število izhodov za 1, ob predpostavki, da število zaposlenih ostane enako.

Slika 49: Sprememba podatkov za računanje učinkovitosti podjetja Plamen

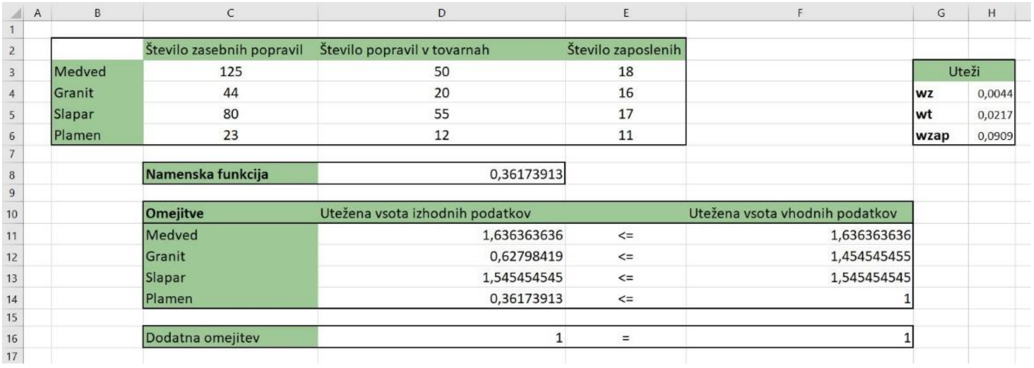
| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | |
| wt | |
| wzap | |

| Namenska funkcija | =C6*H4+D6*H5 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | =C3*$H$4+D3*$H$5 | <= | =E3*$H$6 |
| Granit | =C4*$H$4+D4*$H$5 | <= | =E4*$H$6 |
| Slapar | =C5*$H$4+D5*$H$5 | <= | =E5*$H$6 |
| Plamen | =C6*$H$4+D6*$H$5 | <= | =E6*$H$6 |
| Dodatna omejitev | =E6*H6 | = | 1 |

Slika 50: Rešitev za podjetje Plamen



| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

| Uteži | |
|---|---|
| wz | 0,0044 |
| wt | 0,0217 |
| wzap | 0,0909 |

| Namenska funkcija | 0,36173913 |
|---|---|

| Omejitve | Utežena vsota izhodnih podatkov | | Utežena vsota vhodnih podatkov |
|---|---|---|---|
| Medved | 1,636363636 | <= | 1,636363636 |
| Granit | 0,62798419 | <= | 1,454545455 |
| Slapar | 1,545454545 | <= | 1,545454545 |
| Plamen | 0,36173913 | <= | 1 |

| Dodatna omejitev | 1 | = | 1 |
|---|---|---|---|

Podjetje Plamen dosega 36,17 % relativno učinkovitost, kar lahko preberemo v tabeli Namenska funkcija. Rezultat je enak, kot smo ga dobili na grafični način in z reševanjem v LINGO. V tabeli uteži pa lahko preberemo še, za koliko se bo spremenila učinkovitost podjetja, če se spremeni število izhodov za 1, ob predpostavki, da število zaposlenih ostane enako.

### 2.2.4 Reševanje analize ovojnice podatkov s programom DEAP

V primeru, ko je za analizo ovojnice podatkov na voljo več vhodnih in izhodnih podatkov, lahko sestavljanje linearnega programa postane zakomplicirano in predstavljena programa LINGO in Excel nista več priročna. V tem poglavju zato prikazujemo uporabo programa DEAP, ki nekoliko poenostavi izvedbo analize ovojnice podatkov.

DEAP oziroma Data Envelopment Analysis (Computer) program je relativno enostavna aplikacija, namenjena analizam ovojnice podatkov.

Brez namestitve dodatka Win4DEAP lahko program uporabljamo brez uporabniškega vmesnika – le z ukaznim oknom. Postopek bo opisan v nadaljevanju.

Uporabimo primer, ki smo ga že rešili – problem servisnih podjetij II. Pripravili smo tabelo v programu Excel.

Slika 51: Priprava podatkov za reševanje v DEAP (1)

| | Število zasebnih popravil | Število popravil v tovarnah | Število zaposlenih |
|---|---|---|---|
| Medved | 125 | 50 | 18 |
| Granit | 44 | 20 | 16 |
| Slapar | 80 | 55 | 17 |
| Plamen | 23 | 12 | 11 |

Da bo DEAP lahko prebral podatke za analizo, je potrebno podatke v programu Excel preoblikovati. V tabeli ne sme biti besedila, le številski podatki. Podatki se morajo začeti v celici A1, pred prvim stolpcem podatkov ne sme biti praznih stolpcev, prav tako ne sme biti praznih vrstic.

V začetne stolpce moramo biti najprej vnesti izhodne podatke, šele ko izhodnih podatkov ni več, lahko vnesemo vhodne podatke.

V konkretnem primeru bodo v stolpcu A podatki o številu zasebnih popravil v podjetjih, v stolpcu B podatki o številu popravil v tovarnah, v stolpcu C pa podatki o zaposlenih.

Da bi program lahko prebral podatke, morajo biti shranjeni kot tekstovna datoteka – pri shranjevanju tabele v programu Excel izberemo tip datoteke Besedilo (MS-DOS) (*Text (MS-DOS)(*.txt)).*

Poudariti je treba, da vse tekstovne datoteke za izvedbo DEA analize shranjujemo v mapo DEAP, ki jo program tvori ob namestitvi. Pot do datoteke je  C:\DEAP.

Slika 52: Priprava podatkov za reševanje v DEAP (2)

**Podatki v programu Excel**

| | A | B | C |
|---|---|---|---|
| 1 | 125 | 50 | 18 |
| 2 | 44 | 20 | 16 |
| 3 | 80 | 55 | 17 |
| 4 | 23 | 12 | 11 |

**Podatki v tekstovni datoteki**

dea - Notepad

File Edit Format View Help

| | | |
|---|---|---|
| 125 | 50 | 18 |
| 44 | 20 | 16 |
| 80 | 55 | 17 |
| 23 | 12 | 11 |

V drugo tekstovno datoteko vnesemo ostale potrebne podatke o modelu, ki ga želimo analizirati.

Slika 53: Datoteka z navodili

```
dea-ins - Notepad
File  Edit  Format  View  Help
dea.txt             DATA FILE NAME
dea-out.txt             OUTPUT FILE NAME
4                 NUMBER OF FIRMS
1                NUMBER OF TIME PERIODS
2                 NUMBER OF OUTPUTS
1                 NUMBER OF INPUTS
0                 0=INPUT AND 1=OUTPUT ORIENTATED
0                0=CRS AND 1=VRS
0                0=DEA(MULTI-STAGE), 1=COST-DEA, 2=MALMQUIST-DEA, 3=DEA(1-STAGE), 4=DEA(2-STAGE)
```

V prvo vrstico vnesemo ime datoteke s podatki, ki smo jo ustvarili. V našem primeru smo jo poimenovali dea.txt.

V drugo vrstico vnesemo ime izhodne datoteke s podatki, ki jih bo program izračunal.

V tretji vrstici definiramo število podjetij oziroma DMU-jev, ki jih problem obravnava.

V četrti vrstici definiramo število izhodnih podatkov, ki so na voljo in v peti vrstici število vhodnih podatkov o proučevanih DMU-jih.

V šesti vrstici definiramo ali želimo model usmerjen na vhodne podatke (v tem primeru vpišemo 0) ali model usmerjen na izhodne podatke (v tem primeru vpišemo 1).

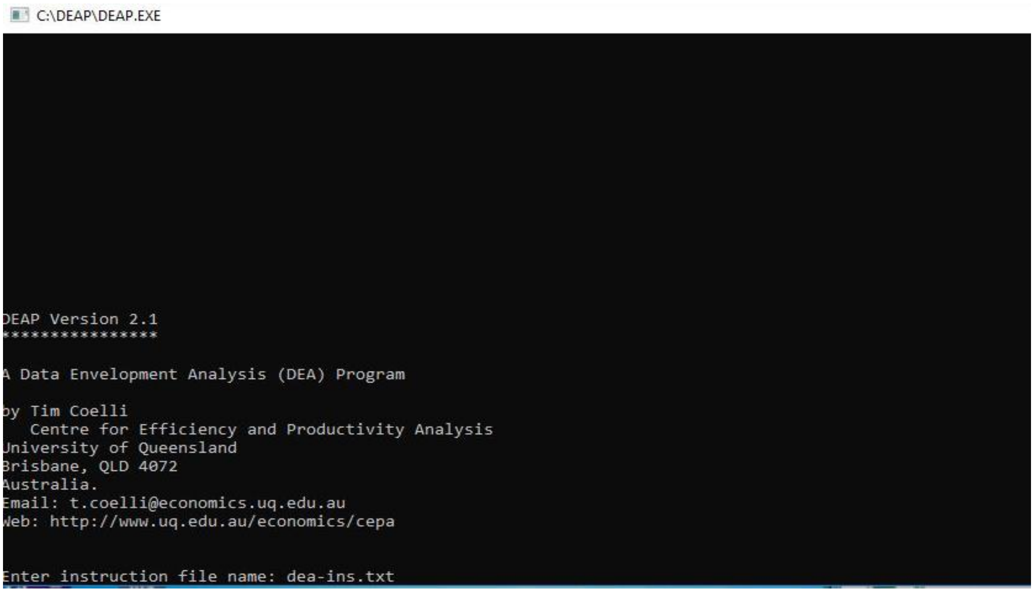
V sedmi vrstici definiramo ali želimo CRS[29] model (0) ali VRS[30] model (1).
V osmi vrstici definiramo model DEA analize. V vseh primerih, ki jih obravnava to gradivo izberemo 0 (večstopenjsko analizo).

[29] CRS model izračuna oceno učinkovitosti, ki jo imenujemo stalni donos tehnične učinkovitosti; primeren je kadar analizirana podjetja poslujejo v optimalnem obsegu
[30] VRS model izračuna oceno učinkovitosti, ki jo imenujemo spremenljivi donos tehnične učinkovitosti; primeren kadar analizirana podjetja ne poslujejo v optimalnem obsegu

Datoteko shranimo kot tekstovno datoteko, katere ime vsebuje –ins (npr. dea-ins.txt). Datoteka postane izvršna datoteka za DEA model.

V oknu programa DEAP v ukazno vrstico vnesemo ime izvršne datoteke, ki smo jo oblikovali.

Slika 54: Vnos ukazne vrstice v programu DEAP




Po kliku na tipko »enter« program zažene DEA analizo in v mapi oblikuje novo tekstovno datoteko z rešitvijo:

Results from DEAP Version 2.1

Instruction file = dea-ins.txt
Data file = dea.txt

Input orientated DEA

Scale assumption: CRS

Slacks calculated using multi-stage method

EFFICIENCY SUMMARY:

| firm | te |
|---|---|
| 1 | 1.000 |
| 2 | 0.432 |
| 3 | 1.000 |
| 4 | 0.362 |

Na začetku poročila se nahaja povzetek relativne učinkovitosti za vsa podjetja in aritmetična sredina, izračunana iz vseh posameznih. Za dani primer smo dobili iste rezultate kakor z ostalimi programi.

mean 0.698

FIRM BY FIRM RESULTS:

V nadaljevanju poročila najdemo podrobnosti za vsako DMU posebej.

Results for firm: 1
$\boxed{\text{Technical efficiency = 1.000}}$

Podatek o učinkovitosti DMU

PROJECTION SUMMARY:

| variable | original value | radial movement | slack movement | projected value |
|---|---|---|---|---|
| output 1 | 125.000 | 0.000 | 0.000 | 125.000 |
| output 2 | 50.000 | 0.000 | 0.000 | 50.000 |
| input 1 | 18.000 | 0.000 | 0.000 | 18.000 |

LISTING OF PEERS:
peer lambda weight
1 1.000

Ciljne vrednosti (Projected values) predstavljajo vrednosti vhodnih oz. izhodnih podatkov, ki bi jih podjetje moralo spremeniti, da bi postalo 100 % učinkovito.

Results for firm: 2
Technical efficiency = 0.432
PROJECTION SUMMARY:

| variable | original value | radial movement | slack movement | projected value |
|---|---|---|---|---|
| output 1 | 44.000 | 0.000 | 0.000 | 44.000 |
| output 2 | 20.000 | 0.000 | 0.000 | 20.000 |
| input 1 | 16.000 | -9.092 | 0.000 | $\boxed{6.908}$ |

Podjetje bi potrebovalo le 7 zaposlenih za takšno število popravil, da bi doseglo 100 % učinkovitost

LISTING OF PEERS:
peer lambda weight
1 0.285
3 0.104

Results for firm: 3
Technical efficiency = 1.000
PROJECTION SUMMARY:

| variable | original value | radial movement | slack movement | projected value |
|---|---|---|---|---|
| output 1 | 80.000 | 0.000 | 0.000 | 80.000 |
| output 2 | 55.000 | 0.000 | 0.000 | 55.000 |
| input 1 | 17.000 | 0.000 | 0.000 | 17.000 |

LISTING OF PEERS:
peer lambda weight
3 1.000

Results for firm: 4
Technical efficiency = 0.362
PROJECTION SUMMARY:

| variable | original value | radial movement | slack movement | projected value |
|---|---|---|---|---|
| output 1 | 23.000 | 0.000 | 0.000 | 23.000 |
| output 2 | 12.000 | 0.000 | 0.000 | 12.000 |
| input 1 | 11.000 | -7.021 | 0.000 | $\boxed{3.979}$ |

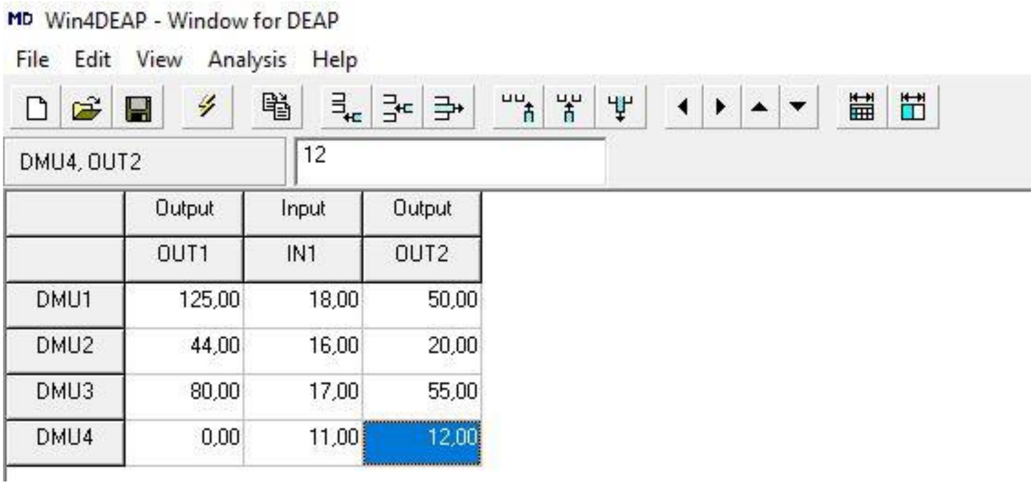
Podjetje bi potrebovalo le 4 zaposlene za takšno število popravil, da bi doseglo 100 % učinkovitost

LISTING OF PEERS:

peer lambda weight
1 0.106
3 0.122

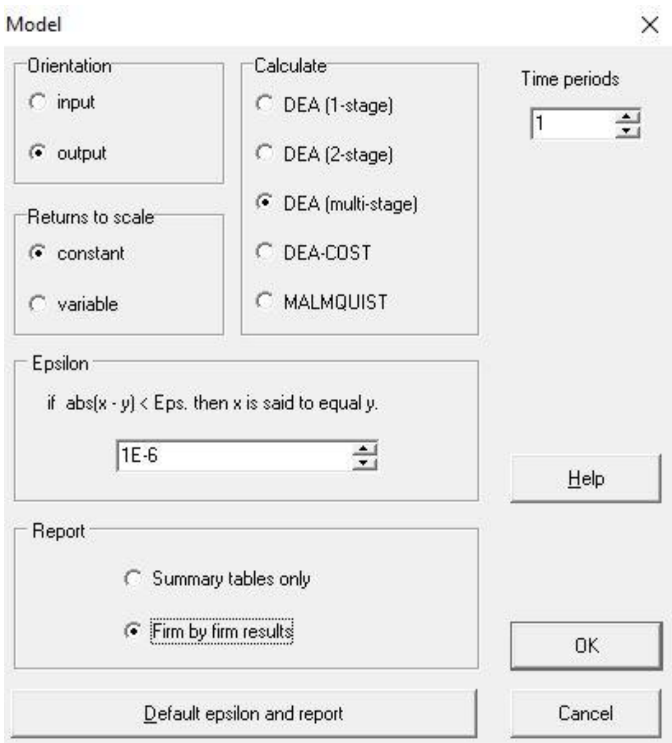
Uporabniški vmesnik Win4DEAP omogoča oblikovanje v uporabniškem vmesniku. Vnesemo število DMU-jev za analizo, dodamo stolpce za vhodne in izhodne podatke ter vtipkamo podatke.

Slika 55: Vnos podatkov v uporabniški vmesni Win4DEAP




Pod zavihkom Analiza (Analysis) kliknemo na Model in se odpre okno, kjer označimo parametre, ki smo jih prej vpisali v izvršno datoteko.

Slika 56: Vnos parametrov za DEA analizo v programu DEAP



Podatki, ki jih program izpiše so enaki, kot če uporabimo program brez uporabniškega vmesnika.

## 2.3 Dodatni primeri

**Primer 12:**

TRIJE KMETJE

Trije kmetje z določenim številom strojev (vhodni podatek) obdelujejo svoje površine in redijo govedo (izhodni podatek). Vrednosti vhodnih in izhodnih podatkov za vsakega kmeta so zapisane v spodnji tabeli. Za vsakega kmeta bomo poskušali ugotoviti, kako uspešno izkorišča svoje stroje.

| | Hektarji obdelovalne zemlje | Živina | Stroji |
|---|---|---|---|
| Govedič | 20 | 15 | 20 |
| Smrekar | 16 | 30 | 25 |
| Bučar | 8 | 10 | 12 |

**Rešitev:**

Najprej izračunamo količnike med vhodnim in izhodnima podatkoma.

| | $\frac{hektar}{stroj}$ | $\frac{živina}{stroj}$ |
|---|---|---|
| Govedič | 1 | 0,75 |
| Smrekar | 0,64 | 1,2 |
| Bučar | 0,67 | 0,83 |

Iz zgornjih vrednosti lahko razberemo, da stroje za obdelavo zemlje najboljše izkorišča kmet Govedič, saj količnik za izkoriščenost strojev pri obdelavi zemlje znaša 1. Kmet Govedič je na tem področju torej 100 % uspešen. Ostala dva kmeta imata na področju izkoriščanja strojev za obdelavo zemlje slabšo učinkovitost. Stroje pri reji živine najbolje izkorišča kmet Smrekar, saj je količnik za ta izhodni podatek pri njemu najvišji (1,2). Kmet Smrekar ima na področju izkoriščenosti strojev za rejo živali 100 % raven uspešnosti, kmeta Govedič in Bučar pa dosegata nižjo uspešnost na področju reje.

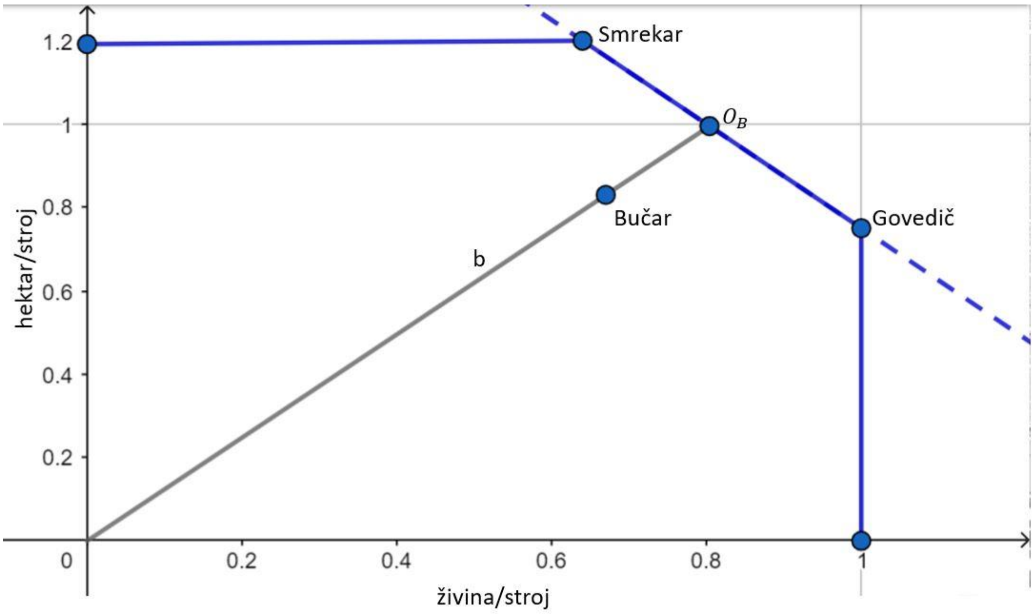
Koliko odstotno uspešnost izkoristka strojev dosegata Smrekar in Bučar pri obdelavi zemlje ter Govedič in Bučar pri reji živali, bomo določili na podlagi njihovih relativnih uspešnosti. Relativno uspešnost Smrekarja pri obdelavi zemlje izračunamo tako, da njegov količnik na tem področju (0,64) delimo s količnikom kmeta, ki dosega na tem področju 100 % uspešnost. To je Govedič, katerega količnik znaša 1.

Relativna uspešnost Smrekarja na področju obdelave zemlje znaša $\frac{0,64}{1} = 0,64$. Dobljeno vrednost pomnožimo s 100 in lahko zaključimo, da je relativna uspešnost kmeta Smrekarja 64 %. Ostale relativne vrednosti so prikazane v spodnji tabeli.

| | $\frac{hektar}{stroj}$ | $\frac{živina}{stroj}$ |
|---|---|---|
| Govedič | 100 % | 62,5 % |
| Smrekar | 64 % | 100 % |
| Bučar | 67 % | 69,17 % |

Za Govediča in Smrekarja torej lahko trdimo, da (prvi na področju obdelave zemlje, drugi na področju živinoreje) dosegata 100 % nivo učinkovitosti. Kolikšen nivo učinkovitosti pa v celoti dosega kmet Bučar, bomo ugotovili z grafično metodo. Na spodnjem grafu so prikazane lege posameznih kmetov. Pri tem smo na abscisno os nanizali količnik $\frac{hektar}{stroj}$, na ordinatno os pa količnik $\frac{živina}{stroj}$.

Slika 57: Graf uspešnosti kmetov




Raven učinkovitosti kmeta Bučarja bomo dobili tako, da bomo primerjali trenutno oddaljenost kmeta Bučarja od izhodišča koordinatnega sistema in oddaljenost optimalne lege kmeta Bučarja na grafu in koordinatnim izhodiščem. Optimalna lega, ki predstavlja optimalno uspešnost, ki jo kmet Bučar lahko doseže, je označena s točko $O_B$ na grafu.

Trenutno oddaljenost kmeta Bučarja od izhodišča izračunamo s formulo za oddaljenost med dvema točkama v koordinatnem sistemu:

$$d(A, B) = |AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

kjer so $x_1, x_2, y_1$in $y_2$ koordinate točk med katerima računamo razdaljo.

Prva točka, ki jo upoštevamo v izračunu je seveda koordinatno izhodišče s koordinatami $(0{,}0)$, druga točka pa trenutni položaj kmeta - točka s koordinatama $(0.67{,}0.83)$. Trenutna oddaljenost kmeta Bučarja od izhodišča koordinatnega izhodišča torej znaša:

$$d_B = \sqrt{(0{,}67 - 0)^2 + (0{,}83 - 0)^2} = \sqrt{0{,}4489 + 0{,}6889} = \sqrt{1{,}1378} = 1{,}07$$

Da bi lahko izračunali presečišče premic, moramo najprej poiskati enačbo premice $o$. Spomnimo, da je osnovna enačba linearne funkcije $y = kx + n$, kjer sta $k$ in $n$ poljubni realni števili.

Postopek iskanja enačbe premice, ki poteka skozi dve točki, za kateri poznamo koordinate, dobimo tako, da najprej izračunamo smerni koeficient premice $k$. Smerni koeficient izračunamo tako, da koordinati poznanih točk vstavimo v enačbo:

$$k = \frac{y_2 - y_1}{x_2 - x_1}$$

Iščemo enačbo premice $o$, ki poteka skozi legi kmetov Smrekarja in Govedarja, zato bomo upoštevali koordinate za kmeta Govedarja $(1, \frac{3}{4})$ in za kmeta Smrekarja $(\frac{16}{25}, \frac{6}{5})$:

$$k = \frac{\frac{6}{5} - \frac{3}{4}}{\frac{16}{25} - 1} = \frac{0{,}45}{-0{,}36} = -1{,}25$$

Število $n$ iz prej zapisane enačbe pa določimo s pomočjo enačbe $y = k(x - x_1) + y_1$, kjer za $k$ uporabimo predhodno izračunano vrednost smernega koeficienta ter koordinati ene izmed točk:

$$y = -1{,}25(x + 1) + 0{,}75$$

$$y = -1{,}25x + 2$$

Enačba premice, ki predstavlja ovoj podatkov je torej $y = -1{,}25x + 2$. Na isti način moramo sedaj določiti še enačbo premice $b$, ki poteka skozi koordinatno izhodišče $(0{,}0)$ in trenutno lego kmeta Bučarja $(\frac{67}{100}, \frac{83}{100})$:

$$k = \frac{\frac{83}{100} - 0}{\frac{67}{100} - 0} = \frac{0{,}83}{0{,}67} = 1{,}24$$

Premica $b$ poteka skozi koordinatno izhodišče, torej je njena začetna vrednost enaka 0. Njen funkcijski predpis je tako:

$$y = 1{,}24x$$

Točka $OB$, ki predstavlja optimalno lego kmeta Bučarja, leži na presečišču premic $o$ in $b$, katerih enačbe smo pravkar določili. Koordinati presečišča izračunamo tako, da enačimo funkcijska predpisa teh premic. Dobimo koordinato $x$:

$$1{,}24x = -1{,}25x + 2$$
$$2{,}49x = 2$$
$$x = 0{,}81$$

Koordinato $y$ pa izračunamo tako, da izračunano koordinato $x = 0{,}81$ vstavimo v eno izmed enačb premic:

$$y = 1{,}24x$$
$$y = 1{,}24 \cdot 0{,}81$$
$$y = 1{,}004$$

Optimalna lega kmeta Bučarja je torej točka $OB\,(0.81, 1.004)$. Sedaj lahko izračunamo še njeno oddaljenost od koordinatnega izhodišča, na enak način kot smo prej izračunali oddaljenost trenutne lege do izhodišča:

$$d_{OB} = \sqrt{(0{,}81 - 0)^2 + (1{,}004 - 0)^2} = \sqrt{0{,}6561 + 1{,}028} = \sqrt{1{,}6841} = 1{,}29$$

Sedaj, ko imamo izračunani razdalji obeh točk s koordinatnim izhodiščem, lahko izračunamo njun količnik. Trenutna lega kmeta Bučarja je od koordinatnega izhodišča oddaljena 1,07, optimalna lega pa 1,29.

$$\frac{d_B}{d_{OB}} = \frac{1{,}07}{1{,}29} = 0{,}829$$
$$0{,}829 \cdot 100 = 82{,}9\ \%$$

Trenutna učinkovitost kmeta Bučarja je 82,9 % napram kmetoma Govedič in Smrekar, ki sta 100 % učinkovita.

**<u>Primer 13:</u>**

TRI OPTIKE

Tri trgovine O1, O2 in O3, ki se ukvarjajo z okulističnimi pregledi in prodajo očal, zaposlujejo različno število ljudi. V spodnji tabeli je podano število zaposlenih ter število pregledov in prodanih očal pri vsakem optiku na mesec. Zanima nas relativna učinkovitost trgovin. Problem rešite s pomočjo programa LINGO.

| | Pregledi | Prodana očala | Zaposleni |
|---|---|---|---|
| O1 | 6 | 24 | 3 |
| O2 | 12 | 60 | 4 |
| O3 | 28 | 35 | 7 |

V primeru optik sta na voljo dva izhodna podatka (število pregledov in število prodanih očal). Označimo uteži za ta dva izhodna podatka kot $W_P$ in $W_O$. Vhodni podatek je v tem primeru samo en – število zaposlenih v optiki, katerega utež označimo z $W_{ZAP}$.

Osredotočimo se zopet na izhodno usmerjeni model DEA analize. Želimo doseči maksimalno uteženo vsoto izhodov v primeru optike O1:

$$\max U_1$$

Poglejmo na primeru optike O1, kako zapisati celotno namensko funkcijo. Optika O1 opravi 6 pregledov in proda 24 očal, uteženo vsoto izhodnih podatkov zato zapišemo kot $6W_P + 24W_O$:

$$\max U_1 = 6W_P + 24W_O$$

Po preureditvi vseh omejitev, kar je natančno opisano pri zgledu o servisnih podjetjih II., smo oblikovali omejitve, kjer mora biti od vsote uteženih izhodnih podatkov odšteta vsota vhodnih podatkov, rezultat pa ne sme biti večji kot 1. Vsoto uteženih izhodnih podatkov za optiko O1 smo že določili ($6W_P + 24W_O$), utežena vsota vhodnih podatkov pa je na število zaposlenih v optiki (3) enaka $3W_{ZAP}$. Omejitev za optiko O1 je tako:

$$(6W_P + 24W_O - 3W_{ZAP}) \leq 0$$

Za ostali dve optiki napišemo omejitev na enak način:

$$(12W_P + 60W_O - 4W_{ZAP}) \leq 0$$

$$(28W_P + 35W_O - 7W_{ZAP}) \leq 0$$

Zapisati moramo še vse dodatne pogoje. Spomnimo se, da uteži ne morejo biti negativno število in da moramo fiksirati uteženo vsoto vhodnih podatkov za optiko O1.

$$W_P \geq 0$$
$$W_O \geq 0$$
$$W_{ZAP} \geq 0$$
$$3W_{ZAP} = 1$$

Linearni program vpišemo v program LINGO:

```
!Namenska funkcija;
max = 6*WP + 24*WO;
!Omejitve;
[O1] 6*WP + 24*WO - 3*WZap <= 0;
[O2] 12*WP + 60*WO - 4*WZap <= 0;
[O3] 28*WP + 35*WO - 7*WZap <= 0;
[Pregledi] WZ >= 0;
[Tovar_popravila]WP >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 3*WZap = 1;
```

Poročilo o rešitvi za optiko O1:

```
 Global optimal solution found.
   Objective value:                              0.6222222
   Infeasibilities:                               0.000000
   Total solver iterations:                              2
   Elapsed runtime seconds:                           0.05

   Model Class:                                         LP


                                Variable           Value        Reduced Cost
                                      WP       0.7407407E-01        0.000000
                                      WO       0.7407407E-02        0.000000
                                    WZAP       0.3333333            0.000000
                                      WZ        0.000000            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1       0.6222222            1.000000
                                      O1       0.3777778            0.000000
                                      O2        0.000000           0.3666667
                                      O3        0.000000         0.5714286E-01
                                PREGLEDI        0.000000            0.000000
                        TOVAR_POPRAVILA        0.7407407E-01        0.000000
                               ZAPOSLENI       0.3333333            0.000000
                       DODATNA_OMEJITEV         0.000000           0.6222222
```

**Vrednost namenske funkcije** je 0,622. Numerična vrednost učinkovitosti optike O1 je tako 62,22 %.

V tabeli ***Vrednost*** vrednost $W_P$ enaka 0,074, kar pomeni, da če bi optika izvedla en pregled več ob istem številu zaposlenih in istem številu prodanih očal, bi s tem svojo učinkovitost dvignila za 7,4 %, če pa bi prodala ena očala več pa bi učinkovitost dvignila za 7,4 %.

Podatek o tem, za koliko mora izboljšati svojo učinkovitost, da bo njegovo delovanje 100% učinkovito izračunamo tako, da vrednost namenske funkcije odštejemo od 1:

$$1 - 0{,}622 = 0{,}378$$

Optika O1 mora svojo učinkovitost povečati za 37,8 %, da bo 100 % učinkovita.

Ostali dve optiki lahko analiziramo tako, da v zgornjem linearnem programu smiselno popravimo namensko funkcijo in dodatno omejitev. Vse ostalo v programu pustimo isto.

Lingo sintaksa za izračun relativne učinkovitosti optike O2:

```
!Namenska funkcija;
max = 12*WP + 60*WO;
!Omejitve;
[O1] 6*WP + 24*WO - 3*WZap <= 0;
[O2] 12*WP + 60*WO - 4*WZap <= 0;
[O3] 28*WP + 35*WO - 7*WZap <= 0;
[Pregledi] WZ >= 0;
[Tovar_popravila]WP >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 4*WZap = 1;
```

Poročilo o rešitvi za optiko O2:

```
 Global optimal solution found.
 Objective value:                              1.000000
 Infeasibilities:                              0.000000
 Total solver iterations:                             1
 Elapsed runtime seconds:                          0.05

 Model Class:                                        LP


                              Variable           Value        Reduced Cost
                                    WP        0.000000            0.000000
                                    WO       0.1666667E-01        0.000000
                                  WZAP       0.2500000            0.000000
                                    WZ        0.000000            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        1.000000            1.000000
                                    O1       0.3500000            0.000000
                                    O2        0.000000            1.000000
                                    O3        1.166667            0.000000
                              PREGLEDI        0.000000            0.000000
                       TOVAR_POPRAVILA        0.000000            0.000000
                             ZAPOSLENI       0.2500000            0.000000
                      DODATNA_OMEJITEV        0.000000            1.000000
```

Optika O2 je 100 % učinkovita. ***Vrednost namenske funkcije*** je namreč enaka 1.

Svoje učinkovitosti glede opravljanja pregledov ne more izboljšati**,** lahko pa izboljša svojo učinkovitost pri prodaji očal. Če bi povečali prodajo očal za 1, ob predpostavki, da ne spremenijo števila pregledov in število zaposlenih ostane enako, bi se učinkovitost povečala za 1,67 %.

Lingo sintaksa za izračun relativne učinkovitosti optike O3:

```
!Namenska funkcija;
max = 28*WP + 35*WO;
!Omejitve;
[O1] 6*WP + 24*WO - 3*WZap <= 0;
[O2] 12*WP + 60*WO - 4*WZap <= 0;
[O3] 28*WP + 35*WO - 7*WZap <= 0;
[Pregledi] WZ >= 0;
[Tovar_popravila]WP >= 0;
[Zaposleni] WZap >= 0;
[Dodatna_omejitev] 7*WZap = 1;
```

Poročilo o rešitvi za optiko O3:

```
Global optimal solution found.
  Objective value:                              1.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             1
  Elapsed runtime seconds:                          0.04

  Model Class:                                        LP

                     Variable           Value        Reduced Cost
                           WP       0.3571429E-01        0.000000
                           WO        0.000000            0.000000
                         WZAP       0.1428571            0.000000
                           WZ        0.000000            0.000000

                          Row    Slack or Surplus      Dual Price
                            1        1.000000            1.000000
                           O1       0.2142857            0.000000
                           O2       0.1428571            0.000000
                           O3        0.000000            1.000000
                     PREGLEDI        0.000000            0.000000
              TOVAR_POPRAVILA       0.3571429E-01        0.000000
                    ZAPOSLENI       0.1428571            0.000000
             DODATNA_OMEJITEV        0.000000            1.000000
```

Tudi optika O3 je 100 % učinkovita. **Vrednost namenske funkcije** je namreč tudi tukaj 1. Če pogledamo v tabelo **Vrednost,** vidimo da 100 % učinkovitost dosega na področju prodaje očal. Vrednost $W_O$ je namreč 0. Lahko pa svojo učinkovitost izboljša na področju opravljanja pregledov. Če bi povečala število pregledov za 1, število prodanih očal in število zaposlenih pa bi ostala nespremenjena, bi se relativna učinkovitost dvignila za 0,0357 oziroma 3,6 %.

**Primer 14:**

RAZVRŠČANJE AVTOMOBILOV

Neko prevozniško podjetje bo najelo določeno število vozil tipa A. Na trgu se je pojavilo podjetje, ki vozila tipa A daje v najem. Na voljo imajo 5 vozil. Za lažjo odločitev najemnika, katera vozila naj najame ponujajo naslednje podatke:

| Vozilo | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Število ur v uporabi za lansko leto | 2100 | 1800 | 2300 | 1900 | 2500 |
| Število okvar | 3 | 4 | 3 | 2 | 4 |
| Število izgubljenih dni zaradi popravil | 5 | 12 | 7 | 6 | 4 |
| Čas porabljen za pripravo na vožnjo v minutah | 23 | 13 | 26 | 31 | 21 |
| Poraba goriva v $L/100$ km | 10 | 11 | 9 | 13 | 14 |

Ker bomo nalogo rešili s pomočjo programa Excel, tabelo prenesemo tja. Vhodni podatek za vsa vozila predstavlja število ur v uporabi za lansko leto, vsi ostali podatki so izhodni podatki.

Uteži za posamezne podatke označimo kot:

$$W_{UR} \dots število\ ur\ v\ uporabi\ za\ lansko\ leto$$

$$W_{OKVAR} \dots število\ okvar$$

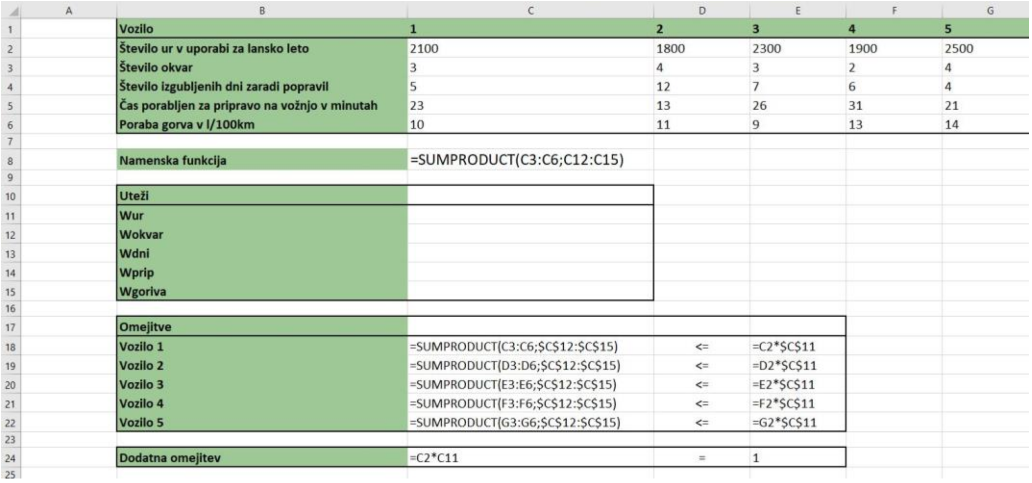
$$W_{DNI} \dots število\ izgubljenih\ dni\ zaradi\ popravil$$

$$W_{PRIP} \dots čas\ porabljen\ za\ pripravo\ na\ vožnjo\ v\ minutah$$

$$W_{GORIVA} \dots poraba\ goriva$$

Namenska funkcija predstavlja uteženo vsoto izhodnih podatkov. Vpišemo še vse omejitve. Spomnimo, da na levo stran neenačb omejitev za posamezna vozila vpišemo uteženo vsoto izhodov, na desno stan neenačb pa uteženo vsoto vhodov. Dodamo še dodatno omejitev, ki določa, da mora biti utežena vsota vhodov opazovanega vozila enaka 1.

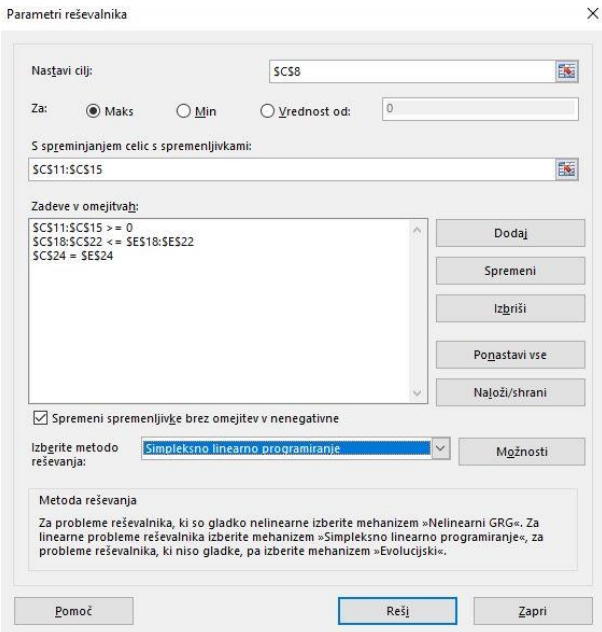
Slika 58: Priprava podatkov v programu Excel za vozilo 1



| | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 1 | Vozilo | 1 | 2 | 3 | 4 | 5 |
| 2 | Število ur v uporabi za lansko leto | 2100 | 1800 | 2300 | 1900 | 2500 |
| 3 | Število okvar | 3 | 4 | 3 | 2 | 4 |
| 4 | Število izgubljenih dni zaradi popravil | 5 | 12 | 7 | 6 | 4 |
| 5 | Čas porabljen za pripravo na vožnjo v minutah | 23 | 13 | 26 | 31 | 21 |
| 6 | Poraba gorva v l/100km | 10 | 11 | 9 | 13 | 14 |
| 7 | | | | | | |
| 8 | Namenska funkcija | =SUMPRODUCT(C3:C6;C12:C15) | | | | |
| 9 | | | | | | |
| 10 | Uteži | | | | | |
| 11 | Wur | | | | | |
| 12 | Wokvar | | | | | |
| 13 | Wdni | | | | | |
| 14 | Wprip | | | | | |
| 15 | Wgoriva | | | | | |
| 16 | | | | | | |
| 17 | Omejitve | | | | | |
| 18 | Vozilo 1 | =SUMPRODUCT(C3:C6;$C$12:$C$15) | <= | =C2*$C$11 | | |
| 19 | Vozilo 2 | =SUMPRODUCT(D3:D6;$C$12:$C$15) | <= | =D2*$C$11 | | |
| 20 | Vozilo 3 | =SUMPRODUCT(E3:E6;$C$12:$C$15) | <= | =E2*$C$11 | | |
| 21 | Vozilo 4 | =SUMPRODUCT(F3:F6;$C$12:$C$15) | <= | =F2*$C$11 | | |
| 22 | Vozilo 5 | =SUMPRODUCT(G3:G6;$C$12:$C$15) | <= | =G2*$C$11 | | |
| 23 | | | | | | |
| 24 | Dodatna omejitev | =C2*C11 | = | 1 | | |
| 25 | | | | | | |

Odpremo orodje Reševalnik in vnesemo potrebne parametre.

Slika 59: Vnos parametrov v Reševalnik




Podrobnejši opis vnosov parametrov v reševalnik je opisan pri primeru Storitvena podjetja II.

Slika 60: Rešitev za vozilo 1

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 7 | | | | | |
| 8 | | **Namenska funkcija** | 0,900389 | | |
| 9 | | | | | |
| 10 | | **Uteži** | | | |
| 11 | | **Wur** | 0,00047619 | | |
| 12 | | **Wokvar** | 0,15111759 | | |
| 13 | | **Wdni** | 0 | | |
| 14 | | **Wprip** | 0,019436346 | | |
| 15 | | **Wgoriva** | 0 | | |
| 16 | | | | | |
| 17 | | **Omejitve** | | | |
| 18 | | **Vozilo 1** | 0,9003887 | <= | 1 |
| 19 | | **Vozilo 2** | 0,8571429 | <= | 0,8571 |
| 20 | | **Vozilo 3** | 0,9586978 | <= | 1,0952 |
| 21 | | **Vozilo 4** | 0,9047619 | <= | 0,9048 |
| 22 | | **Vozilo 5** | 1,0126336 | <= | 1,1905 |
| 23 | | | | | |
| 24 | | **Dodatna omejitev** | 1 | = | 1 |
| 25 | | | | | |

Vozilo 1 dosega 90,03 % učinkovitost. Njegova relativna učinkovitost bi se izboljšala, če bi se zmanjšalo število okvar (ena okvara manj bi pomenila 5,11 % izboljšanje) in čas priprave vozila na vožnjo (ena minuta manj bi pomenila 1,9 %) izboljšanja.

Da bi pridobili relativno učinkovitost za ostala štiri vozila, spremenimo dodatno omejitev in namensko funkcijo.

Slika 61: Rešitev za vozilo 2

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 7 | | | | | |
| 8 | | **Namenska funkcija** | 1 | | |
| 9 | | | | | |
| 10 | | **Uteži** | | | |
| 11 | | **Wur** | 0,000555556 | | |
| 12 | | **Wokvar** | 0 | | |
| 13 | | **Wdni** | 0,058767952 | | |
| 14 | | **Wprip** | 0,022675737 | | |
| 15 | | **Wgoriva** | 0 | | |
| 16 | | | | | |
| 17 | | **Omejitve** | | | |
| 18 | | **Vozilo 1** | 0,8153817 | <= | 1,1667 |
| 19 | | **Vozilo 2** | 1 | <= | 1 |
| 20 | | **Vozilo 3** | 1,0009448 | <= | 1,2778 |
| 21 | | **Vozilo 4** | 1,0555556 | <= | 1,0556 |
| 22 | | **Vozilo 5** | 0,7112623 | <= | 1,3889 |
| 23 | | | | | |
| 24 | | **Dodatna omejitev** | 1 | = | 1 |

Vozilo 2 dosega 100 % učinkovitost na področju števila okvar in porabe goriva, kjer izboljšava ni mogoča. Učinkovitost pa bi lahko povečali z zmanjšanjem števila dni, ki so izgubljeni zaradi popravil (en dan manj bi povečal učinkovitost za 5,9 %) in z zmanjšanim časom priprave (ena minuta manj bi zvišala relativno učinkovitost za 2,2 %).

Slika 62: Rešitev za vozilo 3

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 7 | | | | | |
| 8 | | Namenska funkcija | 0,875333 | | |
| 9 | | | | | |
| 10 | | Uteži | | | |
| 11 | | Wur | 0,000434783 | | |
| 12 | | Wokvar | 0,13797693 | | |
| 13 | | Wdni | 0 | | |
| 14 | | Wprip | 0,017746229 | | |
| 15 | | Wgoriva | 0 | | |
| 16 | | | | | |
| 17 | | Omejitve | | | |
| 18 | | Vozilo 1 | 0,8220941 | <= | 0,913 |
| 19 | | Vozilo 2 | 0,7826087 | <= | 0,7826 |
| 20 | | Vozilo 3 | 0,8753327 | <= | 1 |
| 21 | | Vozilo 4 | 0,826087 | <= | 0,8261 |
| 22 | | Vozilo 5 | 0,9245785 | <= | 1,087 |
| 23 | | | | | |
| 24 | | Dodatna omejitev | 1 | = | 1 |

Vozilo 3 dosega 87,53 % učinkovitost. Svojo učinkovitost bi lahko izboljšalo pri vseh izhodnih parametrih. Kolikšna bi bila sprememba v relativni učinkovitosti ob spremembi posameznega izhodnega parametra, ob predpostavki, da vsi ostali ostanejo enaki, lahko razberemo v tabeli omejitve.

Slika 63: Rešitev za vozilo 4

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 7 | | | | | |
| 8 | | Namenska funkcija | 1 | | |
| 9 | | | | | |
| 10 | | Uteži | | | |
| 11 | | Wur | 0,000526316 | | |
| 12 | | Wokvar | 0 | | |
| 13 | | Wdni | 0 | | |
| 14 | | Wprip | 0,032258065 | | |
| 15 | | Wgoriva | 0 | | |
| 16 | | | | | |
| 17 | | Omejitve | | | |
| 18 | | Vozilo 1 | 0,7419355 | <= | 1,1053 |
| 19 | | Vozilo 2 | 0,4193548 | <= | 0,9474 |
| 20 | | Vozilo 3 | 0,8387097 | <= | 1,2105 |
| 21 | | Vozilo 4 | 1 | <= | 1 |
| 22 | | Vozilo 5 | 0,6774194 | <= | 1,3158 |
| 23 | | | | | |
| 24 | | Dodatna omejitev | 1 | = | 1 |
| 25 | | | | | |

Vozilo 4 dosega 100 % učinkovitost na treh izmed štirih izhodnih parametrih. Svojo učinkovitost bi lahko izboljšalo le, če bi zmanjšalo čas za pripravo vožnje (ena minuta manj bi pomenila 3,2 % izboljšanje relativne učinkovitosti).

Slika 64: Rešitev za vozilo 5

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 7 | | | | | |
| 8 | | **Namenska funkcija** | 0,88 | | |
| 9 | | | | | |
| 10 | | **Uteži** | | | |
| 11 | | **Wur** | 0,0004 | | |
| 12 | | **Wokvar** | 0,033333333 | | |
| 13 | | **Wdni** | 0 | | |
| 14 | | **Wprip** | 0 | | |
| 15 | | **Wgoriva** | 0,053333333 | | |
| 16 | | | | | |
| 17 | | **Omejitve** | | | |
| 18 | | **Vozilo 1** | 0,6333333 | <= | 0,84 |
| 19 | | **Vozilo 2** | 0,72 | <= | 0,72 |
| 20 | | **Vozilo 3** | 0,58 | <= | 0,92 |
| 21 | | **Vozilo 4** | 0,76 | <= | 0,76 |
| 22 | | **Vozilo 5** | 0,88 | <= | 1 |
| 23 | | | | | |
| 24 | | **Dodatna omejitev** | 1 | = | 1 |
| 25 | | | | | |

Vozilo 5 dosega 88 % učinkovitost. Kje bi se lahko izboljšalo in za koliko, ob spremembi izhodnih parametrov, lahko razberemo iz tabele omejitve.

# 3 ANALITIČNI-HIERARHIČNI PROCES (AHP)

Analitični-hierarhični proces[31] (AHP) je metoda, ki nam je lahko v veliko pomoč pri obsežnih in pomembnih odločitvah. Razvil jo je Thomas Saaty leta 1980, od takrat dalje pa je široko uporabljena na različnih področjih raziskovanja. Uporabimo jo lahko pri vseh vrstah večkriterijskih odločitev. Primer njene uporabe bomo predstavili na primeru izbire prevoznega sredstva do letališča.

Najprej je potrebno definirati cilj problema. Popotnik se odloča za izbiro prevoznega sredstva do letališča. Na voljo ima tri prevozna sredstva, med katerimi se mora odločiti.

Kot cilj problema lahko torej zapišemo: izbrati primerno prevozno sredstvo za pot do letališča.

Z osebnim avtomobilom se lahko pripelje do parkirišča letališča, pri prevozu nima težav s prtljago, saj ima na voljo veliko odlagalnega prostora, mora pa upoštevati stroške goriva in stroške parkirnega prostora za dva tedna, kolikor bo odsoten. Čas, ki ga porabi za pot do letališča je z osebnim avtomobilom najkrajša. Ni vezan na čas odhoda.

Z vlakom bodo njegovi stroški najnižji, med potjo pa mora enkrat prestopiti, saj iz njegovega domačega kraja do letališča ni direktne povezave. Pri tem mora vso svojo prtljago prelagati. Od železniške postaje do letališča je organiziran prevoz, vendar to zopet pomeni prestavljanje prtljage, za katero ima zelo malo prostora.

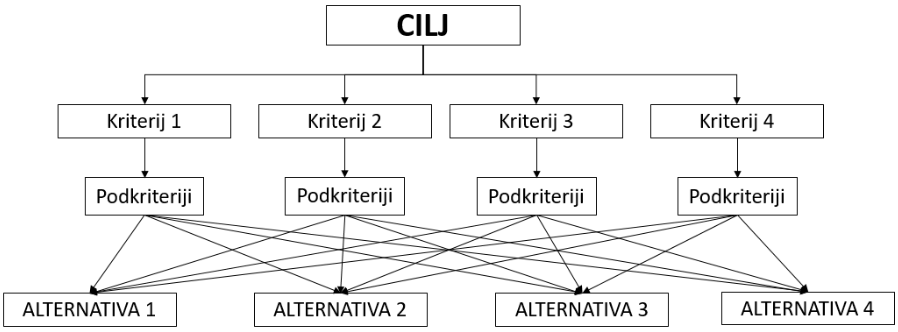
Lahko pa se odloči tudi za taksi, ki predstavlja najdražjo alternativo, vzame pa mu ravno toliko časa, kot če bi se peljal z osebnim avtomobilom. Taksist ga pride iskat pred vrata njegovega doma in ga odloži pred vhodom na letališče. Ker potuje sam, ima veliko prostora za svojo prtljago, s katero med potjo nima veliko dela, za prevoz prtljage mora sicer doplačati.

[31] Analytic hierarchy process

Vse tri opisane alternative bodo izpolnile cilj, ki si ga je popotnik zastavil – prispeti na letališče. Toda katera izmed njih bo zanj najugodnejša?

Definirali smo cilj problema, definirali smo alternative problema (torej sredstva, s katerimi potnik lahko doseže zastavljeni cilj), moramo pa postaviti še kriterije, s katerimi odločevalec alternative oceni (slika 65).

Slika 65: AHP struktura




Pri izbiri prevoza upoštevamo naslednje kriterije:

- kako udobno je ravnanje s prtljago glede na izbrano alternativo [UDOBJE];
- čas potovanja [ČAS];
- prestopanja z enega prevoznega sredstva na drugega [PRESTOPI];
- cena prevoza [CENA].

Poleg glavnih kriterijev za ocenjevanje, si lahko postavimo tudi podkriterije, s katerimi bolj podrobno definiramo kriterij. V našem primeru smo na podkriterije razdelili en kriterij:

- prestopanja z enega prevoznega sredstva na drugega [PRESTOPI];
  - število prestopanj [PRESTOPI 1];
  - čas čakanja na naslednje prevozno sredstvo [PRESTOPI 2]
  - razdalja med postajami prevoznega sredstva [PRESTOPI 3].

## 3.1 Osnovni principi AHP metodologije

Sestavimo matriko, ki izraža relativne vrednosti izbranih kriterijev. Pri danem primeru se bomo spraševali, kakšna je relativna pomembnost cene prevoza v primerjavi z ostalimi tremi kriteriji. Od odločevalca se zahteva, da izbere, ali je cena prevoza pomembnejša ali manj pomembna glede na ostale kriterije. Vsaka izmed sodb se nahaja na številski lestvici. Ena izmed najpogosteje uporabljenih številskih lestvic za odločanje po AHP metodologiji je prilagojena Saatyeva lestvica:

| Moč pomembnosti | Definicija | Opis |
|---|---|---|
| 1 | Enako pomembno | Kriterija, ki ju primerjamo med seboj sta enako pomembna |
| 3 | Zmerno pomembnejše | Kriterij A je zmerno pomembnejši kot kriterij B |
| 5 | Močno pomembnejše | Kriterij A je močno pomembnejši kot kriterij B |
| 7 | Zelo močno pomembnejše | Kriterij A je zelo močno pomembnejši kot kriterij B |
| 9 | Absolutno pomembnejši | Kriterij A je absolutno pomembnejši kot kriterij B |
| 2,4,6,8 | Vmesne vrednosti | Vmesne vrednosti |

Vir: Saaty, 1980

Osnova predpostavka je, če je kriterij A absolutno pomembnejši kot kriterij B in je ocenjen z oceno 9, je kriterij B absolutno manj pomemben kot A in je zato ocenjen z obratno vrednostjo ocene A, torej z $\frac{1}{9}$.

Parne primerjave izvedemo za vse izbrane kriterije v problemu. V primeru s štirimi različnimi kriteriji to pomeni 6 primerjav v prvem koraku. Če z $n$ označimo število izbranih kriterijev, lahko število primerjav enega kriterija izračunamo po formuli: $\frac{n-1}{2}$. Drugi izbrani kriterij primerjamo s preostalima dvema. Primerjave izvajamo toliko časa, dokler niso med seboj primerjani vsi faktorji. Ko imamo ocenjene vse primerjave, lahko sestavimo matriko relativnih vrednosti.

Popotnik ima v mislih štiri dejavnike, s pomočjo katerih se bo odločil za prevozno sredstvo do letališča. Kriterij 1 predstavlja udobje glede prtljage, kriterij 2 čas potovanja, kriterij 3 število prestopov in kriterij 4 cena uporabe prevoznega sredstva.

Pripravimo začetno matriko za primerjavo parov, v kateri glavna diagonala vsebuje vrednosti 1. Kadar primerjamo kriterije same s seboj, so ti vedno enako pomembni in po že predstavljeni Saatyevi lestvici imajo oceno 1.

| | Udobje | Čas | Prestopi | Cena |
|---|---|---|---|---|
| Udobje | 1 | | | |
| Čas | | 1 | | |
| Prestopi | | | 1 | |
| Cena | | | | 1 |

Udobje glede prtljage je popotniku močno pomembnejše kot čas potovanja, torej bo podal oceno 5. V primerjavi s prestopi mu je udobje zelo močno pomembnejše in bo zato udobje ocenil z oceno 7 in v primerjavi s ceno absolutno pomembnejši, torej bo podal oceno 9.

Na isti način se odloči glede ostalih parnih primerjav. Čas potovanja mu je v primerjavi s prestopi zmerno pomembnejši – ocena 2, v primerjavi s ceno pa enako pomemben, zato oceni primerjavo z oceno 1.

Število prestopov mu je v primerjavi s ceno zmerno pomembnejše, zato primerjavo oceni z 2. Na podlagi podanih ocen izpolnimo del matrike nad diagonalo primerjav kriterijev samih s seboj:

| | Udobje | Čas | Prestopi | Cena |
|---|---|---|---|---|
| Udobje | 1 | 5 | 7 | 9 |
| Čas | | 1 | 2 | 1 |
| Prestopi | | | 1 | 2 |
| Cena | | | | 1 |

Ko gledamo pomembnost časa potovanja v primerjavi z udobjem glede prtljage, pomeni, da bo ocena pravzaprav obratna vrednost predhodne primerjave udobja glede na čas, torej $\frac{1}{5}$. Prestopi v primerjavi z udobjem glede prtljage bodo imeli oceno $\frac{1}{7}$ in cena glede na udobje s prtljago $\frac{1}{9}$.

V spodnji del matrike pod glavno diagonalo vpišemo torej obratne vrednosti ocen, ki smo jih podali pri primerjavi posameznih kriterijev.

| | Udobje | Čas | Prestopi | Cena |
|---|---|---|---|---|
| Udobje | 1 | 5 | 7 | 9 |
| Čas | $\frac{1}{5}$ | 1 | 2 | 1 |
| Prestopi | $\frac{1}{7}$ | $\frac{1}{2}$ | 1 | 2 |
| Cena | $\frac{1}{9}$ | 1 | $\frac{1}{2}$ | 1 |

Naslednji korak je izračun relativnih uteži oziroma vrednosti kriterijev, ki so pomembni za problem. Vrednosti imenujemo vektor uteži.

Zadnja faza je izvedba testa skladnosti. Namen te faze je preveriti ali obstaja neskladnost glavnih in subjektivnih presoj. Če vrednost kvocienta konsistentnosti ali indeks nekonsistentosti (CR) preseže vrednost 0,1, potem presoje presojevalca niso zanesljive in moramo meritve ponoviti. Doslednost se določi z uporabo lastne vrednosti $\lambda_{\max}$.

Indeks konsistentnosti (CI) in CR izračunamo z uporabo formul:

$$CI = \frac{\lambda_{\max} - \mathrm{n}}{n-1}$$

$$CR = \frac{CI}{RI_n}$$

Kjer je $\lambda_{\max}$ maksimiran lastni vektor matrične primerjave, $RI_n$ je naključen indeks, pridobljen iz spodnje tabele (Samoei, Gichoya, Odero, 2017).

| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RI | 0 | 0 | 0,58 | 0,90 | 1,12 | 1,25 | 1,32 | 1,41 | 1,45 | 1,49 | 1,54 | 1,48 | 1,56 | 1,57 | 1,59 |

Maksimiran lastni vektor $\lambda_{\max}$ dobimo tako, da najprej seštejemo vrednosti posameznih stolpcev v matriki primerjav.

| | Udobje | Čas | Prestopi | Cena |
|---|---|---|---|---|
| Udobje | 1 | 5 | 7 | 9 |
| Čas | $\frac{1}{5}$ | 1 | 2 | 1 |
| Prestopi | $\frac{1}{7}$ | $\frac{1}{2}$ | 1 | 2 |
| Cena | $\frac{1}{9}$ | 1 | $\frac{1}{2}$ | 1 |
| Vsota | 1,45 | 7,5 | 10,50 | 13 |

Vsoto po posameznih kriterijih uporabimo, da dobimo normalizirano matriko primerjav. Posamezne vrednosti so dobljene tako, da vsak element v matriki primerjav delimo z vsoto stolpca, ki smo jo izračunali v prejšnjem koraku.

| | Udobje | Čas | Prestopi | Cena | **Povprečje/Vektor uteži** |
|---|---|---|---|---|---|
| Udobje | $\frac{1}{1,45}=0,07$ | $\frac{5}{7,33}=0,41$ | $\frac{7}{4,53}=0,04$ | $\frac{9}{4,44}=0,03$ | **0,68** |
| Čas | $\frac{0,2}{15,33}=0,02$ | $\frac{1}{7,33}=0,14$ | $\frac{2}{4,53}=0,66$ | $\frac{1}{4,44}=0,08$ | **0,13** |
| Prestopi | $\frac{0,14}{15,33}=0,33$ | $\frac{0,5}{7,33}=0,05$ | $\frac{1}{4,53}=0,22$ | $\frac{2}{4,44}=0,68$ | **0,10** |
| Cena | $\frac{0,11}{15,33}=0,59$ | $\frac{1}{7,33}=0,41$ | $\frac{0,5}{4,53}=0,07$ | $\frac{1}{4,44}=0,23$ | **0,08** |
| Vsota | 1 | 1 | 1 | 1 | |

Vektor uteži posameznih kriterijev je dobljen tako, da izračunamo povprečje posameznih normaliziranih relativnih uteži.

Izračunamo sodbo uteži ($\lambda_n$) za vsako vrstico posebej. Vrstične vektorje faktorjev iz tabele 34 pomnožimo s stolpčnim vektorjem uteži. Ko izračunamo sodbo za vsako vrstico posebej, dobimo nov vektor.

$$\lambda_1 = [1\ \ 5\ \ 7\ \ 9] \times \begin{bmatrix} 0,68 \\ 0,13 \\ 0,10 \\ 0,08 \end{bmatrix} = 0,68 + 0,65 + 0,7 + 0,72 = 2,75$$

$$\lambda_2 = \begin{bmatrix} \frac{1}{5} & 1 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0,68 \\ 0,13 \\ 0,10 \\ 0,08 \end{bmatrix} = 0,136 + 0,13 + 0,2 + 0,08 = 0,546$$

$$\lambda_3 = \begin{bmatrix} \frac{1}{7} & \frac{1}{2} & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 0,68 \\ 0,13 \\ 0,10 \\ 0,08 \end{bmatrix} = 0,097 + 0,065 + 0,1 + 0,16 = 0,422$$

$$\lambda_4 = \begin{bmatrix} \frac{1}{9} & 1 & \frac{1}{2} & 1 \end{bmatrix} \times \begin{bmatrix} 0,68 \\ 0,13 \\ 0,10 \\ 0,08 \end{bmatrix} = 0,076 + 0,13 + 0,05 + 0,08 = 0,336$$

Vsako vrstico vektorja sodbe delimo s pripadajočim členom vektorja uteži, da dobimo vektor mere skladnosti.

| | Vektor uteži | Sodba uteži | Mera skladnosti |
|---|---|---|---|
| Udobje | 0,68 | 2,75 | $\frac{2,75}{0,68} = 4,05$ |
| Čas | 0,13 | 0,546 | $\frac{0,546}{0,13} = 4,2$ |
| Prestopi | 0,10 | 0,422 | $\frac{0,422}{0,1} = 4,22$ |
| Cena | 0,08 | 0,336 | $\frac{0,336}{0,08} = 4,2$ |

Oceno $\lambda_{\max}$ izračunamo iz povprečja mer skladnosti:

$$\lambda_{\max} = \frac{4,05 + 4,2 + 4,22 + 4,2}{4} = \frac{16,67}{4} = 4,17$$

Izračunamo indeks konsistentnosti $CI$, ki smo ga definirali prej. Oceno $\lambda_{\max}$ smo izračunali, število faktorjev $n$, ki smo jih uporabili je 4. $RI_n$ razberemo iz tabele naključnega indeksa $RI$.

$$CI = \frac{4,17 - 4}{4 - 1} = \frac{0,17}{3} = 0,057$$

$$CR = \frac{0,057}{0,90} = 0,063$$

Ker je vrednost $CR$, ki smo jo izračunali manjša od 0,1 pomeni, da so ocene ocenjevalca konsistentne in jih lahko uporabimo v nadaljnji analizi.

Naredimo še razvrstitev faktorjev glede na izračunan vektor uteži:

| | Povprečje | **Rang** |
|---|---|---|
| Udobje | 0,68 | **1** |
| Čas | 0,13 | **2** |
| Prestopi | 0,10 | **3** |
| Cena | 0,08 | **4** |

Postopek AHP se seveda ne konča z razvrstitvijo osnovnih kriterijev, vendar moramo opraviti isti postopek še za morebitne podkriterije, ki smo si jih postavili. V predstavljenem primeru smo podrobneje razdelali kriterij PRESTOPI. Postopek izračuna vektorja uteži je popolnoma enak kot smo ga opisali, zato bomo v tem delu predstavili še, kako se takšnega izračuna lotimo v programu Excel. Ocene popotnika predstavljamo v spodnji tabeli, ki smo jo izdelali v programu Excel.

Slika 66: Ocene popotnika

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | **Podkriterij 1** | **Podkriterij 2** | **Podkriterij 3** |
| 2 | | **Podkriterij 1** | 1 | 0,13 | 0,33 |
| 3 | | **Podkriterij 2** | | 1 | 2 |
| 4 | | **Podkriterij 3** | | | 1 |
| 5 | | **Vsota** | | | |
| 6 | | | | | |

Program izračuna nasportne vrednosti v tabeli, če vnesemo formulo $\frac{1}{a_i}$, pri čemer $a_i$ predstavlja posamične ocene.

Slika 67: Izračun ocen pod glavno diagonalo

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | **Podkriterij 1** | **Podkriterij 2** | **Podkriterij 3** |
| 2 | | **Podkriterij 1** | 1 | 0,13 | 0,33 |
| 3 | | **Podkriterij 2** | =1/D2 | 1 | 2 |
| 4 | | **Podkriterij 3** | =1/E2 | =1/E3 | 1 |
| 5 | | **Vsota** | | | |
| 6 | | | | | |

Ko vnesemo formulo v vse prazne celice pod diagonalo primerjav podkriterijev s samim seboj, imamo izpolnjeno matriko primerjav.

Slika 68: Matrika primerjav

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 |
| 3 | | Podkriterij 2 | 7,692307692 | 1 | 2 |
| 4 | | Podkriterij 3 | 3,03030303 | 0,5 | 1 |
| 5 | | Vsota | | | |

Seštevek vsakega stolpca dobimo s funkcijo $SUM$:

Slika 69: Seštevek po stolpcih

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 |
| 3 | | Podkriterij 2 | =1/D2 | 1 | 2 |
| 4 | | Podkriterij 3 | =1/E2 | =1/E3 | 1 |
| 5 | | Vsota | =SUM(C2:C4) | =SUM(D2:D4) | =SUM(E2:E4) |

Za izračun potrebujemo novo tabelo z normaliziranimi relativnimi utežmi. Posamezne vrednosti so dobljene tako, da vsak element v matriki relativnih vrednosti delimo z vsoto stolpca, ki smo jo izračunali v prejšnjem koraku.

Slika 70: Povprečja podkriterijev

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 |
| 3 | | Podkriterij 2 | =1/D2 | 1 | 2 |
| 4 | | Podkriterij 3 | =1/E2 | =1/E3 | 1 |
| 5 | | Vsota | =SUM(C2:C4) | =SUM(D2:D4) | =SUM(E2:E4) |
| 6 | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 |
| 8 | | Podkriterij 1 | =C2/$C$5 | =D2/$D$5 | =E2/$E$5 |
| 9 | | Podkriterij 2 | =C3/$C$5 | =D3/$D$5 | =E3/$E$5 |
| 10 | | Podkriterij 3 | =C4/$C$5 | =D4/$D$5 | =E4/$E$5 |
| 11 | | Vsota | =SUM(C8:C10) | =SUM(D8:D10) | =SUM(E8:E10) |

Vektor uteži posameznih kriterijev je dobljen tako, da izračunamo povprečje posameznih normaliziranih relativnih uteži. V Excelu to dobimo s formulo $AVERAGE$ na celi vrstici.

Slika 71: Povprečje vrstic

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 | |
| 3 | | Podkriterij 2 | =1/D2 | 1 | 2 | |
| 4 | | Podkriterij 3 | =1/E2 | =1/E3 | 1 | |
| 5 | | Vsota | =SUM(C2:C4) | =SUM(D2:D4) | =SUM(E2:E4) | |
| 6 | | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje |
| 8 | | Podkriterij 1 | =C2/$C$5 | =D2/$D$5 | =E2/$E$5 | =AVERAGE(C8:E8) |
| 9 | | Podkriterij 2 | =C3/$C$5 | =D3/$D$5 | =E3/$E$5 | =AVERAGE(C9:E9) |
| 10 | | Podkriterij 3 | =C4/$C$5 | =D4/$D$5 | =E4/$E$5 | =AVERAGE(C10:E10) |
| 11 | | Vsota | =SUM(C8:C10) | =SUM(D8:D10) | =SUM(E8:E10) | |
| 12 | | | | | | |

Izračunamo sodbo uteži ($\lambda_n$) za vsako vrstico posebej. Vrstične vektorje ocen posameznih faktorjev pomnožimo s stolpčnim vektorjem uteži (stolpec povprečje). Ko izračunamo sodbo za vsako vrstico posebej, dobimo nov vektor.

Slika 72: Pridobitev vektorja sodbe

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | | | |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 | | | |
| 3 | | Podkriterij 2 | =1/D2 | 1 | 2 | | | |
| 4 | | Podkriterij 3 | =1/E2 | =1/E3 | 1 | | | |
| 5 | | Vsota | =SUM(C2:C4) | =SUM(D2:D4) | =SUM(E2:E4) | | | |
| 6 | | | | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje | Sodba | Mere skladnosti |
| 8 | | Podkriterij 1 | =C2/$C$5 | =D2/$D$5 | =E2/$E$5 | =AVERAGE(C8:E8) | =MMULT(C2:E2;$F$8:$F$10) | |
| 9 | | Podkriterij 2 | =C3/$C$5 | =D3/$D$5 | =E3/$E$5 | =AVERAGE(C9:E9) | =MMULT(C3:E3;$F$8:$F$10) | |
| 10 | | Podkriterij 3 | =C4/$C$5 | =D4/$D$5 | =E4/$E$5 | =AVERAGE(C10:E10) | =MMULT(C4:E4;$F$8:$F$10) | |
| 11 | | Vsota | =SUM(C8:C10) | =SUM(D8:D10) | =SUM(E8:E10) | | | |
| 12 | | | | | | | | |

Vsako vrstico vektorja sodbe delimo s pripadajočim členom vektorja uteži, da dobimo vektor mere skladnosti.

Slika 73: Izračun mere skladnosti

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | | | |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 | | | |
| 3 | | Podkriterij 2 | =1/D2 | 1 | 2 | | | |
| 4 | | Podkriterij 3 | =1/E2 | =1/E3 | 1 | | | |
| 5 | | Vsota | =SUM(C2:C4) | =SUM(D2:D4) | =SUM(E2:E4) | | | |
| 6 | | | | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje | Sodba | Mere skladnosti |
| 8 | | Podkriterij 1 | =C2/$C$5 | =D2/$D$5 | =E2/$E$5 | =AVERAGE(C8:E8) | =MMULT(C2:E2;$F$8:$F$10) | =G8/F8 |
| 9 | | Podkriterij 2 | =C3/$C$5 | =D3/$D$5 | =E3/$E$5 | =AVERAGE(C9:E9) | =MMULT(C3:E3;$F$8:$F$10) | =G9/F9 |
| 10 | | Podkriterij 3 | =C4/$C$5 | =D4/$D$5 | =E4/$E$5 | =AVERAGE(C10:E10) | =MMULT(C4:E4;$F$8:$F$10) | =G10/F10 |
| 11 | | Vsota | =SUM(C8:C10) | =SUM(D8:D10) | =SUM(E8:E10) | | | |

Oceno $\lambda_{\max}$ izračunamo iz povprečja mer skladnosti. Ocena predstavlja povprečje vseh izračunanih mer skladnosti.

Slika 74: Izračun $\lambda_{\max}$

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | | | |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 | | | |
| 3 | | Podkriterij 2 | 7,692 | 1 | 2 | | | |
| 4 | | Podkriterij 3 | 3,03 | 0,5 | 1 | | | |
| 5 | | Vsota | 11,72 | 1,63 | 3,33 | | | |
| 6 | | | | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje | Sodba | Mere skladnosti |
| 8 | | Podkriterij 1 | 0,085 | 0,0798 | 0,099 | 0,088 | 0,264 | 3,002 |
| 9 | | Podkriterij 2 | 0,656 | 0,6135 | 0,601 | 0,623 | 1,878 | 3,012 |
| 10 | | Podkriterij 3 | 0,259 | 0,3067 | 0,3 | 0,289 | 0,867 | 3,005 |
| 11 | | Vsota | 1 | 1 | 1 | | | |
| 12 | | | | | | | | |
| 13 | | | $\lambda_{max}$ | =AVERAGE(H8:H10) | | | | |
| 14 | | | RI | | | | | |
| 15 | | | CI | | | | | |
| 16 | | | CR | | | | | |

Izračunamo indeks konsistentnosti $CI$, ki smo ga definirali prej. Uporabimo formulo za izračun indeksa $CI$.

Slika 75: Indeks konsistentnosti

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | | | |
| 2 | | Podkriterij 1 | 1 | 0,13 | 0,33 | | | |
| 3 | | Podkriterij 2 | 7,692 | 1 | 2 | | | |
| 4 | | Podkriterij 3 | 3,03 | 0,5 | 1 | | | |
| 5 | | Vsota | 11,72 | 1,63 | 3,33 | | | |
| 6 | | | | | | | | |
| 7 | | | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje | Sodba | Mere skladnosti |
| 8 | | Podkriterij 1 | 0,085 | 0,0798 | 0,099 | 0,088 | 0,264 | 3,002 |
| 9 | | Podkriterij 2 | 0,656 | 0,6135 | 0,601 | 0,623 | 1,878 | 3,012 |
| 10 | | Podkriterij 3 | 0,259 | 0,3067 | 0,3 | 0,289 | 0,867 | 3,005 |
| 11 | | Vsota | 1 | 1 | 1 | | | |
| 12 | | | | | | | | |
| 13 | | | $\lambda_{max}$ | 3,0092 | | | | |
| 14 | | | RI | 0,58 | | | | |
| 15 | | | CI | =(D13-3)/2 | | | | |
| 16 | | | CR | | | | | |

Slika 76: Kvocient konsistentnosti

| | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 |
|---|---|---|---|
| Podkriterij 1 | 1 | 0,13 | 0,33 |
| Podkriterij 2 | 7,692 | 1 | 2 |
| Podkriterij 3 | 3,03 | 0,5 | 1 |
| Vsota | 11,72 | 1,63 | 3,33 |

| | Podkriterij 1 | Podkriterij 2 | Podkriterij 3 | Povprečje | Sodba | Mere skladnosti |
|---|---|---|---|---|---|---|
| Podkriterij 1 | 0,085 | 0,0798 | 0,099 | 0,088 | 0,264 | 3,002 |
| Podkriterij 2 | 0,656 | 0,6135 | 0,601 | 0,623 | 1,878 | 3,012 |
| Podkriterij 3 | 0,259 | 0,3067 | 0,3 | 0,289 | 0,867 | 3,005 |
| Vsota | 1 | 1 | 1 | | | |

| | |
|---|---|
| $\lambda_{max}$ | 3,0092 |
| RI | 0,58 |
| CI | 0,004608 |
| CR | =D15/D14 |

Ker je vrednost $CR$, ki smo jo izračunali (0,007945) manjša od 0,1 pomeni, da so ocene ocenjevalca konsistentne in jih lahko uporabimo v nadaljnji analizi.

Naredimo še razvrstitev podfaktorjev glede na izračunan vektor uteži (v Excelu smo ga poimenovali kar povprečje):

| | Lastni vektor | **Rang** |
|---|---|---|
| Podkriterij 1 | 0,09 | **3** |
| Podkriterij 2 | 0,63 | **1** |
| Podkriterij 3 | 0,29 | **2** |

Globalne uteži podkriterijev sedaj izračunamo s pomočjo uteži ustreznega kriterija.

$$Globalna\ utež\ podkriterija = utež\ kriterija\ \times utež\ podkriterija$$

Spomnimo, da smo za kriterij prestopi izračunali utež $0{,}10$.

| | **Lastni vektor** | **Globalni vektor** |
|---|---|---|
| Podkriterij 1 | 0,09 | $0{,}09 \cdot 0{,}10 = 0{,}009$ |
| Podkriterij 2 | 0,63 | $0{,}63 \cdot 0{,}10 = 0{,}063$ |
| Podkriterij 3 | 0,29 | $0{,}29 \cdot 0{,}10 = 0{,}029$ |

Da bi se popotnik lahko dokončno odločil, mora podati še svojo subjektivno oceno glede alternativ prevoza. Matriko relativnih vrednosti smo sestavili iz njegovih ocen:

| | Avto | Vlak | Taksi |
|---|---|---|---|
| Avto | 1,00 | 6,00 | 9,00 |
| Vlak | 0,17 | 1,00 | 2,00 |
| Taksi | 0,11 | 0,50 | 1,00 |

Ker je postopek pridobitve lastnega vektorja podrobno opisan dvakrat, tokrat podajamo samo rezultate lastnega vektorja. Indeks skladnosti ocen alternativ pa znaša 0,007955, torej so ocene popotnika konsistentne:

| | **Lastni vektor** | **Rang** |
|---|---|---|
| Avto | 0,78 | **1** |
| Vlak | 0,14 | **2** |
| Taksi | 0,08 | **3** |

Zadnji korak AHP analize je ocena preferenčnega indeksa (skrajšano PR) na osnovi vektorjev uteži, ki smo jih izračunali. Želimo vedeti, katera izmed alternativ za prevoz do letališča, bo za potnika najbolj primerna glede na njegovo subjektivno oceno alternativ in kriterijev pri odločanju.

$PR$ izračunamo tako, da pomnožimo vektor uteži izbranih alternativ z vektorjem uteži podkriterijev (oziroma neposredno kriterijev, če kriterijev nismo delili naprej na podkriterije), rezultate pa nato seštejemo.

| **Kriterij** | Podkriterij | Utež | Avto | Vlak | Taksi |
|---|---|---|---|---|---|
| **Udobje** | | 0,68 | $\mathbf{0,78 \cdot 0,68 =}$ | $\mathbf{0,14 \cdot 0,68 =}$ | $\mathbf{0,08 \cdot 0,68 =}$ |
| **Čas** | | 0,13 | $\mathbf{0,78 \cdot 0,13 =}$ | $\mathbf{0,14 \cdot 0,13 =}$ | $\mathbf{0,08 \cdot 0,13 =}$ |
| **Prestopi** | Podkriterij 1 | 0,009 | $\mathbf{0,78 \cdot 0,009 =}$ | $\mathbf{0,14 \cdot 0,009 =}$ | $\mathbf{0,08 \cdot 0,009 =}$ |
| | Podkriterij 2 | 0,063 | $\mathbf{0,78 \cdot 0,063 =}$ | $\mathbf{0,14 \cdot 0,063 =}$ | $\mathbf{0,08 \cdot 0,063 =}$ |
| | Podkriterij 3 | 0,029 | $\mathbf{0,78 \cdot 0,029 =}$ | $\mathbf{0,14 \cdot 0,029 =}$ | $\mathbf{0,08 \cdot 0,029 =}$ |
| **Cena** | | 0,08 | $\mathbf{0,78 \cdot 0,08 =}$ | $\mathbf{0,14 \cdot 0,08 =}$ | $\mathbf{0,08 \cdot 0,08 =}$ |

| Kriterij | **Podkriterij** | **Utež** | **Avto** | **Vlak** | **Taksi** |
|---|---|---|---|---|---|
| **Udobje** | | 0,68 | 0,53 | 0,095 | 0,054 |
| **Čas** | | 0,13 | 0,101 | 0,018 | 0,01 |
| **Prestopi** | Podkriterij 1 | 0,009 | 0,007 | 0,001 | 0,0007 |
| | Podkriterij 2 | 0,063 | 0,049 | 0,009 | 0,005 |
| | Podkriterij 3 | 0,029 | 0,023 | 0,004 | 0,002 |
| **Cena** | | 0,08 | 0,062 | 0,011 | 0,006 |
| Vsota | | | **0,773** | **0,139** | **0,079** |

Vsota predstavlja preferenčni indeks posameznih alternativ, ki so na voljo. Preferenčne indekse razvrstimo po velikosti od največjega do najmanjšega:

| **Alternativa** | **PR** | **Rang** |
|---|---|---|
| Avto | 0,773 | **1** |
| Vlak | 0,139 | **2** |
| Taksi | 0,079 | **3** |

Popotnikova najboljša izbira prevoza do letališča glede na njegovo subjektivno mnenje je avtomobil. Najmanj verjetno pa bi se odločil za taksi prevoz.

## 3.2 Več različnih odločevalcev

V prejšnjem poglavju je predstavljen izpeljan postopek AHP analize za enega odločevalca. Pri znanstvenih raziskavah je v večini primerov vključenih več odločevalcev. To pomeni, da je potrebno ocene, ki so jih posamezni ocenjevalci podali, združiti v eno samo. Ostanimo pri prej opisanem primeru, vendar recimo, da se popotnik sedaj odpravlja na pot še z dvema prijateljema in se skupaj odločajo za izbiro prevoznega sredstva do letališča. V tem poglavju bomo prikazali samo, kako iz posamičnih odločitev treh prijateljev pridemo do ene matrike relativnih vrednosti.

Prijatelji so vsak zase ocenili kako pomembni so zanje faktorji pri odločitvi za prevoz. Njihove ocene so zbrane v naslednjih 3 tabelah.

Ocene prvega popotnika:

| | **Udobje** | **Čas** | **Prestopi** | **Cena** |
|---|---|---|---|---|
| **Udobje** | 1 | 5 | 7 | 9 |
| **Čas** | $\frac{1}{5}$ | 1 | 2 | 1 |
| **Prestopi** | $\frac{1}{7}$ | $\frac{1}{2}$ | 1 | 2 |
| **Cena** | $\frac{1}{9}$ | 1 | $\frac{1}{2}$ | 1 |

Ocene drugega popotnika:

| | **Udobje** | **Čas** | **Prestopi** | **Cena** |
|---|---|---|---|---|
| **Udobje** | 1 | 2 | $\frac{1}{4}$ | $\frac{1}{2}$ |
| **Čas** | $\frac{1}{2}$ | 1 | 5 | $\frac{1}{8}$ |
| **Prestopi** | 4 | $\frac{1}{5}$ | 1 | $\frac{1}{2}$ |
| **Cena** | 2 | 8 | 2 | 1 |

Ocene tretjega popotnika:

| | **Udobje** | **Čas** | **Prestopi** | **Cena** |
|---|---|---|---|---|
| **Udobje** | 1 | 8 | 5 | $\frac{1}{8}$ |
| **Čas** | $\frac{1}{8}$ | 1 | 1 | $\frac{1}{6}$ |
| **Prestopi** | $\frac{1}{5}$ | 1 | 1 | $\frac{1}{8}$ |
| **Cena** | 8 | 6 | 8 | 1 |

Da bi dobili enotno matriko relativnih vrednosti, je treba za vsako primerjavo (vsako celico v tabeli) izračunati geometrijsko sredino vseh ocen – v konkretnem primeru geometrijsko sredino treh ocen. Seveda pa lahko ta postopek izvedemo za poljubno število odločevalcev, ki so ocenjevali enake stvari.

Geometrijska sredina se izračuna po formuli:

$$(\prod_{i=1}^{n} a_i)^{\frac{1}{n}} = (a_1 \cdot a_2 \cdot \cdots \cdot a_n)^{\frac{1}{n}} = \sqrt[n]{a_1 \cdot a_2 \cdot \cdots \cdot a_n}$$

pri čemer $a_i$ predstavlja vrednosti posamičnih ocen, $n$ pa število ocenjevalcev.

| | **Udobje** | **Čas** | **Prestopi** | **Cena** |
|---|---|---|---|---|
| **Udobje** | $\sqrt[3]{1 \cdot 1 \cdot 1} = 1$ | $\sqrt[3]{5 \cdot 2 \cdot 8} = 4{,}31$ | $\sqrt[3]{7 \cdot \frac{1}{4} \cdot 5} = 2{,}06$ | $\sqrt[3]{9 \cdot \frac{1}{2} \cdot \frac{1}{8}} = 0{,}83$ |
| **Čas** | $\sqrt[3]{\frac{1}{5} \cdot \frac{1}{2} \cdot \frac{1}{5}} = 0{,}23$ | $\sqrt[3]{1 \cdot 1 \cdot 1} = 1$ | $\sqrt[3]{2 \cdot 5 \cdot 1} = 2{,}15$ | $\sqrt[3]{1 \cdot \frac{1}{8} \cdot \frac{1}{6}} = 0{,}28$ |
| **Prestopi** | $\sqrt[3]{\frac{1}{7} \cdot 4 \cdot \frac{1}{5}} = 4{,}9$ | $\sqrt[3]{\frac{1}{2} \cdot \frac{1}{5} \cdot 1} = 0{,}46$ | $\sqrt[3]{1 \cdot 1 \cdot 1} = 1$ | $\sqrt[3]{2 \cdot \frac{1}{2} \cdot \frac{1}{8}} = 0{,}5$ |
| **Cena** | $\sqrt[3]{\frac{1}{9} \cdot 2 \cdot 8} = 1{,}21$ | $\sqrt[3]{1 \cdot 8 \cdot 6} = 3{,}6$ | $\sqrt[3]{\frac{1}{2} \cdot 2 \cdot 8} = 2$ | $\sqrt[3]{1 \cdot 1 \cdot 1} = 1$ |

Tabela z izračunanimi geometrijskimi sredinami predstavlja skupno matriko primerjav, ki jo uporabimo dalje v postopku AHP analize, ki je bila podrobno predstavljena v tem poglavju.

| | **Udobje** | **Čas** | **Prestopi** | **Cena** |
|---|---|---|---|---|
| **Udobje** | 1 | 4,31 | 2,06 | 0,83 |
| **Čas** | 0,27 | 1 | 2,15 | 0,28 |
| **Prestopi** | 4,89 | 0,46 | 1 | 0,5 |
| **Cena** | 1,21 | 3,6 | 2 | 1 |

## 3.3 AHP analiza s pomočjo programa AHP software

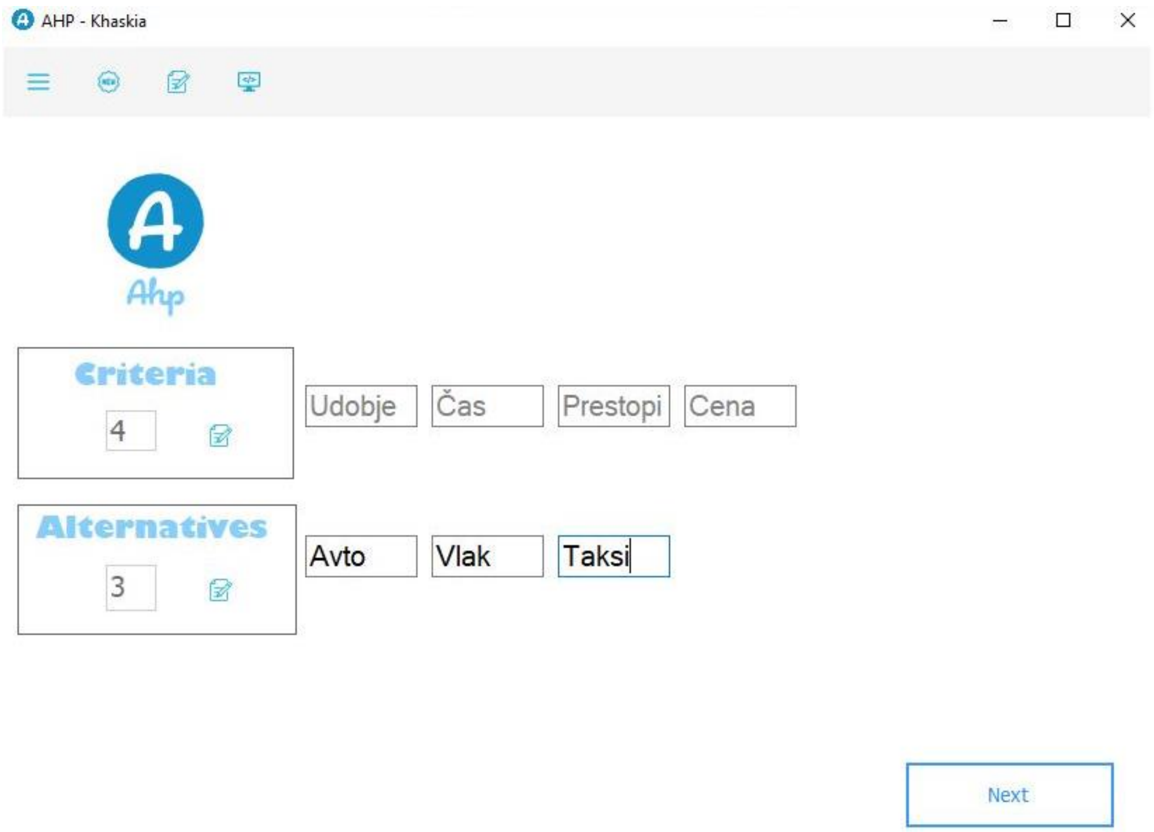
Tudi za AHP analizo obstaja veliko plačljivih in brezplačnih programov in spletnih orodij, ki omogočajo relativno preprosto analizo rezultatov. V tem poglavju je predstavljen AHP software .

Ob odprtju programa se odpre nov projekt, kamor vnesemo željeno število kriterijev[32] in željeno število alternativ[33]. Program ustvari temu ustrezno število praznih okenc za posamezno postavko. Vsako okence ima svoje ime, ki pa ga lahko po želji spremenimo in vpišemo svoje.

Za prikaz delovanja programa bomo uporabili primer, ki je predstavljen na začetku tega poglavja.

[32] criteria
[33] alternatives

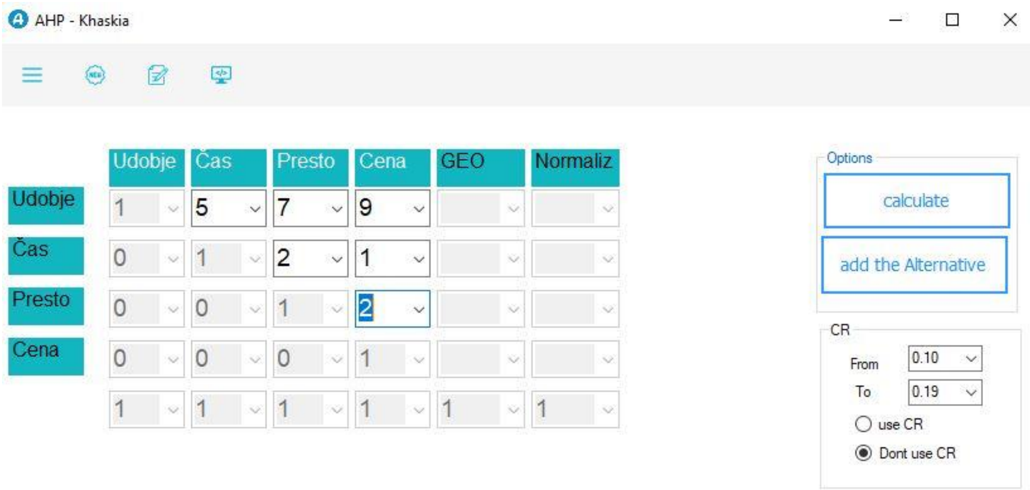
Slika 77: Priprava za AHP analizo v AHP software




Ob kliku na gumb Naprej[34] se odpre matrika primerjav za vnešene kriterije. Izpolnimo samo del matrike nad glavno diagonalo, vrednosti pod glavno diagonalo se bodo preračunale same. Po želji lahko dodamo interval na katerem naj se nahaja vrednost *CR*, da nas program, v primeru prevelike vrednosti opozori, da ocene niso konsistentne.

[34] Next

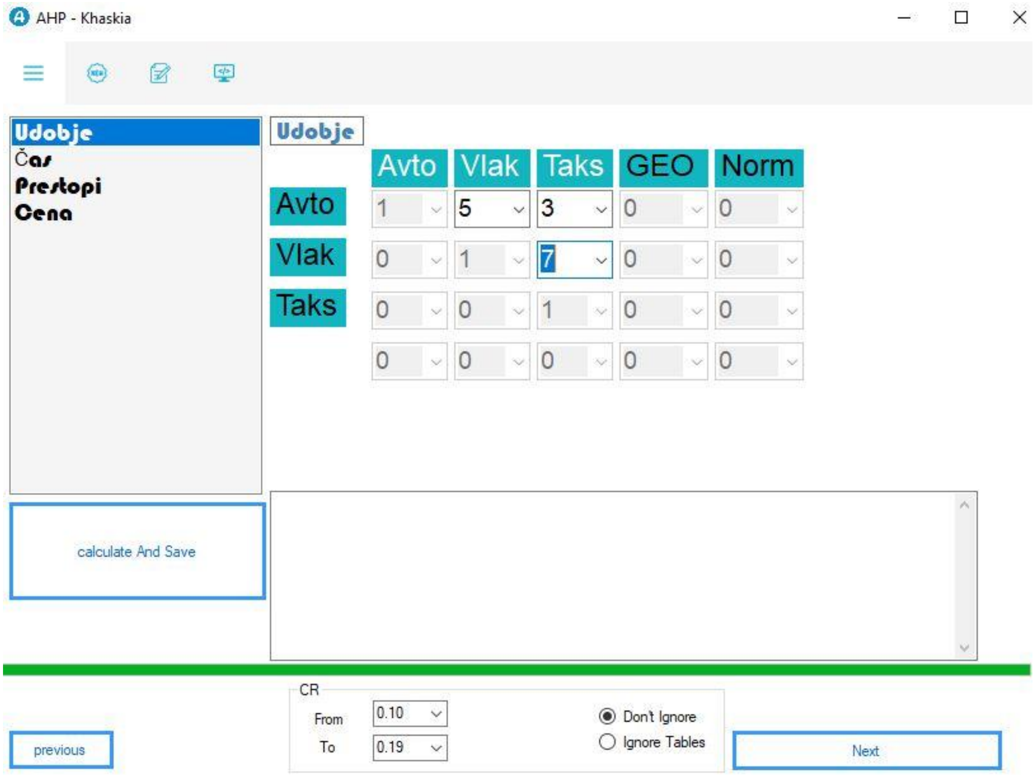
Slika 78: Vnos ocen kriterijev v matriko primerjav




Ob kliku na gumb Dodaj alternativo[35] na isti način vnesemo še ocene primerjav glede alternativ, ki so na voljo. Ko je matrika izpolnjena, kliknemo na Izračunaj in shrani[36]. Ponovimo za vse kriterije, ki smo jih vnesli in na koncu kliknemo na Naprej. Tudi tukaj program nudi možnost spretnega preverjanja vrednosti *CR* in v primeru, da ta ni v dovoljenih mejah, lahko kako sodbo izključimo iz končnega izračuna.

[35] Add the alternative
[36] Calculate and Save

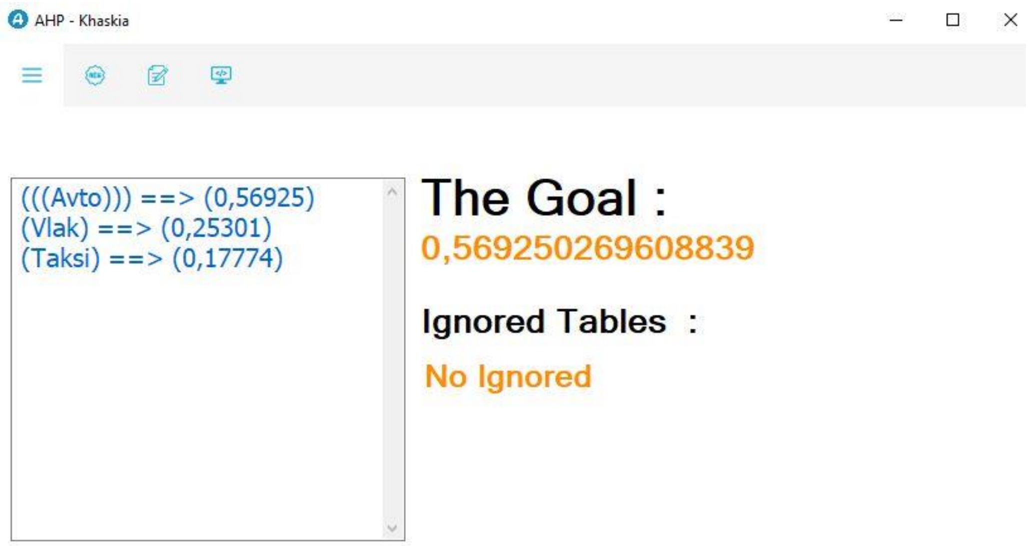
Slika 79: Vnos ocen alternativ v matriko primerjav




Da bi lahko uspešno uporabili ta program, je anketni vprašalnik oziroma raziskavo potrebno načrtovati nekoliko drugače, kot pri prej opisanem postopku. Udeleženci morajo alternative, ki so na voljo, oceniti na podlagi vsakega faktorja zase. Torej, kako bi se na primer odločali med alternativami ob upoštevanju udobja, ki ga nudi vsaka izmed alternativ.

Ko so vnesene vse ocene, kliknemo Naprej in program izpiše poročilo s preferenčnimi indeksi za vse alternative.

Slika 80: Rezultati AHP analize v programu AHP software




Vrednosti preferenčnih indeksov se razlikujejo od tistih, ki smo jih dobili z reševanjem v programu Excel, saj je bila za pridobitev rezultatov uporabljena drugačna metoda (ocenjevanje alternativ na podlagi posameznih faktorjev odločitve). Pri njeni uporabi se moramo zavedati, da bo vprašalnik za ocenjevalce nekoliko daljši in da zaradi tega morda rezultati ne bodo najbolj realni.

# 4 UPORABA NABORA PODATKOV V LINGO PROGRAMU

V prvem delu gradiva Linearnega programiranja v logistiki smo pokazali osnovne funkcije in uporabo programa LINGO. V primeru linearnih programov z več spremenljivkami oziroma več omejitvami, je bolj kakor vpisovanje osnovnega linearnega programa kot ga obravnava prvi del gradiva, smiselno program zgraditi s pomočjo nabora podatkov.

Spodaj je predstavljen takšen način zapisa na podlagi primera ujemajočih se parov iz poglavja Problem kombiniranja ujemajočih se parov, pretvorjen na problem maksimalnega pretoka.

```
MODEL:
SETS:
NODES/Izvor,Bojan,Dejan,Tomaz,Miha,Luka,Tina,Monika,Katari
na,Lina,Viktorija,Ponor/;
ARCS(NODES,NODES)/Izvor,Bojan Izvor,Dejan, Izvor,Tomaz,
Izvor,Miha Izvor,Luka
Bojan,Monika Dejan,Tina Tomaz,Tina Tomaz,Monika Miha,Tina
Miha,Monika Miha,Viktorija Luka,Katarina Luka,Lina
Luka,Viktorija
Tina,Ponor Monika,Ponor Katarina,Ponor Lina,Ponor
Viktorija,Ponor Ponor,Izvor/
:CAP,FLOW;
ENDSETS
MAX= FLOW (@index(Ponor),@index(Izvor));
@FOR(ARCS(I,J):FLOW(I,J)<CAP(I,J));
@FOR(NODES(I):@SUM(ARCS(J,I):FLOW(J,I))
=@SUM(ARCS(I,J):FLOW(I,J)));
DATA:
CAP=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1000;
ENDDATA
END
```

Odsek z modelom se začne s ključno besedo `MODEL:` in se zaključi z `END`. Vanj vpišemo celotni linearni program kot je opisano v nadaljevanju.

Odsek z naborom podatkov se začne s ključno besedo `SETS:` in se konča z `ENDSETS`. V tem odseku definiramo vsa vozlišča in obstoječe povezave problema.

V danem primeru smo definirali vozlišča (NODES) in povezave med vozlišči (ARCS). CAP in FLOW predstavljata kapacitete in odločitvene spremenljivke.

V vrstici `MAX= FLOW (@index(Ponor),@index(Izvor);` smo zapisali, da želimo poiskati maksimalno vrednost odločitvene spremenljivke, ki predstavlja povezavo Ponor-Izvor. To je namenska funkcija problema.

Omejitve so zapisane v: `@FOR(ARCS(I,J):FLOW(I,J)<CAP(I,J));`
`@FOR(NODES(I):@SUM(ARCS(J,I):FLOW(J,I))`
`=@SUM(ARCS(I,J):FLOW(I,J)));`
Kjer za vsako povezavo upoštevamo, da je pretok manjši ali enak kapaciteti povezave in za vsako vozlišče upošteva, da je tok v vozlišče enak toku iz vozlišča.

Podatkovni odsek se začne s ključno besedo `DATA:` in se konča s ključno besedo `ENDDATA` v vrstici samo po sebi. V ta odsek uvrstimo podatke o problemu, npr. kapaciteto povezav, povpraševanje/ponudbo v vozliščih, ceno povezav in podobno. V danem primeru so vse kapacitete povezav 1, saj lahko povežemo v par le dve osebi.

### 4.1.1 Uporaba zank

Program omogoča tudi uporabo nekaterih zank, ki poenostavijo pisanje linearnega programa. Te zanke omogočajo, da izvedemo določeno operacijo na vseh komponentah, ki smo jih definirali v odseku nabora podatkov, ne da bi morali pisati funkcijo za vsako izmed njih.

| Funkcija | Uporaba |
|---|---|
| @FOR | v osnovi uporabna za ponavljanje in uporabo omejitev za vse podatke vključene v linearni program |
| @SUM | funkcija izvede operacijo seštevanja na celem naboru podatkov |

# 5 SLOVENSKO-ANGLEŠKI SLOVARČEK UPORABLJENIH POJMOV

| **Slovenski prevod** | **Angleški izraz** |
|---|---|
| alternative | alternatives |
| Analitični-hierarhični proces | Analytic hierarchy process |
| analiza občutljivosti | Range report |
| Analiza ovojnice podatkov | Data Envelopment Analysis |
| celica s cilji | Objective Cell |
| celice s spremenljivkami | Variable Cells |
| dodaj alternativo | Add the alternative |
| dopolnilna spremenljivka | Slack or Surplus |
| izhodno vozlišče | Demand point |
| izračunaj in shrani | Calculate and Save |
| kriterij | Criteria |
| model vozlišča in lokalnih povezav | Hub and Spoke model |
| namenska funkcija | Objective function |
| naprej | Next |
| odločitvene enote | Decision-making units |
| omejitev | Constraint |
| poročilo o rešitvi | Solution Report |
| pretovorna vozlišča | Trans-shippment point |
| problem asignacije | Assignment Problem |
| problem dodeljevanja virov | Assignment flow problem |
| problem maksimalnega pretoka | Maximum flow problem |
| problem najkrajše poti | Shortest path problem |
| problem pretovora | Transshipment problem |
| problem omrežnega pretoka | Network flow problem |
| reducirani strošek | Reduced Cost |
| senčna cena/dualna cena | Shadow price / Marginal value / Pi value / Dual value |
| transportni problem | Transportation problem |

| | |
|---|---|
| vhodno vozlišče | Supply point |
| vir | Resource |

# SEZNAM LITERATURE


Bradley, S. P., Hax, A. C., & Magnanti, T. L. (1977). *Applied mathematical programming*. Minnesota: Addison-Wesley.

Charnes, A., Cooper, W. W. & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research, 2*(4), 429 - 444.

*DEA, home page.* (b. d.). Najdeno 5. maja 2019 na spletni strani: http://www.etm.pdx.edu/dea/homedea.html

Kramberger, T. (2010). *Problem kitajskega poštarja s prioritetnimi vozlišči*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko.

Kuosmanen, T., Cherchye, L.,Sipilaäinen, T. (b. d.). *The law of one price in data envelopment analysis: restricting weight flexibility across firms.* Najdeno 10. aprila 2020 na spletni strani: https://econwpa.ub.uni-muenchen.de/econ-wp/mic/papers/0312/0312006.pdf

Lindo Systems Inc. (2018). *LINGO The Modeling Language and Optimizer.* Lindo Systems Inc. Chicago: Illinois.

O'Kelly, M.E. (1987). A Quadratic Integer Program for the Location of Interacting Hub Facilities. *European Journal of Operational Research*, 32, 393-404.

Nocedal, J. in Wright, S. J. (1999). *Numerical optimization*. ZDA: Springer.

Ramanathan, R. (2003). *An Introduction to data envelopment analysis: A tool for performance measurement*. New Delhi: SAGE Publications.

Saaty, T.L. (1980). *The Analytic Hierarchy Process*. New York: McGraw Hill.

Samoei, D., Gichoya, D., Odero, D. (2017). An AHP variation modelling approach for perfomance measurement of suppliers of information communication technology product. *International Journal of the Analytic Hierarchy Process*, *9*(2).

Sherman, H. D., Zhu, J. (2006). I*mproving Service Performance using Data envelopment Analysis (DEA).* ZDA: Springer Science+Business Media.

Thore, S. & Iannone, F. (2005). *The hub-and-spoke model.* Najdeno 10. maja 2019 na spletni strani: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=950753