mr.sc. Bojan Kovačić

Operacijska istraživanja

skripta

Zagreb, srpanj 2008.

Predgovor

Ova skripta obuhvaća drugi dio modula *Operacijska istraživanja* koji se predaje na završnoj, 3. godini stručnoga studija logistike na Visokoj poslovnoj školi BA Nordhessen u Bad Wildungenu. Dok se u prvom dijelu modula pozornost usmjerava na linearno programiranje i njegove primjene, poglavito na transportni problem, u drugom se dijelu obrađuju osnovna načela i principi dinamičkoga programiranja, te njegovih primjena na rješavanje različitih praktičnih problema iz područja logistike, ekonomije itd.

Koncipirajući nastavu iz ovoga dijela modula, opredijelio sam se za pristup donekle različit od uobičajenoga pristupa ovoj problematici. Naime, smatrao sam da je za studente stručnoga studija logistike najvažnije da u okviru ovoga predmeta savladaju matematičko modeliranje određenih ekonomskih i poslovnih procesa, njihovo rješavanje pomoću nekih edukacijskih računalnih programa i interpretaciju dobivenih rezultata. Izbor primjerenih računalnih programa prilagodio sam predviđenom gradivu i predznanju studenata iako nesumnjivo postoje daleko bolji (ali i skuplji!) računalni programi koji još uspješnije rješavaju razmatrane probleme. Također, gotovo u potpunosti sam izostavio relativno složen matematički aparat ("teoriju") potreban za precizno formalno definiranje načela dinamičkoga programiranja, ali sam zato nastojao pojasniti osnovne ideje i načela na nizu detaljno riješenih primjera. Gdje god je to bilo moguće, primjeri su riješeni i analitički (uobičajenim metodama i tehnikama dinamičkoga programiranja) i pomoću nekoga od spomenutih edukacijskih programa, pri čemu su posebno naglašeni slučajevi gdje se odabrani računalni programi ne mogu primijeniti, te odgovarajući razlozi za to. Na kraju svake pojedine točke navedeno je nekoliko zadataka za vježbu namijenjenih samostalnom radu studenata, a koje bi trebao znati ispravno riješiti svaki student koji je usvojio izloženo gradivo. U posljednjoj je točki najprije naveden primjer pismenoga ispita predviđen za rješavanje na redovnoj nastavi, a potom i zadaci postavljeni na ispitnim rokovima u akademskoj godini 2007/2008.

Na ovom mjestu želim izraziti svoju zahvalnost svima koji su mi izravno ili neizravno pomogli u pisanju skripta. Tu ponajprije mislim na gđu Carmen Tariba, voditeljicu stručnih studija u zagrebačkom nastavnom centru BA Nordhessen na čiji je poticaj skripta i nastala, prof.dr.sc. Jurja Šiftara, redovnog profesora na PMF-MO Sveučilišta u Zagrebu čiji su mi savjeti proizašli iz dugogodišnjega nastavnoga iskustva bili od neprocjenjive koristi, te na sve studente koji su svojim pitanjima i prijedlozima pomogli poboljšati i nadopuniti osnovni tekst.

Iako sam nastojao minimizirati ukupan broj pogrešaka u skripti, svjestan sam da je izvjestan broj njih "preživio" sve dosadašnje korekture, pa unaprijed zahvaljujem svima koji pronađu bilo koju od njih i obavijeste me o njoj.

U Zagrebu, 19. srpnja 2008.

Bojan Kovačić

1. *Uvod u dinamičko programiranje*

U većini područja i problema kojima se bave operacijska istraživanja potrebno je odrediti vrijednosti nekih *varijabli* tako da se postigne optimalna vrijednost određene *funkcije cilja*. U tom smislu spomenute varijable možemo nazvati i *varijable odluke*. Uobičajene metode i algoritmi operacijskih istraživanja istodobno tretiraju sve varijable, pa npr. nastoje postići optimalnu vrijednost funkcije cilja *istodobnim* smanjivanjem vrijednosti barem jedne varijable i povećavanjem vrijednosti barem jedne od preostalih varijabli. Kod strukturno vrlo složenih problema takav je pristup, gledajući sa stanovišta složenosti i efikasnosti algoritama, vrlo spor i neefikasan. Stoga je nerijetko primjerenije i efikasnije strukturno vrlo složen problem rastaviti na komponente, tj. niz manjih i za rješavanje jednostavnijih problema, pa nakon dobivanja optimalnoga rješenja svake pojedine komponente sustavno povezati ta rješenja kako bi se dobilo optimalno rješenje početnoga problema. Ovakav način predstavlja idejnu osnovu za tzv. *dinamičko programiranje*.

Dinamičko programiranje je posebna vrsta matematičkoga programiranja koje se vrlo uspješno primjenjuje u gotovo svim područjima operacijskih istraživanja. Njegovi praktični nedostaci su složena matematička formulacija, kao i nemogućnost preciznoga definiranja klasa problema koji se mogu riješiti metodama dinamičkoga programiranja. Stoga se može zaključiti da primjena dinamičkoga programiranja u operacijskim istraživanjima praktično ovisi o ljudskom čimbeniku, odnosno istraživaču. Istraživač je taj koji treba prepoznati problem koji je moguće riješiti dinamičkim programiranjem, te definirati metode kojima će to i ostvariti. Ipak, osnovna prednost dinamičkoga programiranja je vrlo pogodno korištenje različitih programskih jezika za dobivanje numeričkih rješenja.

Osnovno načelo dinamičkoga programiranja je tzv. *Bellmanovo načelo*[1] prema kojemu se optimizacijski problem rješava po *fazama*, pri čemu se *u proračunu svake faze koriste optimalne vrijednosti dobivene u prethodnoj fazi*. Zbog toga se u matematičkoj formulaciji odgovarajuće metode koriste netrivijalne *rekurzivne relacije*[2] (ili sustavi takvih relacija) i *funkcijske jednadžbe*, što opravdava tvrdnju o složenoj matematičkoj formulaciji samih metoda. U ovom se kolegiju nećemo detaljno baviti metodama dinamičkoga programiranja, nego ćemo upoznati sljedeće primjere njegove primjene:

- određivanje najkraćega puta između dvaju vrhova *mreže* (usmjereni težinski graf);
- određivanje optimalnoga rješenja problema ruksaka;
- problem razdiobe ulaganja;
- problem trgovačkoga putnika;
- problem kineskoga poštara.

[1] Richard Ernest Bellman (1920. – 1984.), američki matematičar.

[2] *Rekurzivna relacija* za niz $(a_n)_{n\in\mathbf{N}}$ je bilo koja funkcija koja povezuje n – ti član niza s barem jednim od prethodnih članova niza. Npr. za niz prirodnih brojeva ta relacija glasi: $a_n = a_{n-1} + 1$, pri čemu se dodatno definira $a_1 = 1$.

1.1. Problem najkraćega puta u grafu

Na početku podsjetimo ukratko na osnovne pojmove teorije grafova. (*Konačan*) graf G je uređeni par (V, E) koji se sastoji od konačnoga skupa V čije elemente nazivamo vrhovi ili čvorovi grafa G, te konačnoga skupa E čije elemente nazivamo bridovi grafa G. Radi jednostavnosti, obično ćemo uzimati da je $V = [n]$ za neki $n \in \mathbf{N}$.[3] Na skupu E zadaje se *funkcija incidencije f* koja svakom elementu toga skupa pridružuje ili neuređeni ili uređeni par ne nužno različitih elemenata skupa V. Ako funkcija f svakom elementu skupa E pridružuje neuređeni par elemenata skupa V, graf G nazivamo neusmjereni graf ili, kraće, samo graf. Ako funkcija f svakom elementu skupa E pridružuje uređeni par elemenata skupa V, graf G nazivamo uređeni graf ili digraf, a u tom slučaju elemente skupa E nazivamo lukovi grafa G. Grafovi koji sadrže i neusmjerene i usmjerene bridove (lukove) nazivaju se miješani grafovi.

Nadalje, ako je $e \in E$ i $u, v \in V$, onda zapis $e \xrightarrow{f} \{u, v\}$ čitamo kao: "*Brid e spaja vrhove u i v.*", odnosno "*Vrhovi u i v su incidentni s bridom e.*", odnosno "*Vrhovi u i v su krajevi brida e.*" Zapis $e \xrightarrow{f} (u, v)$ čitamo kao: "*Luk e spaja vrhove u i v.*", odnosno "*Vrh u je početak luka e, a vrh v je kraj luka e.*", odnosno "*Vrh u je pozitivno incidentan s lukom e, a vrh v je negativno incidentan s lukom e.*". Kažemo da su vrhovi u i v susjedni ukoliko postoji (ne)usmjereni brid e koji ih spaja. Ukupan broj različitih bridova incidentnih s nekim vrhom nazivamo stupanj vrha.

(Di)graf G je jednostavan ukoliko su svaka dva njegova vrha spojena najviše jednim bridom i niti jedan brid nema početak i kraj u istom vrhu grafa. Ukoliko su, pak, svaka dva različita vrha grafa G spojena točno jednim bridom, kažemo da je graf G potpun.

Na skupu E moguće je i zadati realnu funkciju $w : E \to \mathbf{R}$ koja svakom bridu (di)grafa G pridružuje točno jedan realan broj $w(e)$. Tada se uređena trojka (V, E, w) naziva težinski (di)graf, a realan broj $w(e)$ težina brida e. Za težine bridova obično ćemo uzimati nenegativne realne brojeve, odnosno, još preciznije, elemente skupa $\mathbf{N}_0 = \{0, 1, 2, \ldots\}$, a zadavat ćemo ih uglavnom na jednostavnim (di)grafovima. Budući da ćemo uzimati $V = [n]$, težine bridova jednostavnih (di)grafova zadavat ćemo kvadratnom matricom $W = [w_{ij}]$ reda n takvom da je element na mjestu (i, j) te matrice jednak težini brida koji spaja vrh i i vrh j. Ako takav brid ne postoji, dogovorno definiramo $w_{ij} = \infty$ za $i \neq j$, te $w_{ii} = 0$ za svaki $i \in [n]$.

Konačna neusmjerena šetnja ili, kraće, šetnja u neusmjerenom grafu G konačan niz $\check{S} = v_1\ e_1\ v_2\ e_2\ v_3\ \ldots\ v_k\ e_k\ v_{k+1}$ vrhova i bridova toga grafa takvih da za svaki $j = 1, 2, \ldots, k$ vrijedi $e_k \xrightarrow{f} \{v_k, v_{k+1}\}$. Tada kažemo da je vrh v_1 početak ili polazište, vrhovi $v_2, \ldots, v_k$ unutrašnji vrhovi, vrh v_{k+1} kraj ili odredište, a prirodan broj k duljina šetnje $\check{S}$. Ako su svi vrhovi šetnje međusobno različiti i ako su svi bridovi šetnje međusobno različiti, šetnja se naziva put. U težinskom se grafu težina puta P definira kao zbroj težina svih bridova koji tvore put P. Težina puta ujedno je i udaljenost polazišta i odredišta toga puta. Potpuno analogno se definiraju *konačna usmjerena šetnja*, odnosno *usmjereni put* u bilo kojem digrafu, te *težina usmjerenoga puta* i *udaljenost* (bilo kojih dvaju vrhova) u bilo kojem težinskom digrafu.

[3] $[n] = \{1, 2, \ldots, n\} \subset \mathbf{N}$, skup prvih n prirodnih brojeva.

Jedan od poznatih problema je <u>*problem određivanja najkraćega puta*</u> (odnosno, najkraće udaljenosti) između bilo kojih dvaju vrhova (ne)usmjerenoga težinskoga grafa. Pritom odmah napomenimo da se termin *najkraći* ne odnosi na minimizaciju duljine puta (tj. ukupnoga broja bridova (ili lukova) koji tvore put), nego na minimizaciju težine puta (tj. zbroja težina svih bridova koji tvore put). Taj se problem može riješiti raznim metodama (npr. Dijkstrinim[4] ili Dantzigovim[5] algoritmom), a mi ćemo njegovo *modeliranje*, rješavanje *i interpretaciju* dobivenih rezultata ilustrirati uglavnom rabeći računalni program *Graph Magics*. Odmah istaknimo da najkraći put ne mora nužno biti jedinstven, tj. može postojati više različitih najkraćih putova koji nužno imaju istu težinu, ali ne nužno i istu duljinu.

<u>Primjer 1.</u> U nekoj je računalnoj mreži potrebno poslati elektroničku poštu od računala 1 do računala 11. Izravna veza tih dvaju računala ne postoji, ali komunikacija je moguća posredstvom ostalih računala u mreži. Prosječni intenzitet prometa informacija na svakoj pojedinoj liniji u mreži zadan je sljedećom težinskom matricom W:

$$W = \begin{bmatrix} 0 & 2 & 3 & 4 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & \infty & 7 & 4 & 5 & \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty & 4 & 2 & 1 & \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & 0 & 2 & 2 & 8 & \infty & \infty & \infty & \infty \\ \infty & 7 & 4 & 2 & 0 & \infty & \infty & 5 & 3 & 5 & \infty \\ \infty & 4 & 2 & 2 & \infty & 0 & \infty & 1 & 5 & 4 & \infty \\ \infty & 5 & 1 & 8 & \infty & \infty & 0 & 1 & 3 & 2 & \infty \\ \infty & \infty & \infty & \infty & 5 & 1 & 1 & 0 & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & 3 & 5 & 3 & \infty & 0 & \infty & 3 \\ \infty & \infty & \infty & \infty & 5 & 4 & 2 & \infty & \infty & 0 & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 2 & 3 & 3 & 0 \end{bmatrix}$$

Odredimo računala preko kojih treba obaviti prijenos podataka tako da e-pošta u najkraćem vremenu stigne od polazišta do odredišta (računala 11.

"<u>*Klasično*</u>" <u>*rješenje*</u>: Opišimo ukratko osnovnu ideju rješavanja. Računala modeliramo kao vrhove neusmjerenoga jednostavnoga težinskoga grafa G, a izravnu dvosmjernu vezu među njima kao bridove toga grafa. Budući da imamo ukupno 11 računala, za skup vrhova V uzet ćemo skup [11], tj. $V = [11]$. Skup bridova E ne znamo, ali ga lagano možemo "očitati" koristeći zadanu težinsku matricu W. Istaknimo odmah: element w_{ij} matrice W interpretiramo kao prosječni intenzitet prometa na izravnoj vezi između računala i i računala j. Ako je $w_{ij} = \infty$, računala i i j nisu povezana izravnom vezom, a $w_{ii} = 0$ stavljamo dogovorno.

Za svaki $i \in [11]$ označimo s $t^i_{\min}$ težinu najkraćega puta od vrha i do vrha 11. To možemo učiniti jer, sukladno gornjim napomenama, svi najkraći putovi imaju istu težinu Radi jednostavnosti, činjenicu da su vrhovi i i j međusobno susjedni pisat ćemo kao $i \leftrightarrow j$. Tada za svaki $i \in [10]$ vrijedi sljedeća jednakost:

[4] Edsger Wybe Dijkstra (1930. – 2002.), nizozemski informatičar.
[5] George Bernard Dantzig (1914. – 2005.), američki matematičar, "otac linearnoga programiranja".

$$t_{\min}^{i} = \min_{j \leftrightarrow i}(w_{ij} + t_{\min}^{j}) .$$

Ona se dobije kao izravna posljedica odluke da se iz vrha *i* u vrh 11 dođe preko nekog vrha *j* susjednoga s vrhom *i*. Ostatak puta – od vrha *j* do vrha 11 – treba odabrati tako da težina puta od vrha *j* do vrha 11 bude najmanja moguća. Sukladno definiciji veličina $t_{\min}^{i}$, ta je težina jednaka $t_{\min}^{j}$. Stoga je ukupna težina puta od vrha *i* do vrha 11 jednaka w_{ij} + $t_{\min}^{j}$. Budući da želimo minimizirati ukupnu težinu, odabiremo najmanju od svih dobivenih vrijednosti, a otuda izravno proizlazi navedena formula.

Za *i* = 11 dogovorno stavljamo $t_{\min}^{11} = 0$, čime dobivamo sljedeći sustav rekurzivnih relacija:

$$t_{\min}^{i} = \min_{j \leftrightarrow i}(w_{ij} + t_{\min}^{j})$$
$$t_{\min}^{11} = 0$$

Taj sustav nije moguće riješiti uobičajenim tehnikama rješavanja sustava rekurzivnih relacija, ali se može pokazati da on ima barem jedno rješenje. R. Bellman je navedeni sustav riješio tzv. *metodom sukcesivnih aproksimacija*. Osnovna ideja te metode je u *k* – tom koraku promatrati težine svih puteva od vrha *i* do odredišta (u našem slučaju, vrha 11) čija je duljina najviše *k*, tj. koji se sastoje od najviše *k* bridova, te naći najmanju od svih promatranih težina. Može se pokazati da je niz tako dobivenih težina stacionaran, tj. počevši od nekoga mjesta niz postaje konstantan. Ugrubo, jedan od algoritama za određivanje najkraćega puta od vrha *i* do odredišta *s minimalnom duljinom* (minimalnim brojem bridova) može se formulirati ovako:

Korak 1. Staviti *k* = 1 i $w_{\min}^{0} = 0$.
Korak 2. Promatrati sve putove od vrha *i* do odredišta koji se sastoje od najviše *k* bridova. Izračunati težinu svakoga pojedinoga puta.
Korak 3. Između svih težina izračunatih u Koraku 2. odabrati najmanju ($w_{\min}^{k}$).
Korak 4. Ako je $w_{\min}^{k} = w_{\min}^{k-1}$, stop. Minimalna duljina najkraćega puta jednaka je *k* – 1, a pripadna težina je $w_{\min}^{k-1}$. Inače, povećati *k* na *k* + 1 i vratiti se na Korak 2.

Sukladno opisanom algoritmu, *u prvoj fazi* za svaki $i \in [11]$ stavimo:

$$w_i^1 = w_{i,11}.$$

Veličine w_i^1 zapravo određuju prosječan intenzitet prometa na izravnoj vezi između računala *i* i računala 11 ukoliko takva veza uopće postoji. Zbog toga je:

$$w_1^1 = w_2^1 = w_3^1 = w_4^1 = w_5^1 = w_6^1 = w_7^1 = \infty,$$
$$w_8^1 = w_{8,11} = 2, w_9^1 = w_{9,11} = 3, w_{10}^1 = w_{10,11} = 3, w_{11}^1 = 0$$

jer između vrhova 1 i 11, 2 i 11, 3 i 11, …, 7 i 11 ne postoji izravni brid, između vrhova 8 i 11 postoji izravni brid težine 2, a između vrhova 9 i 11, odnosno 10 i 11 postoje izravni bridovi težine 3.

U *drugoj fazi* za svaki $i \in [11]$ računamo vrijednosti w_i^2 definirane relacijama:

$$w_i^2 = \min_{j \leftrightarrow i}(w_{ij} + w_j^1)\,.$$

w_i^2 je, zapravo, najmanja od svih težina putova duljine najviše 2 između vrha i i vrha 11 (ukoliko takvi putovi uopće postoje). Gornja relacija je izravna posljedica Bellmanova načela: iz vrha i najprije idemo u neki vrh j susjedan vrhu i, pa iz vrha j *optimalnim putem dobivenim u fazi* 1 dolazimo u vrh 11. Ukupna težina takvoga puta je $w_{ij} + w_j^1$, pa među svim takvim putovima odabiremo onaj najmanje težine. Tako imamo:

$$w_1^2 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^1) = \min_{j \in \{2,3,4\}}(w_{1j} + w_j^1) = \min\left\{w_{12} + w_2^1, w_{13} + w_3^1, w_{14} + w_4^1\right\} =$$
$$= \min\left\{2+\infty, 3+\infty, 4+\infty\right\} = \infty,$$
$$w_2^2 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^1) = \min_{j \in \{1,5,6,7\}}(w_{2j} + w_j^1) = \min\left\{w_{21} + w_1^1, w_{25} + w_5^1, w_{26} + w_6^1, w_{27} + w_7^1\right\} =$$
$$= \min\left\{2+\infty, 7+\infty, 4+\infty, 5+\infty\right\} = \infty,$$
$$w_3^2 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^1) = \min_{j \in \{1,5,6,7\}}(w_{3j} + w_j^1) = \min\left\{w_{31} + w_1^1, w_{35} + w_5^1, w_{36} + w_6^1, w_{37} + w_7^1\right\} =$$
$$= \min\left\{3+\infty, 4+\infty, 2+\infty, 1+\infty\right\} = \infty,$$
$$w_4^2 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^1) = \min_{j \in \{1,5,6,7\}}(w_{4j} + w_j^1) = \min\left\{w_{41} + w_1^1, w_{45} + w_5^1, w_{46} + w_6^1, w_{47} + w_7^1\right\} =$$
$$= \min\left\{4+\infty, 2+\infty, 2+\infty, 8+\infty\right\} = \infty,$$
$$w_5^2 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^1) = \min_{j \in \{2,3,4,8,9,10\}}(w_{5j} + w_j^1) = \left\{\min w_{52} + w_2^1, w_{53} + w_3^1, w_{54} + w_4^1, w_{58} + w_8^1, w_{59} + w_9^1,\right.$$
$$\left. w_{5,10} + w_{10}^1\right\} = \min\left\{7+\infty, 4+\infty, 2+\infty, 5+2, 3+3, 5+3\right\} = 6,$$
$$w_6^2 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^1) = \min_{j \in \{2,3,4,8,9,10\}}(w_{6j} + w_j^1) = \min\left\{w_{62} + w_2^1, w_{63} + w_3^1, w_{64} + w_4^1, w_{68} + w_8^1, w_{69} + w_9^1,\right.$$
$$\left. w_{6,10} + w_{10}^1\right\} = \min\left\{4+\infty, 2+\infty, 2+\infty, 1+2, 5+3, 4+3\right\} = 3,$$
$$w_7^2 = \min_{j \leftrightarrow 5}(w_{7j} + w_j^1) = \min_{j \in \{2,3,4,8,9,10\}}(w_{7j} + w_j^1) = \min\left\{w_{72} + w_2^1, w_{73} + w_3^1, w_{74} + w_4^1, w_{78} + w_8^1, w_{79} + w_9^1,\right.$$
$$\left. w_{7,10} + w_{10}^1\right\} = \min\left\{5+\infty, 1+\infty, 8+\infty, 1+2, 3+3, 2+3\right\} = 3,$$
$$w_8^2 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^1) = \min_{j \in \{5,6,7,11\}}(w_{8j} + w_j^1) = \min\left\{w_{85} + w_5^1, w_{86} + w_6^1, w_{87} + w_7^1, w_{8,11} + w_{11}^1\right\} =$$
$$= \min\left\{5+\infty, 1+\infty, 1+\infty, 2+0\right\} = 2,$$
$$w_9^2 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^1) = \min_{j \in \{5,6,7,11\}}(w_{9j} + w_j^1) = \min\left\{w_{95} + w_5^1, w_{96} + w_6^1, w_{97} + w_7^1, w_{9,11} + w_{11}^1\right\} =$$
$$= \min\left\{3+\infty, 5+\infty, 3+\infty, 3+0\right\} = 3,$$
$$w_{10}^2 = \min_{j \leftrightarrow 10}(w_{10,j} + w_j^1) = \min_{j \in \{5,6,7,11\}}(w_{10,j} + w_j^1) = \min\left\{w_{10,5} + w_5^1, w_{10,6} + w_6^1, w_{10,7} + w_7^1, w_{10,11} + w_{11}^1\right\} =$$
$$= \min\left\{5+\infty, 4+\infty, 2+\infty, 3+0\right\} = 3,$$
$$w_{11}^2 = 0.$$

U *trećoj fazi* za svaki $i \in [11]$ računamo vrijednosti w_i^3 definirane relacijama:

$$w_i^3 = \min_{j \leftrightarrow i}(w_{ij} + w_j^2).$$

w_i^3 je zapravo najmanja od svih težina putova duljine najviše 3 između vrha *i* i vrha 11 (ukoliko takvi putovi uopće postoje). I u ovoj fazi dosljedno primjenjujemo Bellmanovo načelo: iz vrha *i* najprije idemo u neki vrh *j* susjedan vrhu *i*, pa iz vrha *j* *optimalnim putem dobivenim u fazi* 2 dolazimo u vrh 11. Tako imamo:

$$w_1^3 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^2) = \min_{j \in \{2,3,4\}}(w_{1j} + w_j^2) = \min\left\{w_{12} + w_2^2, w_{13} + w_3^2, w_{14} + w_4^2\right\} =$$
$$= \min\{2+\infty, 3+\infty, 4+\infty\} = \infty,$$
$$w_2^3 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^2) = \min_{j \in \{1,5,6,7\}}(w_{2j} + w_j^2) = \min\left\{w_{21} + w_1^2, w_{25} + w_5^2, w_{26} + w_6^2, w_{27} + w_7^2\right\} =$$
$$= \min\{2+\infty, 7+6, 4+3, 5+3\} = 7,$$
$$w_3^3 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^2) = \min_{j \in \{1,5,6,7\}}(w_{3j} + w_j^2) = \min\left\{w_{31} + w_1^2, w_{35} + w_5^2, w_{36} + w_6^2, w_{37} + w_7^2\right\} =$$
$$= \min\{3+\infty, 4+6, 2+3, 1+3\} = 4,$$
$$w_4^3 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^2) = \min_{j \in \{1,5,6,7\}}(w_{4j} + w_j^2) = \min\left\{w_{41} + w_1^2, w_{45} + w_5^2, w_{46} + w_6^2, w_{47} + w_7^2\right\} =$$
$$= \min\{4+\infty, 2+6, 2+3, 8+3\} = 5,$$
$$w_5^3 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^2) = \min_{j \in \{2,3,4,8,9,10\}}(w_{5j} + w_j^2) = \min\left\{w_{52} + w_2^2, w_{53} + w_3^2, w_{54} + w_4^2, w_{58} + w_8^2, w_{59} + w_9^2,\right.$$
$$\left. w_{5,10} + w_{10}^2\right\} = \min\{7+\infty, 4+\infty, 2+\infty, 5+2, 3+3, 5+3\} = 6,$$
$$w_6^3 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^2) = \min_{j \in \{2,3,4,8,9,10\}}(w_{6j} + w_j^2) = \min\left\{w_{62} + w_2^2, w_{63} + w_3^2, w_{64} + w_4^2, w_{68} + w_8^2, w_{69} + w_9^2,\right.$$
$$\left. w_{6,10} + w_{10}^2\right\} = \min\{4+\infty, 2+\infty, 2+\infty, 1+2, 5+3, 4+3\} = 3,$$
$$w_7^3 = \min_{j \leftrightarrow 7}(w_{7j} + w_j^2) = \min_{j \in \{2,3,4,8,9,10\}}(w_{7j} + w_j^2) = \min\left\{w_{72} + w_2^2, w_{73} + w_3^2, w_{74} + w_4^2, w_{78} + w_8^2, w_{79} + w_9^2,\right.$$
$$\left. w_{7,10} + w_{10}^2\right\} = \min\{5+\infty, 1+\infty, 8+\infty, 1+2, 3+3, 2+3\} = 3,$$
$$w_8^3 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^2) = \min_{j \in \{5,6,7,11\}}(w_{8j} + w_j^2) = \min\left\{w_{85} + w_5^2, w_{86} + w_6^2, w_{87} + w_7^2, w_{8,11} + w_{11}^2\right\} =$$
$$= \min\{5+6, 1+3, 1+3, 2+0\} = 2,$$
$$w_9^3 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^2) = \min_{j \in \{5,6,7,11\}}(w_{9j} + w_j^2) = \min\left\{w_{95} + w_5^2, w_{96} + w_6^2, w_{97} + w_7^2, w_{9,11} + w_{11}^2\right\} =$$
$$= \min\{3+6, 5+3, 3+3, 3+0\} = 3,$$
$$w_{10}^3 = \min_{j \leftrightarrow 10}(w_{10,j} + w_j^2) = \min_{j \in \{5,6,7,11\}}(w_{10,j} + w_j^2) = \min\left\{w_{10,5} + w_5^2, w_{10,6} + w_6^2, w_{10,7} + w_7^2, w_{10,11} + w_{11}^2\right\} =$$
$$= \min\{5+6, 4+3, 2+3, 3+0\} = 3,$$
$$w_{11}^3 = 0.$$

U *četvrtoj fazi* za svaki $i \in [11]$ računamo vrijednosti w_i^4 definirane relacijama:

$$w_i^4 = \min_{j \leftrightarrow i}(w_{ij} + w_j^3).$$

w_i^4 je, zapravo, najmanja od svih težina putova duljine najviše 4 između vrha i i vrha 11 (ukoliko takvi putovi uopće postoje). Tako imamo:

$$w_1^4 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^3) = \min_{j \in \{2,3,4\}}(w_{1j} + w_j^3) = \min\left\{w_{12} + w_2^3, w_{13} + w_3^3, w_{14} + w_4^3\right\} =$$
$$= \min\{2+7, 3+4, 4+5\} = 7,$$
$$w_2^4 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^3) = \min_{j \in \{1,5,6,7\}}(w_{2j} + w_j^3) = \min\left\{w_{21} + w_1^3, w_{25} + w_5^3, w_{26} + w_6^3, w_{27} + w_7^3\right\} =$$
$$= \min\{2+\infty, 7+6, 4+3, 5+3\} = 7,$$
$$w_3^4 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^3) = \min_{j \in \{1,5,6,7\}}(w_{3j} + w_j^3) = \min\left\{w_{31} + w_1^3, w_{35} + w_5^3, w_{36} + w_6^3, w_{37} + w_7^3\right\} =$$
$$= \min\{3+\infty, 4+6, 2+3, 1+3\} = 4,$$
$$w_4^4 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^3) = \min_{j \in \{1,5,6,7\}}(w_{4j} + w_j^3) = \min\left\{w_{41} + w_1^3, w_{45} + w_5^3, w_{46} + w_6^3, w_{47} + w_7^3\right\} =$$
$$= \min\{4+\infty, 2+6, 2+3, 8+3\} = 5,$$
$$w_5^4 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^3) = \min_{j \in \{2,3,4,8,9,10\}}(w_{5j} + w_j^3) = \min\left\{w_{52} + w_2^3, w_{53} + w_3^3, w_{54} + w_4^3, w_{58} + w_8^3, w_{59} + w_9^3,\right.$$
$$\left. w_{5,10} + w_{10}^3\right\} = \min\{7+7, 4+4, 2+5, 5+2, 3+3, 5+3\} = 6,$$
$$w_6^4 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^3) = \min_{j \in \{2,3,4,8,9,10\}}(w_{6j} + w_j^3) = \min\left\{w_{62} + w_2^3, w_{63} + w_3^3, w_{64} + w_4^3, w_{68} + w_8^3, w_{69} + w_9^3,\right.$$
$$\left. w_{6,10} + w_{10}^3\right\} = \min\{4+7, 2+4, 2+5, 1+2, 5+3, 4+3\} = 3,$$
$$w_7^4 = \min_{j \leftrightarrow 7}(w_{7j} + w_j^3) = \min_{j \in \{2,3,4,8,9,10\}}(w_{7j} + w_j^3) = \min\left\{w_{72} + w_2^3, w_{73} + w_3^3, w_{74} + w_4^3, w_{78} + w_8^3, w_{79} + w_9^3,\right.$$
$$\left. w_{7,10} + w_{10}^3\right\} = \min\{5+7, 1+4, 8+5, 1+2, 3+3, 2+3\} = 3,$$
$$w_8^4 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^3) = \min_{j \in \{5,6,7,11\}}(w_{8j} + w_j^3) = \min\left\{w_{85} + w_5^3, w_{86} + w_6^3, w_{87} + w_7^3, w_{8,11} + w_{11}^3\right\} =$$
$$= \min\{5+6, 1+3, 1+3, 2+0\} = 2,$$
$$w_9^4 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^3) = \min_{j \in \{5,6,7,11\}}(w_{9j} + w_j^3) = \min\left\{w_{95} + w_5^3, w_{96} + w_6^3, w_{97} + w_7^3, w_{9,11} + w_{11}^3\right\} =$$
$$= \min\{3+6, 5+3, 3+3, 3+0\} = 3,$$
$$w_{10}^4 = \min_{j \leftrightarrow 10}(w_{10,j} + w_j^3) = \min_{j \in \{5,6,7,11\}}(w_{10,j} + w_j^3) = \min\left\{w_{10,5} + w_5^3, w_{10,6} + w_6^3, w_{10,7} + w_7^3, w_{10,11} + w_{11}^3\right\} =$$
$$= \min\{5+6, 4+3, 2+3, 3+0\} = 3,$$
$$w_{11}^4 = 0.$$

U *petoj fazi* za svaki $i \in [11]$ treba računati vrijednosti w_i^5 definirane relacijama:

$$w_i^5 = \min_{j \leftrightarrow i}(w_{ij} + w_j^4).$$

w_i^5 je, zapravo, najmanja od svih težina putova duljine najviše 5 između vrha i i vrha 11 (ukoliko takvi putovi uopće postoje). Za vrh 1, odnosno $i = 1$ dobivamo:

$$w_1^5 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^4) = \min_{j \in \{2,3,4\}}(w_{1j} + w_j^4) = \min\left\{w_{12} + w_2^4, w_{13} + w_3^4, w_{14} + w_4^4\right\} =$$
$$= \min\left\{2+7, 3+4, 4+5\right\} = 7$$

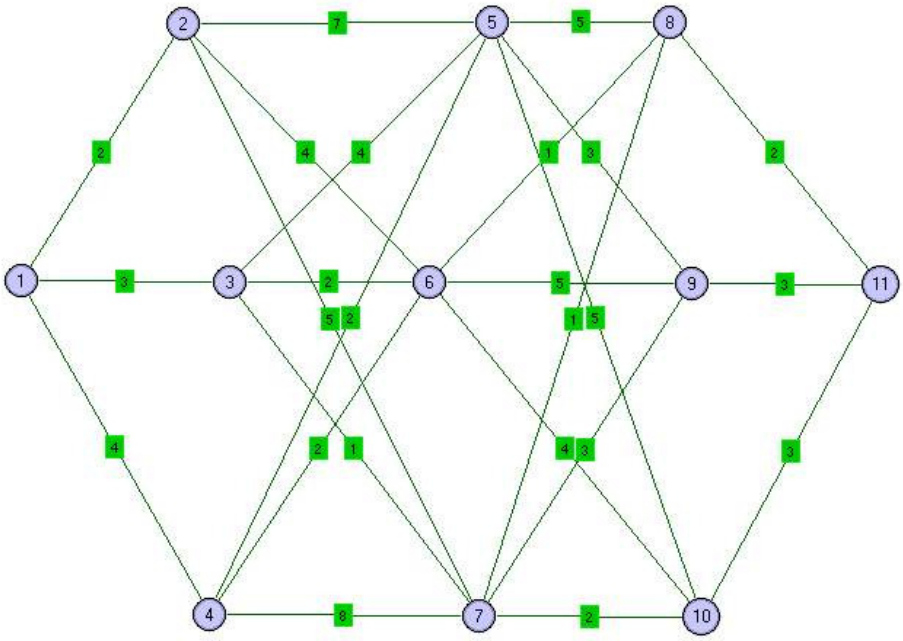
pa očito vrijedi jednakost

$$w_1^5 = w_1^4 .$$

Stoga je *minimalna duljina najkraćega puta* od vrha 1 do vrha 11 jednaka 4, a pripadna optimalna težina $w_1^4 = 7$.

Odredimo kojim vrhovima prolazi dobiveni najkraći put. Za zadavanje puta dovoljno je odrediti samo vrhove puta jer su time svi bridovi puta jednoznačno određeni. Najmanja vrijednost $w_1^4 = 7$ dobivena je kao zbroj $w_{13} + w_3^3$, pa je prvi unutrašnji vrh traženoga puta vrh 3. Vrijednost $w_3^3 = 4$ dobivena je kao zbroj $w_{37} + w_7^2$, pa je drugi unutrašnji vrh traženoga puta vrh 7. Vrijednost $w_7^2 = 3$ dobivena je kao zbroj $w_{78} + w_8^1$, pa je preostali unutrašnji vrh traženoga puta vrh 8. Dakle, traženi najkraći put je: 1 – 3 – 7 – 8 – 11, tj. prijenos e-poruke treba obaviti koristeći računala 3, 7 i 8. Najmanji prosječan intenzitet prometa informacija na tom putu jednak je težini puta, odnosno 7.

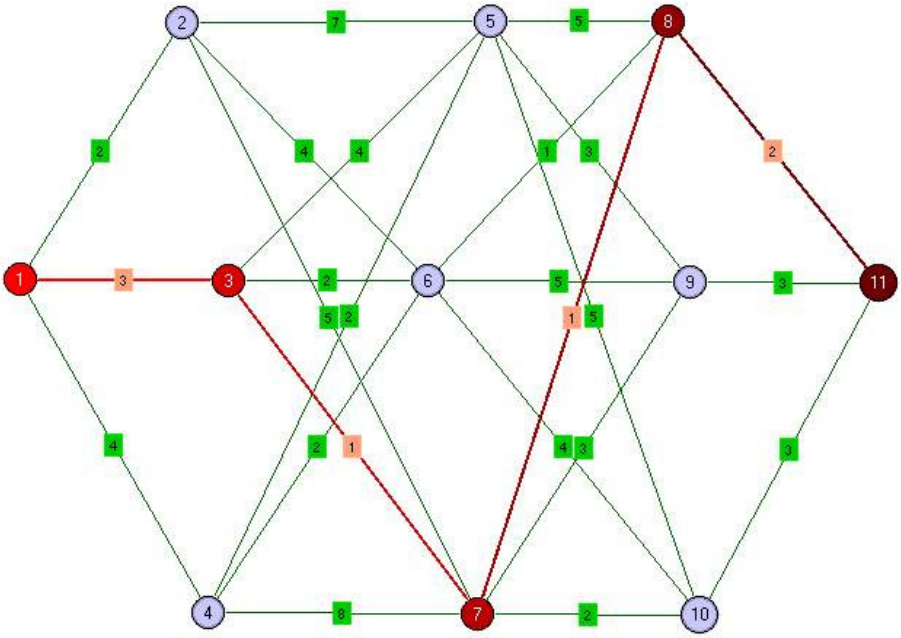
<u>*Rješenje pomoću programa Graph Magics*</u>: Sukladno polaznim razmatranjima iz "klasičnoga" rješenja ovoga primjera, razmatrani problem modeliramo sljedećim neusmjerenim težinskim grafom *G*:



Slika 1. Model problema iz Primjera 1.

Desnom tipkom miša kliknimo na vrh 1, pa označimo taj vrh kao polazište odabirom podopcije *Set/Unset as Start Vertex* (*Source*). Potom desnom tipkom miša kliknimo na vrh 11, pa označimo taj vrh kao odredište odabirom podopcije *Set/Unset as End Vertex* (*Sink*). Sada

kliknimo desnom tipkom miša na bilo koju bjelinu unutar radne površine, pa na dobivenom izborniku odaberimo opciju *Find*, a na pripadnom podizborniku opciju *Shortest Path (from Start to End Vertex)*. Tako dobivamo:



Slika 2. Rješenje Primjera 1. dobiveno programom *Graph Magics*.

Crveno označen put u grafu je traženi najkraći put: 1 – 3 – 7 – 8 – 11, a njegova je težina $w_{min} = 3 + 1 + 1 + 2 = 7$, upravo kako smo dobili i "klasičnim" načinom rješavanja.

<u>Primjer 2.</u> 10 mjesta međusobno je povezano dvosmjernim putevima. Duljine izravnih dvosmjernih putova [u km] između pojedinih mjesta dane su u sljedećoj težinskoj matrici W:

$$W = \begin{bmatrix} 0 & 1 & 10 & 6 & 3 & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & 10 & \infty & \infty & 10 & \infty & \infty & \infty & \infty \\ 10 & 10 & 0 & 4 & \infty & 2 & 4 & 1 & \infty & \infty \\ 6 & \infty & 4 & 0 & 2 & \infty & \infty & 3 & \infty & \infty \\ 3 & \infty & \infty & 2 & 0 & \infty & \infty & 6 & 8 & \infty \\ \infty & 10 & 2 & \infty & \infty & 0 & 2 & \infty & \infty & 5 \\ \infty & \infty & 4 & \infty & \infty & 2 & 0 & 6 & \infty & 2 \\ \infty & \infty & 1 & 3 & 6 & \infty & 6 & 0 & 10 & 9 \\ \infty & \infty & \infty & \infty & 8 & \infty & \infty & 10 & 0 & 5 \\ \infty & \infty & \infty & \infty & \infty & 5 & 2 & 9 & 5 & 0 \end{bmatrix}.$$

Odredimo najkraći put iz mjesta 1 u mjesto 10.

"*<u>Klasično</u>*" *<u>rješenje</u>*: Analogno kao u rješenju Primjera 1., problem modeliramo neusmjerenim jednostavnim težinskim grafom G čiji su vrhovi promatrana mjesta, a bridovi izravni dvosmjerni putovi između odgovarajućih mjesta. Težina svakoga brida jednaka je duljini

odgovarajućega izravnoga puta. Budući da imamo ukupno 10 mjesta, za skup vrhova V grafa G uzet ćemo [10], tj. $V = [10]$. Skup bridova E grafa G ne znamo, ali ga možemo jednostavno "očitati" pomoću zadane težinske matrice W. Ponovimo: element w_{ij} matrice W interpretiramo kao duljinu [u km] izravnoga dvosmjernoga puta između mjesta i i mjesta j. Ako je $w_{ij} = \infty$, mjesta i i j nisu povezana izravnim dvosmjernim putem, a $w_{ii} = 0$ stavljamo dogovorno. Analogno kao u rješenju Primjera 1., problem rješavamo dinamičkim programiranjem.

U *prvoj fazi* za svaki $i \in [10]$ stavimo:

$$w_i^1 = w_{i,10}\,.$$

w_i^1 je, zapravo, duljina [u km] izravnoga dvosmjernoga puta (puta duljine 1) između mjesta i i mjesta 10 ukoliko takav izravni put uopće postoji. Zbog toga je:

$$w_1^1 = w_2^1 = w_3^1 = w_4^1 = w_5^1 = \infty, w_6^1 = w_{6,10} = 5,$$
$$w_7^1 = w_{7,10} = 2, w_8^1 = w_{8,10} = 9, w_9^1 = w_{9,10} = 5, w_{10}^1 = 0$$

jer između vrhova 1 i 10, 2 i 10, 3 i 10, 4 i 10, te 5 i 10 ne postoji izravni brid, između vrhova 6 i 10, odnosno 9 i 10 postoje izravni bridovi težine 5, između vrhova 7 i 10 postoji izravan brid težine 2, a između vrhova 8 i 10 postoji izravan brid težine 9.

U *drugoj fazi* za svaki $i \in [10]$ računamo vrijednosti w_i^2 definirane relacijama:

$$w_i^2 = \min_{j \leftrightarrow i}(w_{ij} + w_j^1)\,.$$

Izvod navedene formule je isti kao i u rješenju Primjera 1. Tako redom dobivamo:

$$w_1^2 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^1) = \min_{j \in \{2,3,4,5\}}(w_{1j} + w_j^1) = \min\left\{w_{12} + w_2^1, w_{13} + w_3^1, w_{14} + w_4^1, w_{15} + w_5^1\right\} =$$
$$= \min\left\{1 + \infty, 10 + \infty, 6 + \infty, 3 + \infty\right\} = \infty,$$
$$w_2^2 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^1) = \min_{j \in \{1,3,6\}}(w_{2j} + w_j^1) = \min\left\{w_{21} + w_1^1, w_{23} + w_3^1, w_{26} + w_6^1\right\} =$$
$$\min\left\{1 + \infty, 10 + \infty, 10 + 5\right\} = 15,$$
$$w_3^2 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^1) = \min_{j \in \{1,2,4,6,7,8\}}(w_{3j} + w_j^1) = \min\left\{w_{31} + w_1^1, w_{32} + w_2^1, w_{34} + w_4^1, w_{36} + w_6^1,\right.$$
$$\left. w_{37} + w_7^1, w_{38} + w_8^1\right\} = \min\left\{10 + \infty, 10 + \infty, 4 + \infty, 2 + 5, 4 + 2, 1 + 9\right\} = 6,$$
$$w_4^2 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^1) = \min_{j \in \{1,3,5,8\}}(w_{4j} + w_j^1) = \min\left\{w_{41} + w_1^1, w_{43} + w_3^1, w_{45} + w_5^1, w_{48} + w_8^1\right\} =$$
$$= \min\left\{6 + \infty, 4 + \infty, 2 + \infty, 3 + 9\right\} = 12,$$
$$w_5^2 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^1) = \min_{j \in \{1,4,8,9\}}(w_{5j} + w_j^1) = \min\left\{w_{51} + w_1^1, w_{54} + w_4^1, w_{58} + w_8^1, w_{59} + w_9^1\right\} =$$
$$= \min\left\{3 + \infty, 2 + \infty, 6 + 9, 8 + 5\right\} = 13,$$

$$w_6^2 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^1) = \min_{j \in \{2,3,7,10\}}(w_{6j} + w_j^1) = \min\left\{w_{62} + w_2^1, w_{63} + w_3^1, w_{67} + w_7^1, w_{6,10} + w_{10}^1\right\} =$$
$$= \min\left\{10 + \infty, 2 + \infty, 2 + 2, 5 + 0\right\} = 4,$$
$$w_7^2 = \min_{j \leftrightarrow 7}(w_{7j} + w_j^1) = \min_{j \in \{3,6,8,10\}}(w_{7j} + w_j^1) = \min\left\{w_{73} + w_3^1, w_{76} + w_6^1, w_{78} + w_8^1, w_{7,10} + w_{10}^1\right\} =$$
$$= \min\left\{4 + \infty, 2 + 5, 6 + 9, 2 + 0\right\} = 2,$$
$$w_8^2 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^1) = \min_{j \in \{3,4,5,7,9,10\}}(w_{8j} + w_j^1) = \min\left\{w_{83} + w_3^1, w_{84} + w_4^1, w_{85} + w_5^1, w_{87} + w_7^1,\right.$$
$$\left. w_{89} + w_9^1, w_{8,10} + w_{10}^1\right\} = \min\left\{1 + \infty, 3 + \infty, 6 + \infty, 6 + 2, 10 + 5, 9 + 0\right\} = 8,$$
$$w_9^2 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^1) = \min_{j \in \{5,8,10\}}(w_{5j} + w_j^1) = \min\left\{w_{95} + w_5^1, w_{98} + w_8^1, w_{9,10} + w_{10}^1\right\} =$$
$$= \min\left\{8 + \infty, 10 + 9, 5 + 0\right\} = 5,$$
$$w_{10}^2 = 0.$$

U trećem koraku za svaki $i \in [10]$ računamo vrijednosti w_i^3 definirane relacijama:

$$w_i^3 = \min_{j \leftrightarrow i}(w_{ij} + w_j^2).$$

Tako redom dobivamo:

$$w_1^3 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^2) = \min_{j \in \{2,3,4,5\}}(w_{1j} + w_j^2) = \min\left\{w_{12} + w_2^2, w_{13} + w_3^2, w_{14} + w_4^2, w_{15} + w_5^2\right\} =$$
$$= \min\left\{1 + 15, 10 + 6, 6 + 12, 3 + 13\right\} = 16,$$
$$w_2^3 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^2) = \min_{j \in \{1,3,6\}}(w_{2j} + w_j^2) = \min\left\{w_{21} + w_1^2, w_{23} + w_3^2, w_{26} + w_6^2\right\} =$$
$$\min\left\{1 + \infty, 10 + 6, 10 + 4\right\} = 14,$$
$$w_3^3 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^2) = \min_{j \in \{1,2,4,6,7,8\}}(w_{3j} + w_j^2) = \min\left\{w_{31} + w_1^2, w_{32} + w_2^2, w_{34} + w_4^2, w_{36} + w_6^2,\right.$$
$$\left. w_{37} + w_7^2, w_{38} + w_8^2\right\} = \min\left\{10 + \infty, 10 + 15, 4 + 12, 2 + 4, 4 + 2, 1 + 8\right\} = 6,$$
$$w_4^3 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^2) = \min_{j \in \{1,3,5,8\}}(w_{4j} + w_j^2) = \min\left\{w_{41} + w_1^2, w_{43} + w_3^2, w_{45} + w_5^2, w_{48} + w_8^2\right\} =$$
$$= \min\left\{6 + \infty, 4 + 6, 2 + 13, 3 + 8\right\} = 10,$$
$$w_5^3 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^2) = \min_{j \in \{1,4,8,9\}}(w_{5j} + w_j^2) = \min\left\{w_{51} + w_1^2, w_{54} + w_4^2, w_{58} + w_8^2, w_{59} + w_9^2\right\} =$$
$$= \min\left\{3 + \infty, 2 + 12, 6 + 8, 8 + 5\right\} = 13,$$
$$w_6^3 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^2) = \min_{j \in \{2,3,7,10\}}(w_{6j} + w_j^2) = \min\left\{w_{62} + w_2^2, w_{63} + w_3^2, w_{67} + w_7^2, w_{6,10} + w_{10}^2\right\} =$$
$$= \min\left\{10 + 15, 2 + 6, 2 + 2, 5 + 0\right\} = 4,$$
$$w_7^3 = \min_{j \leftrightarrow 7}(w_{7j} + w_j^2) = \min_{j \in \{3,6,8,10\}}(w_{7j} + w_j^2) = \min\left\{w_{73} + w_3^2, w_{76} + w_6^2, w_{78} + w_8^2, w_{7,10} + w_{10}^2\right\} =$$
$$= \min\left\{4 + 6, 2 + 4, 6 + 8, 2 + 0\right\} = 2,$$

$$w_8^3 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^2) = \min_{j \in \{3,4,5,7,9,10\}}(w_{8j} + w_j^2) = \min\{w_{83} + w_3^2, w_{84} + w_4^2, w_{85} + w_5^2, w_{87} + w_7^2,$$
$$w_{89} + w_9^2, w_{8,10} + w_{10}^2\} = \min\{1+6, 3+12, 6+13, 6+2, 10+5, 9+0\} = 7,$$
$$w_9^3 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^2) = \min_{j \in \{5,8,10\}}(w_{5j} + w_j^2) = \min\{w_{95} + w_5^2, w_{98} + w_8^2, w_{9,10} + w_{10}^2\} =$$
$$= \min\{8+13, 10+8, 5+0\} = 5,$$
$$w_{10}^3 = 0.$$

U *četvrtoj fazi* za svaki $i \in [10]$ računamo vrijednosti w_i^4 definirane relacijama:

$$w_i^4 = \min_{j \leftrightarrow i}(w_{ij} + w_j^3).$$

pa redom dobivamo:

$$w_1^4 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^3) = \min_{j \in \{2,3,4,5\}}(w_{1j} + w_j^3) = \min\{w_{12} + w_2^3, w_{13} + w_3^3, w_{14} + w_4^3, w_{15} + w_5^3\} =$$
$$= \min\{1+14, 10+6, 6+10, 3+13\} = 15,$$
$$w_2^4 = \min_{j \leftrightarrow 2}(w_{2j} + w_j^3) = \min_{j \in \{1,3,6\}}(w_{2j} + w_j^3) = \min\{w_{21} + w_1^3, w_{23} + w_3^3, w_{26} + w_6^3\} =$$
$$\min\{1+16, 10+6, 10+4\} = 14,$$
$$w_3^4 = \min_{j \leftrightarrow 3}(w_{3j} + w_j^3) = \min_{j \in \{1,2,4,6,7,8\}}(w_{3j} + w_j^3) = \min\{w_{31} + w_1^3, w_{32} + w_2^3, w_{34} + w_4^3, w_{36} + w_6^3,$$
$$w_{37} + w_7^3, w_{38} + w_8^3\} = \min\{10+16, 10+14, 4+10, 2+4, 4+2, 1+7\} = 6,$$
$$w_4^4 = \min_{j \leftrightarrow 4}(w_{4j} + w_j^3) = \min_{j \in \{1,3,5,8\}}(w_{4j} + w_j^3) = \min\{w_{41} + w_1^3, w_{43} + w_3^3, w_{45} + w_5^3, w_{48} + w_8^3\} =$$
$$= \min\{6+16, 4+6, 2+13, 3+7\} = 10,$$
$$w_5^4 = \min_{j \leftrightarrow 5}(w_{5j} + w_j^3) = \min_{j \in \{1,4,8,9\}}(w_{5j} + w_j^3) = \min\{w_{51} + w_1^3, w_{54} + w_4^3, w_{58} + w_8^3, w_{59} + w_9^3\} =$$
$$= \min\{3+16, 2+10, 6+7, 8+5\} = 12,$$
$$w_6^4 = \min_{j \leftrightarrow 6}(w_{6j} + w_j^3) = \min_{j \in \{2,3,7,10\}}(w_{6j} + w_j^3) = \min\{w_{62} + w_2^3, w_{63} + w_3^3, w_{67} + w_7^3, w_{6,10} + w_{10}^3\} =$$
$$= \min\{10+14, 2+6, 2+2, 5+0\} = 4,$$
$$w_7^4 = \min_{j \leftrightarrow 7}(w_{7j} + w_j^2) = \min_{j \in \{3,6,8,10\}}(w_{7j} + w_j^3) = \min\{w_{73} + w_3^3, w_{76} + w_6^3, w_{78} + w_8^3, w_{7,10} + w_{10}^3\} =$$
$$= \min\{4+6, 2+4, 6+8, 2+0\} = 2,$$
$$w_8^4 = \min_{j \leftrightarrow 8}(w_{8j} + w_j^3) = \min_{j \in \{3,4,5,7,9,10\}}(w_{8j} + w_j^3) = \min\{w_{83} + w_3^3, w_{84} + w_4^3, w_{85} + w_5^3, w_{87} + w_7^3,$$
$$w_{89} + w_9^3, w_{8,10} + w_{10}^3\} = \min\{1+6, 3+10, 6+13, 6+2, 10+5, 9+0\} = 7,$$
$$w_9^4 = \min_{j \leftrightarrow 9}(w_{9j} + w_j^3) = \min_{j \in \{5,8,10\}}(w_{5j} + w_j^3) = \min\{w_{95} + w_5^3, w_{98} + w_8^3, w_{9,10} + w_{10}^3\} =$$
$$= \min\{8+13, 10+7, 5+0\} = 5,$$
$$w_{10}^4 = 0.$$

U *petoj fazi* za svaki $i \in [10]$ treba računati vrijednosti w_i^5 definirane relacijama:

$$w_i^5 = \min_{j \leftrightarrow i}(w_{ij} + w_j^4).$$

Za $i = 1$ dobivamo:

$$w_1^5 = \min_{j \leftrightarrow 1}(w_{1j} + w_j^4) = \min_{j \in \{2,3,4,5\}}(w_{1j} + w_j^4) = \min\left\{w_{12} + w_2^4, w_{13} + w_3^4, w_{14} + w_4^4, w_{15} + w_5^4\right\} =$$
$$= \min\left\{1+14, 10+6, 6+10, 3+12\right\} = 15,$$

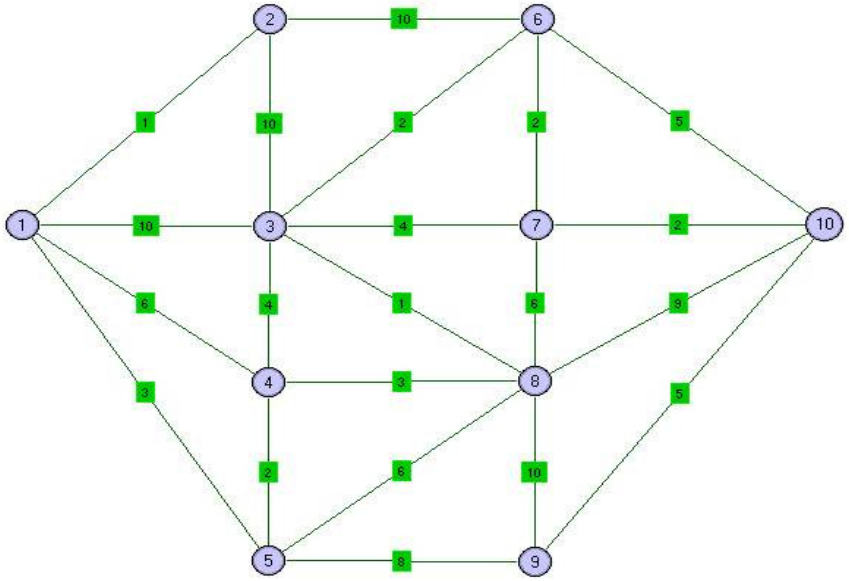
pa vrijedi jednakost

$$w_1^5 = w_1^4 = 15.$$

Stoga minimalna duljina najkraćega puta između vrhova 1 i 10 iznosi 4, a pripadna optimalna težina 15. To znači da *duljina najkraćega puta između mjesta* 1 *i mjesta* 10 *iznosi* 15 km.

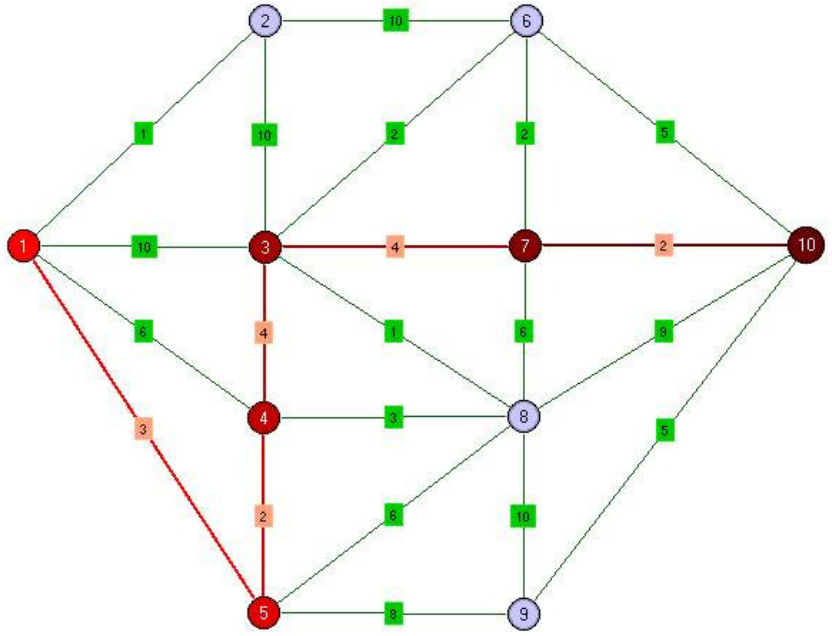
Ispišimo dobiveni najkraći put. Uočimo da se vrijednost w_i^4 = 15 dobiva ili kao zbroj w_{12} + + w_2^3 ili kao zbroj w_{12} + w_5^4. Odaberemo li prvi zbroj, slijedi da je prvi unutrašnji vrh najkraćega puta vrh 2. Vrijednost w_2^3 dobivena je kao zbroj w_{26} + w_6^2, pa je sljedeći unutrašnji vrh najkraćega puta vrh 6. Napokon, vrijednost w_6^2 dobivena je kao zbroj w_{67} + w_7^1, pa je četvrti unutrašnji vrh najkraćega puta vrh 7. Dakle, dobiveni najkraći put glasi: 1 – 2 – 6 – 7 – 10. Izborom drugoga zbroja (w_{12} + w_5^4) dobivaju se još četiri različita najkraća puta.

<u>*Rješenje pomoću programa Graph Magics*</u>: Sukladno polaznim razmatranjima iz "klasičnoga" rješenja ovoga primjera, razmatrani problem modeliramo sljedećim neusmjerenim jedinstavnim težinskim grafom *G*:



Slika 3. Model problema iz Primjera 2.

Analogno kao u odgovarajućem rješenju Primjera 1., označimo vrh 1 kao polazište (*Start Vertex*), a vrh 10 kao odredište (*End Vertex*). Primjenom procedure *Find Shortest Path* (*from Start to End Vertex*) dobivamo:



Slika 4. Rješenje Primjera 2. dobiveno programom *Graph Magics*.

Stoga dobiveni najkraći put glasi: 1 – 5 – 4 – 3 – 7 – 10 i ima težinu w_{min} = 3 + 2 + 4 + 4 + 2 = 15. To ponovno znači da *najkraća udaljenost između mjesta* 1 *i mjesta* 10 *iznosi* 15 km. No, u ovom slučaju kao rješenje nismo dobili najkraći put dobiven "klasičnim" rješenjem, ali težine obaju najkraćih putova su međusobno jednake.

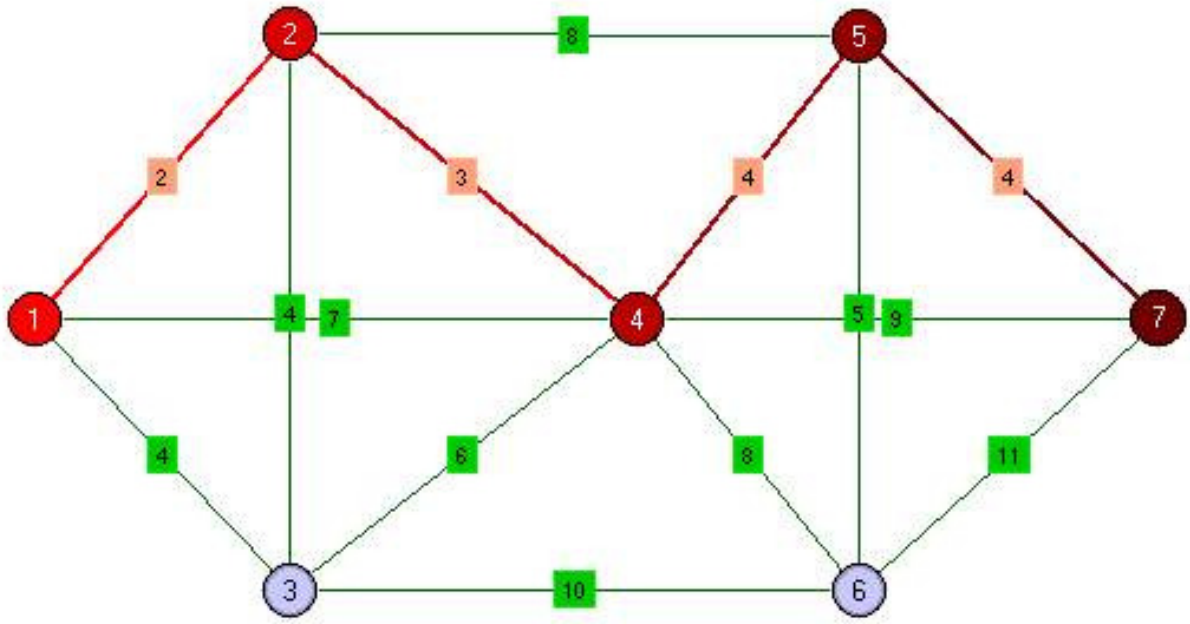
Primijetimo da, osim dvaju navedenih, u mreži postoji još nekoliko putova (s istim polazištem i odredištem) iste težine, ali različitih duljina: 1 – 5 – 4 – 3 – 6 – 7 – 10, 1 – 5 – 4 – 8 – 3 – 6 – – 7 – 10 i 1 – 5 – 4 – 8 – 3 – 7 – 10.

<u>Primjer 3.</u> Mreža dvosmjernih putova povezuje ukupno 7 gradova. Duljine [u km] izravnih dvosmjernih putova dane su u sljedećoj težinskoj matrici *W*:

$$W = \begin{bmatrix} 0 & 2 & 4 & 7 & \infty & \infty & \infty \\ 2 & 0 & 4 & 3 & 8 & \infty & \infty \\ 4 & 4 & 0 & 6 & 10 & \infty & \infty \\ 7 & 3 & 6 & 0 & 4 & 8 & 9 \\ \infty & 8 & 10 & 4 & 0 & 5 & 4 \\ \infty & \infty & \infty & 8 & 5 & 0 & 11 \\ \infty & \infty & \infty & 9 & 4 & 11 & 0 \end{bmatrix}.$$

Modelirajmo navedenu mrežu putova odgovarajućim grafom, pa primjenom odgovarajućega algoritma odredimo najkraći put između gradova 1 i 7.

Rezultat (dobiven programom *Graph Magics*): Pripadni model je ponovno neusmjereni jednostavni težinski graf s ukupno 7 vrhova. Stoga možemo uzeti $V = [7]$. Skup E se lako "očita" iz težinske matrice W. Pritom $w_{ij} = \infty$ znači da mjesta i i j nisu povezana izravnim dvosmjernim putem, a $w_{ii} = 0$ stavljamo dogovorno.



Slika 5. Rješenje Primjera 3. dobiveno programom *Graph Magics*.

Traženi najkraći put je 1 – 2 – 4 – 5 – 7 i njegova je težina $w_{min} = 13$. Dakle, *najkraća udaljenost gradova* 1 *i* 7 *iznosi* 13 km.

U praksi se nerijetko pojavljuje problem određivanja *najkraćih udaljenosti* između *bilo kojih* dvaju mjesta neke mreže (putova, željezničkih pruga itd.) Budući da ukupan broj takvih udaljenosti ovisi o ukupnom broju mjesta (n) u mreži, dobivene rezultate vrlo je pregledno zapisati u matričnom obliku. Tako se dobiva tzv. *matrica najkraćih udaljenosti* D_{min}. To je kvadratna matrica reda n koja, dogovorno, na glavnoj dijagonali ima same nule (jer se pretpostavlja da je udaljenost mjesta i do samoga sebe jednaka 0). U slučaju neusmjerenih jednostavnih težinskih grafova ta je matrica uvijek simetrična, dok u slučaju jednostavnih težinskih digrafova ta matrica nije simetrična.

Primjer 4. Za mrežu iz Primjera 3. odredimo matricu najkraćih udaljenosti $D_{min} = [d_{ij}]$, gdje element d_{ij} označava duljinu [u km] najkraćega puta iz grada i u grad j.

Rješenje pomoću programa Graph Magics: Koristit ćemo proceduru *Find All Shortest Path* (*from Start Vertex*). Najprije kao polazište (*Start Vertex*) označimo vrh 1. Desnom tipkom miša kliknemo na bilo koju bjelinu unutar radne površine. Na dobivenom izborniku odaberemo opciju *Find*, a potom podopciju *All Shortest Path* (*from Start Vertex*). Tako dobijemo:

```
Results:
#        Vertex        Total Distance        Came from Vertex
1        2             2                     1
2        3             4                     1
3        4             5                     2
```

```
Results:
#        Vertex        Total Distance        Came from Vertex
4        5             9                     4
5        6             13                    4
6        7             13                    5
```

Kako smo već istakli, budući da je graf G neusmjeren jednostavan težinski graf, matrica D_{min} bit će simetrična matrica koja će na glavnoj dijagonali imati same nule. Njezin prvi redak (odnosno, prvi stupac) je vektor (0, 2, 4, 5, 9, 13, 13). Potom kao polazište označimo 2, pa sljedećom primjenom navedene procedure dobivamo:

```
Results:
#        Vertex        Total Distance        Came from Vertex
1        1             2                     2
2        3             4                     2
3        4             3                     2
4        5             7                     4
5        6             11                    4
6        7             11                    5
```

Stoga je drugi redak (odnosno, drugi stupac) matrice D_{min} vektor (2, 0, 4, 3, 7, 11, 11). Ponavljanjem opisanoga postupka za svaki od preostalih 5 vrhova[6] konačno dobivamo:

$$D_{min} = \begin{bmatrix} 0 & 2 & 4 & 5 & 9 & 13 & 13 \\ 2 & 0 & 4 & 3 & 7 & 11 & 11 \\ 4 & 4 & 0 & 6 & 10 & 10 & 14 \\ 5 & 3 & 6 & 0 & 4 & 8 & 8 \\ 9 & 7 & 10 & 4 & 0 & 5 & 4 \\ 13 & 11 & 10 & 8 & 5 & 0 & 9 \\ 13 & 11 & 14 & 8 & 4 & 9 & 0 \end{bmatrix}.$$

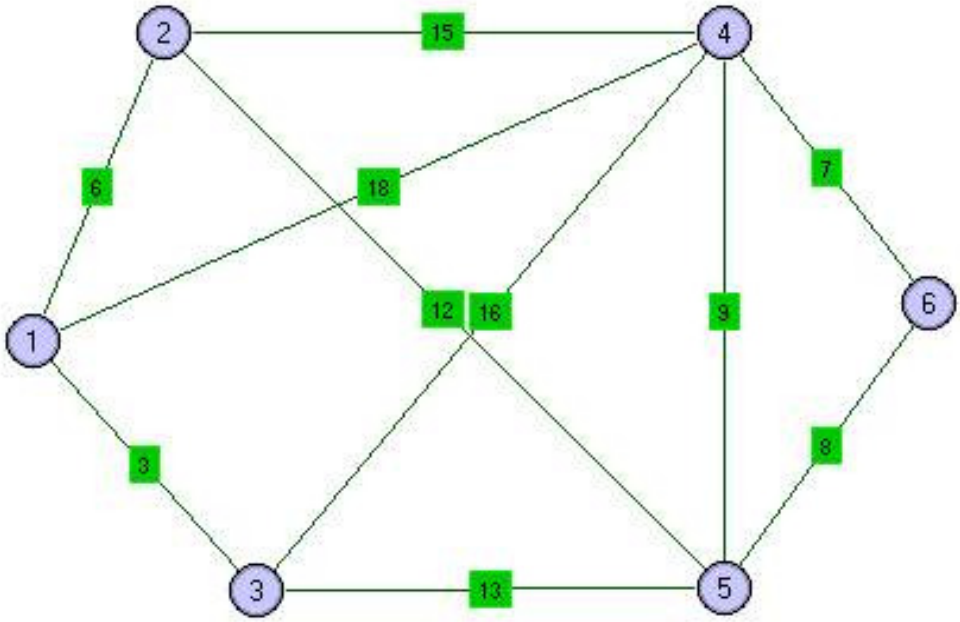
<u>Primjer 5.</u> Mreža dvosmjernih putova između 6 različitih lokacija definirana je matricom izravnih udaljenosti D pojedinih lokacija:

$$D = \begin{bmatrix} 0 & 6 & 3 & 18 & \infty & \infty \\ 6 & 0 & \infty & 15 & 12 & \infty \\ 3 & \infty & 0 & 16 & 13 & \infty \\ 18 & 15 & 16 & 0 & 9 & 7 \\ \infty & 12 & 13 & 9 & 0 & 8 \\ \infty & \infty & \infty & 7 & 8 & 0 \end{bmatrix}.$$

(Uobičajeno, $d_{ij} = \infty$ znači da *ne postoji* izravan dvosmjerni put iz mjesta i u mjesto j, a dogovorno stavljamo $d_{ii} = 0$, $\forall i \in$ [6].) Modelirajmo navedenu mrežu putova odgovarajućim grafom, pa odredimo matricu najkraćih udaljenosti D_{min} lokacija koje tvore mrežu..

<u>*Rezultat*</u>: Mrežu možemo modelirati sljedećim neusmjerenim jednostavnim težinskim grafom G takvim da vrh i označava lokaciju i:

[6] Iako formalno nije potrebno provesti postupak za posljednji vrh (7), praktično ga je dobro povesti radi provjere ispravnosti ranije dobivenih rezultata.



Slika 6. Model mreže iz Primjera 5.

Primjenom procedure *Find All Shortest Path* (*from Start Vertex*) opisane u prethodnom primjeru dobivamo:

$$D_{min} = \begin{bmatrix} 0 & 6 & 3 & 18 & 16 & 24 \\ 6 & 0 & 9 & 15 & 12 & 20 \\ 3 & 9 & 0 & 16 & 13 & 21 \\ 18 & 15 & 16 & 9 & 0 & 7 \\ 16 & 12 & 13 & 9 & 0 & 8 \\ 24 & 20 & 21 & 7 & 8 & 0 \end{bmatrix}.$$

Na temelju zadane matrice najkraćih putova D_{min} vrlo jednostavno se može odrediti najkraći put između bilo kojih dvaju lokacija određene mreže. Pokažimo to na sljedećem primjeru.

<u>Primjer 6.</u> Isključivo koristeći rezultat prethodnoga primjera odredimo najkraći put između lokacija:

a) 1 i 6;
b) 2 i 6.

<u>*Rješenje*</u>: **a)** U prvom retku (stupcu) matrice D_{min} navedene su težine najkraćih putova od vrha 1 do svakoga od preostalih vrhova promatrane mreže. Analogno, u šestom retku (stupcu) navedene su težine najkraćih putova od vrha 6 do svakoga od preostalih vrhova promatrane mreže. Zbrojimo li prvi i šesti redak, dobit ćemo vektor čija će i – ta komponenta biti *težina najkraćega puta između vrhova* 1 i 6 *koji nužno prolazi vrhom i*:

$$d_{1-6} = (24, 26, 24, 25, 24, 24).$$

Konkretno, težina najkraćega puta između vrhova 1 i 6 jednaka je prvoj, odnosno šestoj komponenti vektora d_{1-6}, tj. 24. Odredimo sve preostale komponente vektora d_{1-6} koje su jednake 24. To su treća i peta komponenta. Stoga traženi najkraći put između vrhova 1 i 6 prolazi vrhovima 3 i 5, tj. traženi najkraći put glasi: 1 – 3 – 5 – 6.

b) Postupimo analogno kao u **a)**, samo što u ovom slučaju zbrajamo drugi redak (stupac) i šesti redak (stupac). Dobivamo vektor

$$d_{2-6} = (30, 20, 30, 22, 20, 20).$$

Težina najkraćega puta između vrhova 2 i 6 jednaka je drugoj, odnosno šestoj komponenti vektora d_{2-6}, tj. 20 Odredimo sve preostale komponente vektora d_{2-6} koje su jednake 20. Od svih njih jedino je peta komponenta jednaka 20. Stoga je traženi najkraći put: 2 – 5 – 6.

Osim u neusmjerenom, najkraći put između dvaju vrhova može se odrediti i u digrafu. Primjer primjene toga problema u sklopu problema optimalne zamjene opreme navest ćemo u točki 1.5., a ovdje ilustrativno navodimo sljedeći primjer.

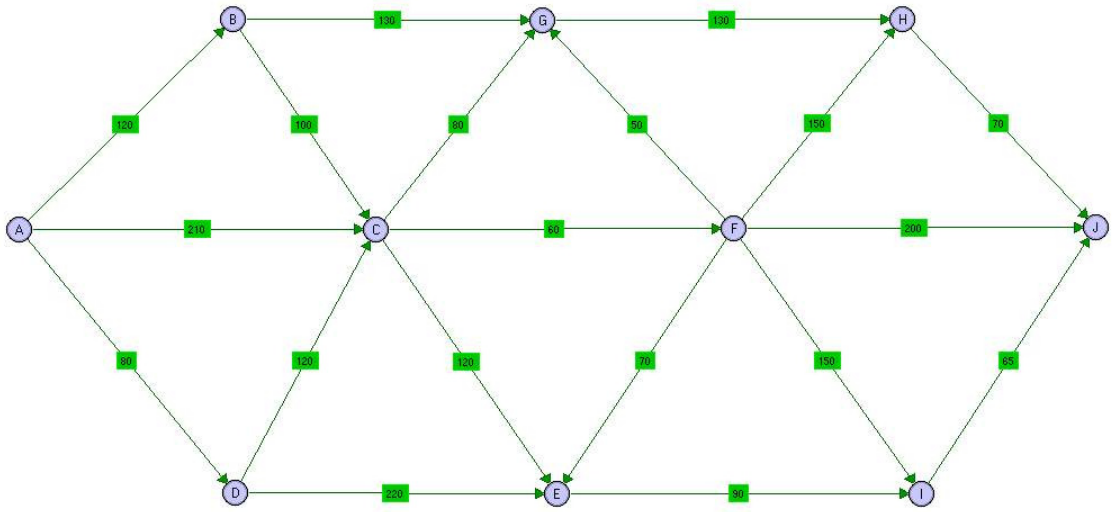
Primjer 7. Visoka poslovna škola "*Naša mala akademija*" iz Frkljevaca organizira mini-uličnu utrku svojih studenata. Start utrke je ispred zgrade škole na lokaciji *A*, dok je cilj ispred zgrade općinskoga poglavarstva na lokaciji *J*. U sljedećoj su tablici navedena procijenjena najkraća vremena (iskazana u sekundama) potrebna za prolazak dionice određene pojedinim lokacijama.

dozvoljeni smjer prolaska	*procijenjeno najkraće vrijeme* [s]
(*A*, *B*)	120
(*A*, *C*)	210
(*A*, *D*)	80
(*B*,*C*)	100
(*B*, *G*)	130
(*C*, *E*)	120
(*C*, *F*)	60
(*C*, *G*)	80
(*D*, *C*)	120
(*D*, *E*)	220
(*E*, *I*)	90
(*F*, *E*)	70
(*F*, *G*)	50
(*F*, *H*)	150
(*F*, *I*)	150
(*F*, *J*)	200
(*G*, *H*)	130
(*H*, *J*)	70
(*I*, *J*)	65

Svaki trkač može sâm odabrati rutu kojom će od starta doći do cilja, ali uz nužno uvažavanje dozvoljenih smjerova prolaska. Mateja, najbolja trkačica na školi, procijenjuje da svaku pojedinu dionicu utrke može istrčati u procijenjenom najkraćem vremenu, pa razrađuje strategiju kojim dionicama treba trčati tako da što prije stigne na cilj. Modelirajmo navedeni problem odgovarajućim grafom, pa odredimo optimalnu rutu trčanja i izračunajmo pripadno optimalno vrijeme.

Rješenje: Budući da su zadani dozvoljeni smjerovi prolaska, zadani problem modeliramo jednostavnim težinskim digrafom. Vrhovi toga digafa predstavljaju zadane lokacije, pa ćemo za skup vrhova digrafa uzeti $V = \{A, B, C, D, E, F, G, H, I\}$. Lukovi digrafa označat će dozvoljeni smjer između pojedinih lokacija, a težina svakoga od njih procijenjeno najkraće vrijeme prolaska dionicom između odgovarajućih lokacija. Tako se problem svodi na određivanje najkraćega puta u digrafu između vrha koji predstavlja lokaciju *A* i vrha koji predstavlja lokaciju *J*.

Navedeni se problem, inače, uobičajeno rješava Dijkstrinim algoritmom, a mi ćemo ga riješiti koristeći računalni program *Graph Magics*. Kao matematički model razmatranoga problema dobiva se sljedeći digraf s ukupno 10 vrhova i 19 lukova:



Slika 7. Model problema iz Primjera 7.

Označavanjem vrha *A* kao polazišta (*Start Vertex*), vrha *J* kao odredišta (*End Vertex*) i primjenom procedure *Find Shortest Path form Start Vertex to End* dobivamo traženu optimalnu rutu: *A – B – G – H – J* čija je težina 450. Prema tome, procijenjeno optimalno ukupno vrijeme za istrčavanje utrke iznosi 450 sekundi, odnosno 7.5 minuta.

Zadaci za vježbu

1. 6 različitih mjesta: 1, 2, …, 6 međusobno je povezano mrežom cesta. Međusobne izravne udaljenosti pojedinih mjesta zadane su matricom

$$D = \begin{bmatrix} 0 & 6 & 9 & 11 & 5 & 9 \\ 6 & 0 & 3 & 6 & 5 & 2 \\ 9 & 3 & 0 & 1 & 4 & 4 \\ 11 & 6 & 1 & 0 & 5 & 6 \\ 5 & 5 & 4 & 5 & 0 & 8 \\ 9 & 2 & 4 & 6 & 8 & 0 \end{bmatrix}.$$

gdje je d_{ij} duljina [u km] izravne dvosmjerne ceste između mjesta i i mjesta j. "Klasično" i pomoću računalnoga programa *Graph Magics* odredite najkraći put od mjesta 1 do mjesta 6, te njegovu duljinu.

2. U mreži dvosmjernih cesta iz prethodnoga zadatka odredite:
a) najkraću udaljenost između mjesta 1 i svakoga od ostalih pet mjesta, te pripadni najkraći put kojim se postiže ta udaljenost;
b) najkraći put između mjesta 1 i mjesta 5 koji nužno prolazi mjestom 2, te njegovu duljinu.

3. 8 različitih lokacija: A, B, C, D, E, F, G i H povezano je mrežom dvosmjernih cesta. Duljine pojedinih cesta [u km] su: $|AB| = 5$, $|AC| = |CE| = |EF| = 6$, $|BD| = |EG| = 7$, $|BE| = |DF| = 8$, $|DG| = 9$, $|FH| = 13$ i $|GH| = 11$. "Klasično" i pomoću računalnoga programa *Graph Magics* odredite najkraći put od lokacije A do lokacije H, te njegovu duljinu.

4. Riješite prethodni zadatak ako su duljine pojedinih cesta [u km]:
a) $|AB| = |FH| = 8$, $|AC| = |BF| = |EH| = 5$, $|AD| = |BE| = |DF| = 7$, $|CF| = 10$, $|CG| = |FH| = 8$, $|DG| = 4$ i $|GH| = 11$;
b) $|AB| = |CE| = 5$, $|AC| = |CF| = 6$, $|BE| = 7$, $|BD| = 14$, $|DH| = 12$, $|EG| = |GH| = 10$ i $|FG| = 8$.

5. U mreži dvosmjernih cesta iz Zadatka 5. odredite:
a) najkraću udaljenost između lokacije C i svake od ostalih sedam lokacija, te pripadni najkraći put kojim se postiže ta udaljenost;
b) najkraći put između lokacija B i G koji nužno prolazi lokacijom C.

6. 7 različitih lokacija: A, B, C, D, E, F i G povezano je mrežom dvosmjernih cesta. Procijenjena najkraća vremena prolaska [u minutama] pojedinim cestama su: $|AB| = 15$, $|AC| = 10$, $|AD| = 25$, $|BD| = 8$, $|BE| = |DF| = 17$, $|CD| = |DE| = 14$, $|DG| = 20$, $|EG| = 9$ i $|FG| = 13$. "Klasično" i pomoću računalnoga programa *Graph Magics* odredite put između lokacija A i G za čiji je prolazak potrebno najmanje vremena, pa izračunajte pripadno optimalno vrijeme.

7. Riješite prethodni zadatak ako su procijenjena najkraća vremena prolaska [u minutama] pojedinim cestama: $|AB| = |CD| = |EF| = |FG| = 15$, $|AC| = 20$, $|BD| = 22$, $|BE| = |DF| = 25$, $|CE| = 30$, i $|EG| = 27$.

9. U mreži dvosmjernih cesta iz Zadatka 7. odredite:
a) najkraće vrijeme potrebno za dolazak s lokacije D na svaku od ostalih šest lokacija, te pripadni najkraći put kojim se postiže to vrijeme;
b) najkraće vrijeme potrebno za dolazak s lokacije B na lokaciju F preko lokacije D.

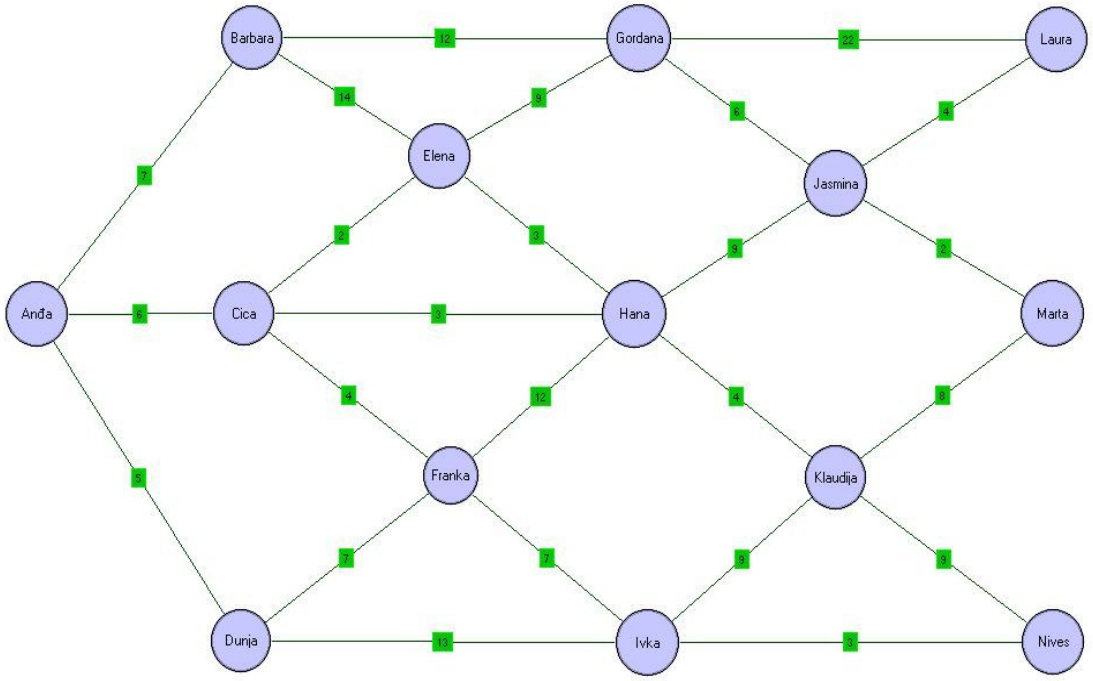
9. Dvije studentice – Lorena i Morena – stanuju u kućama smještenima redom na lokacijama A i B. Svakoga dana njih dvije idu pješice na veleučilište smješteno na lokaciji H koristeći ulice (dozvoljene za prolaz pješaka u oba smjera) određene lokacijama D, E, F i G. Ako su duljine [u m] pojedinih ulica $|AC| = 900$, $|BD| = |EH| = 800$, $|CE| = |DG| = 200$, $|CF| = 500$, $|DF| = 400$, $|FH| = 600$ i $|GH| = 700$, "klasično" i pomoću programa *Graph Magics* odredite koja od njih dvije stanuje bliže veleučilištu.

10. Riješite prethodni zadatak ako su duljine [u m] pojedinih ulica $|AC| = 500$, $|AD| = |CE| = |EG| = 600$, $|BC| = |DF| = 400$, $|BD| = |FG| = 700$, $|EH| = |FH| = 1000$ i $|GH| = 300$.

11. Tri studenta – Dinko, Hinko i Vinko – stanuju u stambenim zgradama smještenima redom na lokacijama *A*, *B* i *C*. Svakoga dana njih trojica idu pješice na fakultet smješten na lokaciji *J* koristeći ulice (dozvoljene za prolaz pješaka u oba smjera) određene lokacijama *D*, *E*, *F*, *G*, *H* i *I*. Ako su duljine [u m] pojedinih ulica $|AE| = 200$, $|BD| = |FG| = 120$, $|CF| = |DG| = |GJ| = 250$, $|DE| = |EH| = 90$, $|DF| = 130$, $|EG| = 150$, $|FI| = 80$, $|GH| = 100$, $|GI| = 110$, $|HJ| = 300$ i $|IJ| = 280$, "klasično" i pomoću programa *Graph Magics* odredite koji od njih trojice stanuje najbliže fakultetu.

12. Sedam željezničkih kolodvora: *A*, *B*, *C*, *D*, *E*, *F* i *G* međusobno je povezano mrežom jednokolosječnih pruga. Prosječne duljine putovanja [u minutama] brzim vlakom između pojedinih postaja su: $|AB| = |DE| = 30$, $|AC| = 75$, $|AD| = |BD| = |CE| = 35$, $|BE| = 50$, $|BF| = 65$, $|CD| = 40$, $|CG| = 60$, $|EF| = 25$ i $|FG| = 20$. Pritom se u svakom od kolodvora *B*, *C*, *D*, *E* i *F* brzi vlak zadržava točno dvije minute radi otpreme i prihvata putnika. "Klasično" i pomoću računalnoga programa *Graph Magics* odredite najkraći put između kolodvora *A* i kolodvora *G*, te pripadno optimalno vrijeme potrebno za prelazak toga puta.

13. Svakoga dana dežurna veleučilišna "tračerica" Anđa prenosi svježi sočni trač svojim kolegicama s veleučilišta, a one ga potom prenose dalje. Donji graf pokazuje rute kojima se trač širi, a pripadne težine bridova označavaju vrijeme potrebno da se trač prenese između odgovarajućih osoba.



Slika 8. Model problema iz zadatka 13.

a) Odredite najkraće vrijeme potrebno da svaka pojedina osoba dozna svježi trač.
b) Ispišite najkraću "rutu širenja trača" putem koje Marta doznaje svježi trač.
c) Ukoliko se Cica razboli, pa određenoga dana ne dođe na nastavu, odredite novu najkraću rutu putem koje će Marta tada doznati svježi trač i pripadno optimalno vrijeme.
d) Riješite **a)** podzadatak u slučaju da se Anđa razboli, pa "dužnost" dežurne "tračerice" preuzme Laura.
Napomena: Sve podzadatke riješite koristeći računalni program *Graph Magics*.

1.2. Problem naprtnjače (ranca)

Problem naprtnjače (ranca) je tipičan problem tzv. *kombinatorne optimizacije*. Naziv problema potječe od originalnoga problema kojega je prije 20 godina postavio američki informatičar Robert Sedgewick[7]. Iskazan u slobodnijoj formi, taj problem glasi:

Lopov Kradljivko Lupeškić kreće u pljačku kuće bogataša Tajkunčića s naprtnjačom obujma M kubičnih jedinica. Uspješno provalivši u bogataševu kuću, Kradljivko u kući nalazi na puno vrijedniji i veći plijen nego što ga može ponijeti u svojoj naprtnjači. Svaki potencijalni sastojak plijena ima svoj *obujam* i *procijenjenu vrijednost*. Kradljivkovi džepovi su propusni, pa niti jedan sastojak plijena ne smije staviti u džep jer bi ga izgubio putem. Stoga sve što uzme mora staviti isključivo u naprtnjaču. Pitanje na koje Kradljivko mora naći odgovor je: Kako napuniti naprtnjaču sa što više predmeta tako da ukupna procijenjena vrijednost svih predmeta bude što veća?[8] (Pretpostavljamo da nijedan predmet nije moguće razdijeliti na manje dijelove.)

U daljnjim ćemo razmatranjima najveću moguću masu (obujam i sl.) namirnica koje možemo staviti u naprtnjaču kraće nazivati *kapacitet naprtnjače*. Npr. kažemo li da je kapacitet naprtnjače 10 kg, to znači da najveća moguća masa svih stvari stavljenih u naprtnjaču ne smije biti strogo veća od 10 kg. Pod *količinom* pojedine namirnice podrazumijevat ćemo određeni broj komada (ne: masu!) dotične namirnice, dok će *jedinična cijena* i *jedinična masa* namirnice biti cijena, odnosno masa jednoga komada te namirnice. Ne istaknemo li drugačije, pretpostavljat ćemo da niti jedan komad niti jedne vrste namirnica ne možemo podijeliti na manje dijelove.

Istaknimo da se *vrijednost* neke robe obično definira kao umnožak jedinične cijene i količine te robe. Primjerice, ako cijena jedne krafne s čokoladom u pekari "*Mljac–mljac*" iznosi 4,00 kn, onda je vrijednost 5 krafni s čokoladom jednaka $5 \cdot 4{,}00 = 20{,}00$ kn. Vrijednost robe može se izračunati i kao umnožak mase i jedinične cijene te robe. Primjerice, ako je cijena jednoga kilograma bijeloga kruha u pekari "*Mljac–mljac*" 8,00 kn, onda je vrijednost 5 tona toga kruha jednaka 5 000 [kg] · 8,00 [kn/kg] = 4.000,00 kn. Pod pojmom *jedinična vrijednost* podrazumijeva se vrijednost jednoga komada neke robe. Lako se vidi da jedinična vrijednost i jedinična cijena imaju iste mjerne brojeve, ali različite ekonomske interpretacije.

Uvodno spomenimo i ekvivalentnu formulaciju problema naprtnjače u terminima *teorije odlučivanja*:

Za zadane strogo pozitivne realne brojeve M i V, te skup namirnica S, odrediti barem jedan neprazan podskup S_1 skupa S takav da je ukupna vrijednost svih elemenata skupa S_1 jednaka najmanje V, a njihova ukupna masa najviše jednaka M.[9]

---

[7] Robert Sedgewick, profesor računalskih znanosti na Univerzitetu u Princetonu i predsjednik upravnoga odbora tvrtke *Adobe Systems* najpoznatije po softveru *Adobe Acrobat Reader*.

[8] Današnja uobičajena formulacija (jednodimenzionalnoga) problema ruksaka je bitno manje "kriminalistička": Zadan je skup namirnica od kojih svaka ima svoju masu i svoju cijenu. Na raspolaganju nam je ruksak u kojega možemo staviti predmete ukupne mase najviše M. Kako treba odabrati namirnice tako da njihova ukupna masa bude najviše M, a njihova ukupna vrijednost što je moguće veća?

[9] Ovdje je nemogućnost uzimanja dijela pojedine namirnice iskazana zahtjevom da skup S_1 nužno mora biti podskup skupa S.

Primjer 1. Na raspolaganju nam je po jedan komad svake od 5 vrsta namirnica. Podaci o jediničnoj cijeni i jediničnoj masi svake namirnice navedeni su u donjoj tablici.

namirnica	*jedinična cijena* [€]	*jedinična masa* [kg]
N_1	4	12
N_2	2	1
N_3	10	4
N_4	1	1
N_5	2	2

Na raspolaganju nam je naprtnjača kapaciteta 15 kg. Treba odabrati namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne premaši kapacitet naprtnjače.

Kasnije ćemo pokazati da se *optimalna vrijednost* 15 € dobije izborom po jednoga komada svake od namirnica osim namirnice N_1.

Formulirajmo sada <u>*opći* 0–1 *problem naprtnjače*</u>. Pretpostavimo da nam je na raspolaganju po jedan komad svake od ukupno n različitih vrsta namirnica, te naprtnjača kapaciteta M. Radi određenosti, označimo namirnice s 1, 2, …, n. Za svaki $i \in [n]$ označimo s p_i jediničnu cijenu, a s m_i jediničnu masu namirnice i, pa definirajmo varijablu x_i s:

$$x_i = \begin{cases} 0, & \text{ako namirnicu } i \text{ ne stavljamo u naprtnjaču;} \\ 1, & \text{ako namirnicu } i \text{ stavljamo u naprtnjaču.} \end{cases}$$

Varijable x_i, $i \in [n]$, pripadaju tipu tzv. <u>*varijabli odlučivanja*</u> jer, prema definiciji, označavaju jesmo li odlučili staviti namirnicu i u naprtnjaču ili nismo. Uz tako definirane oznake i varijable, opći 0–1 problem naprtnjače možemo *matematički modelirati* na sljedeći način:

$$\text{maksimizirati } f(x_1, \ldots, x_n) = \sum_{i=1}^{n} p_i \cdot x_i$$

pod uvjetima

$$\begin{aligned} &\sum_{i=1}^{n} m_i \cdot x_i \le M; \\ &x_i \in \{0,1\}, \text{ za svaki } i \in [n]. \end{aligned}$$

Osim formuliranoga 0 – 1 problema naprtnjače, u praksi se često pojavljuju <u>*otvoreni*</u> i <u>*zatvoreni problem naprtnjače*</u>. U oba slučaja pretpostavljamo da raspolažemo s *barem jednim* komadom svake pojedine vrste namirnica. (Ponekad pretpostavljamo i da barem jedan komad barem jedne vrste namirnica možemo podijeliti na manje dijelove.) Tada za svaki $i \in [n]$ definiramo varijablu x_i kao *ukupnu količinu namirnice i koju stavljamo u naprtnjaču*. Vrijednosti varijable x_i obično su prirodni brojevi ili nula, ali u nekim slučajevima mogu biti i konačni decimalni brojevi (npr. stavimo li u naprtnjaču pola kilograma kruha).

<u>*Otvoreni problem naprtnjače*</u> pretpostavlja da raspolažemo s proizvoljnom količinom svake vrste namirnica, tj. da (teoretski) imamo na raspolaganju beskonačno mnogo komada svake vrste namirnica. Stoga taj problem možemo *matematički modelirati* na sljedeći način:

$$\text{maksimizirati } f(x_1, \ldots, x_n) = \sum_{i=1}^{n} p_i \cdot x_i$$

pod uvjetima

$$\begin{aligned} &\sum_{i=1}^{n} m_i \cdot x_i \le M; \\ &x_i \in \mathbf{N}_0, \text{ za svaki } i \in [n] \end{aligned}.$$

Ukoliko je dozvoljeno barem jedan komad barem jedne namirnice podijeliti na manje dijelove, drugi uvjet prelazi u:

$$x_i \in \mathbf{Q}^+, \text{ za svaki } i \in [n].^{10}$$

Za razliku od otvorenoga, <u>*zatvoreni problem naprtnjače*</u> pretpostavlja da nam je na raspolaganju ograničena količina svake pojedine vrste namirnica, pa neka za svaki $i \in [n]$ varijabla b_i označava ukupnu raspoloživu količinu namirnice i. Tada zatvoreni problem naprtnjače možemo *matematički modelirati* na sljedeći način:

$$\text{maksimizirati } f(x_1, \ldots, x_n) = \sum_{i=1}^{n} p_i \cdot x_i$$

pod uvjetima

$$\begin{aligned} &\sum_{i=1}^{n} m_i \cdot x_i \le M \\ &x_i \in [b_i] \cup \{0\}, \text{ za svaki } i \in [n] \end{aligned}.$$

Ukoliko barem jedan komad barem jedne namirnice možemo podijeliti na manje dijelove, drugi uvjet prelazi u

$$x_i \in \mathbf{Q}^+,$$

a, sukladno tome, i vrijednosti varijable b_i mogu biti strogo pozitivni racionalni brojevi.

Spomenimo i da se u *kriptografiji*[11] razmatra sljedeći poseban slučaj 0–1 problema naprtnjače, tzv. <u>*problem zbroja elemenata podskupa*</u> naročito važan za mogućnosti stvaranja *privatnih* i *javnih ključeva* za dešifriranje podataka:

Za zadani skup *cijelih* brojeva S i zadani *cijeli* broj A ispitati postoji li (i, ako postoji, odrediti) neprazan podskup S_1 skupa S takav da je zbroj svih elemenata skupa S_1 identički jednak A. (Posebno razmotriti slučaj kad skup S sadrži i negativne cijele brojeve i kad je $A = 0$.)

Iako je riječ o *problemu odlučivanja*, a ne o optimizacijskom problemu, i navedeni se problem obično rješava metodama dinamičkoga programiranja. Ta klasa problema, zajedno s općim problemom naptnjače, pripada u tzv. <u>*NP – teške probleme*</u>, dok ekvivalentna formulacija

[10] Pritom je $\mathbf{Q}^+ := \{x \in \mathbf{Q}: x \ge 0\}$.

[11] Grubo govoreći, *kriptografija* je znanost o različitim mogućnostima sakrivanja (šifriranja) podataka, a osobitu primjenu ima vojnim i informacijskim znanostima.

problema naprtnjače u teoriji odlučivanja pripada u tzv. *NP – potpune probleme*. Naime, pokazuje se da ne postoji algoritam za rješavanje navedenih problema čija je složenost polinomijalna[12], iako postoji algoritam za njihovo rješavanje čija je složenost pseudopolinomijalna. Detalji se mogu naći u odgovarajućoj informatičkoj literaturi.

U rješenjima sljedećih primjera navodimo i način rješavanja problema naprtnjače pomoću programskoga paketa *Winqsb*. Taj paket sadrži potprogram *DP* čija je jedna od varijanti mogućnost rješavanja problema naprtnjače i analiza dobivenoga rezultata, čime se u većini slučajeva može znatno ubrzati postupak rješavanja. Nedostatak navedenoga programa je nemogućnost ispisivanja više optimalnih izbora namirnica u slučajevima u kojima takvi izbori postoje, pa prigodom primjene toga programa na rješavanje problema dinamičkoga programiranja treba imati na umu da rješenje koje ispiše program ne mora biti jedinstveno.

Matematički model Primjera 1: Za svaki $i \in [5]$ definirajmo varijablu odlučivanja x_i s:

$$x_i = \begin{cases} 0, & \text{ako namirnicu } i \text{ ne stavljamo u naprtnjaču;} \\ 1, & \text{ako namirnicu } i \text{ stavljamo u naprtnjaču.} \end{cases}$$

Stoga traženi *matematički model* glasi:

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 4 \cdot x_1 + 2 \cdot x_2 + 10 \cdot x_3 + 1 \cdot x_4 + 2 \cdot x_5$$

pod uvjetima

$$12 \cdot x_1 + 1 \cdot x_2 + 4 \cdot x_3 + 1 \cdot x_4 + 2 \cdot x_5 \leq 15$$

$$x_i \in \{0, 1\}, \text{ za svaki } i \in [5].$$

"*Klasično*" *rješenje*: Za svaki $i \in [5]$ označimo sa S_i skup koji tvore namirnice $N_1, N_2, \ldots, N_i$. Tako je npr. $S_1 = \{N_1\}$, $S_2 = \{N_1, N_2\}$, $S_3 = \{N_1, N_2, N_3\}$ itd. Dodatno definirajmo $S_0 := \varnothing$. Nadalje, za svaki $i \in [5]$ označimo s c_i jediničnu cijenu, a s m_i jediničnu masu namirnice N_i. Za bilo koji $m \in [15]$ definiramo:

S_i^m = skup svih nepraznih podskupova skupa $S_i = \{N_1, \ldots, N_i\}$ takvih da je ukupna masa elemenata svakoga podskupa najviše m kg.

Npr. za $i = 4$, $m = 2$ dobivamo skup S_4^2 kojega tvore svi neprazni podskupovi skupa $S_4 = \{N_1, N_2, N_3, N_4\}$ takvi da je ukupna masa elemenata svakoga od njih najviše 2 kg. Iz tablice je razvidno da su ti podskupovi $\{N_1\}$ (ukupna masa: 1 kg), $\{N_4\}$ (ukupna masa: 1 kg) i $\{N_1, N_4\}$ (ukupna masa: 1 + 1 = 2 kg). Stoga je $S_4^2 = \{\{N_1\}, \{N_4\}, \{N_1, N_4\}\}$. Korisno je primijetiti da izravno iz definicije skupova S_i slijedi skupovna inkluzija $S_{i-1}^m \subseteq S_i^m$.

Sada definiramo realnu funkciju dvije realne varijable $V : \mathbf{R}^2 \to \mathbf{R}$ s

[12] Algoritam ima *polinomijalnu* složenost ako je ukupan broj njegovih operacija (a time i brzina izvršavanja programa) razmjeran nekoj potenciji ukupnoga broja ulaznih podataka.

$$V(i, m) = \max_{S_1 \in S_i^m} \left\{ \sum_j c_j : N_j \in S_1 \right\}.$$ [13]

Iako navedeni formalni zapis izgleda vrlo komplicirano, on u osnovi izriče jednostavno razmatranje. Promatramo, dakle, skup S_i^m. Prema definiciji toga skupa, njegovi elementi su također skupovi, i to neprazni (svaki od njih sadrži barem jednu namirnicu). Zasebno promatramo svaki element skupa S_i^m i izračunamo ukupnu vrijednost svih namirnica koje tvore taj element. Budući da nam je na raspolaganju točno jedan komad svake pojedine vrste namirnica, za svaki $i \in [5]$ vrijednost namirnice N_i iznosi c_i €. Stoga je ukupna vrijednost svakoga elementa skupa S_i^m jednaka zbroju cijena svih namirnica koje tvore taj skup. Upravo tu činjenicu formalno reprezentira izraz naveden unutar vitičastih zagrada. Tako dobivamo onoliko (općenito, realnih) brojeva koliko ima različitih elemenata skupa S_i^m, pa među njima sigurno postoji najveći[14] i njega uzimamo za vrijednost funkcije $V(i, m)$.

Vrijednosti funkcije $V(i, m)$ za svaki $i \in [5]$ i svaki $m \in [15] \cup \{0\}$ računat ćemo pomoću sljedeće rekurzivne jednakosti:

$$V(i, m) = \max\{V(i-1, m), c_i + V(i-1, m-m_i)\}$$

uz početne uvjete (koji vrijede za sve dopustive i i m)

$$V(0, m) = V(i, 0) = 0,$$
$$(V(i, m) = -\infty) \Leftrightarrow (m < 0),$$

Komentirajmo sve navedene jednakosti. Najprije promotrimo početne uvjete. Uvjet $V(0, m) = 0$ znači da promatramo najveću moguću ukupnu vrijednost elemenata skupa S_0^m. Skup S_0^m, prema definiciji, sadrži sve neprazne podskupove skupa S_0 čija je ukupna masa najviše m kg. No, $S_0 = \varnothing$, tj. u skupu S_0 ne nalazi se niti jedna namirnica, pa vrijednost $V(0, m)$ za bilo koji m mora biti jednaka nuli. Analogno, $V(i, 0)$ znači da promatramo sve moguće neprazne podskupove skupa S_i takve da je njihova masa najviše 0 kg. Takvih podskupova očito nema (svaka namirnica ima masu i njezina je vrijednost strogo pozitivan realan broj), pa je pripadna vrijednost funkcije V jednaka 0. Preostala logička ekvivalencija je dogovornoga tipa i poslužit će nam za efektivan izračun vrijednosti funkcije V.

Izvedimo prvu (rekurzivnu) jednakost. Da bismo izračunali vrijednost $V(i, m)$ razmatramo točno dvije, međusobno disjunktne, mogućnosti:

I. namirnica N_i ne pripada niti jednom elementu skupa S_i^m (tj. ako vrijedi nejednakost $m_i > m$);

II. namirnica N_i pripada barem jednom elementu skupa S_i^m (tj. ako vrijedi nejednakost $m_i \leq m$).

[13] Prirodno područje definicije funkcije V obično će biti skup $\mathbf{N}^2$, ali radi izbjegavanja određivanja prirodnoga područja definicije funkcije u svakom pojedinom slučaju ovdje za to područje uzimamo skup $\mathbf{R}$.

[14] Podsjetimo da *svaki* konačan neprazan podskup skupa realnih brojeva ima najveći element.

U prvom slučaju skup S_i^m zapravo tvore svi neprazni podskupovi skupa $S_{i-1} = \{N_1, \ldots, N_{i-1}\}$ čija je masa najviše m kg. Prema definiciji funkcije $V(i, m)$, najveći element skupa koji tvore ukupne vrijednosti tih podskupova jednak je $V(i - 1, m)$.

U drugom slučaju prigodom određivanja najveće među svim vrijednostima elemenata skupa S_i^m dodatno razlikujemo dva podslučaja:

a) Vrijednost namirnice N_i ne ulazi u izračun najveće vrijednosti. Kao i u prvom slučaju, to znači da namirnica N_i ne pripada elementu skupa S_i^m koji ima najveću ukupnu vrijednost, odnosno da je taj element neki podskup skupa $\{N_1, N_2, \ldots, N_{i-1}\}$. Stoga je u ovom podslučaju najveća ukupna vrijednost jednaka $V(i - 1, m)$ jer zbog inkluzije $S_{i-1}^m \subseteq S_i^m$ ne postoji niti jedan element skupa S_{i-1}^m koji ne pripada u skupu S_i^m i čija bi ukupna masa bila veća od uočenoga elementa skupa S_i^m.

b) Vrijednost namirnice N_i ulazi u izračun najveće vrijednosti. To znači da namirnica N_i pripada elementu skupa S_i^m koji ima najveću ukupnu vrijednost. U izračunu te vrijednosti sigurno se pojavljuje pribrojnik c_i koji predstavlja jediničnu vrijednost namirnice N_i. Pogledajmo kako možemo opisati sve preostale pribrojnike. Prema definiciji skupa S_i^m ti su pribrojnici sigurno neke od vrijednosti namirnica $N_1, \ldots, N_{i-1}$. Budući da namirnica N_i ima masu m_i, najveća moguća masa svih ostalih namirnica čije vrijednosti sudjeluju u izračunu najveće vrijednosti iznosi $m - m_i$. Stoga namirnice kojima odgovaraju svi ostali pribrojnici u izračunu najveće vrijednosti tvore neki podskup skupa $S_{i-1} = \{N_1, \ldots, N_{i-1}\}$ čija je ukupna masa najviše $m - m_i$ kg. *Najveća moguća* ukupna vrijednost takvoga podskupa jednaka je $V(i - 1, m - m_i)$, pa najveća ukupna vrijednost u ovom podslučaju iznosi $c_i + V(i - 1, m - m_i)$.

Između dvaju mogućih podslučajeva izabrat ćemo onaj s većom ukupnom vrijednosti, tj.

$$V(i, m) = \max\{V(i - 1, m), c_i + V(i - 1, m - m_i)\},$$

a između dvaju mogućnosti I. i II. ponovno onu koja daje veću vrijednost:

$$V(i, m) = \max\{V(i - 1, m), \{V(i - 1, m), c_i + V(i - 1, m - m_i)\}\}.$$

U skupu na desnoj strani gornje jednakosti pojavljuju se točno tri vrijednosti, od kojih su prva i druga međusobno jednake. Za izračun maksimuma dovoljno je uzeti samo jednu od njih, pa otuda slijedi navedena rekurzivna relacija.

Rješenje Primjera 1. je optimalna vrijednost $V(5, 15)$ s pripadnim optimalnim izborom namirnica. Vrijednost $V(5, 15)$ izračunat ćemo pomoću navedene rekurzivne relacije i pripadnih početnih uvjeta. To ćemo učiniti tako da ćemo za svaki $i \in [5]$ izračunati redom vrijednosti $V(i, 1), V(i, 2), \ldots, V(i, 15)$.

Za $i = 1$ promatramo sve neprazne podskupove skupa $S_1 = \{N_1\}$. Takav je jedan jedini: to je sam skup S_1. Njegova ukupna vrijednost je $c_1 = 4$ €, a ukupna masa $m_1 = 12$ kg. Stoga izravno iz definicije funkcije V (uz uvažavanje $V(0, m) = 0$ za svaki m)) slijedi:

$V(1, 1) = V(1, 2) = \ldots = V(1, 11) = 0$, $V(1, 12) = V(1, 13) = V(1, 14) = V(1, 15) = 4$.

Za $i = 2$ promatramo sve neprazne podskupove skupa $S_2 = \{N_1, N_2\}$. Njih je ukupno 3: $\{N_1\}$, $\{N_2\}$ i $\{N_1, N_2\}$. Vrijednost prvoga podskupa jednaka je $c_1 = 4$ €, a masa $m_1 = 12$ kg. Vrijednost drugoga podskupa jednaka je $c_2 = 2$ €, a masa $m_2 = 1$ kg. Vrijednost trećega podskupa jednaka je $c_1 + c_2 = 4 + 2 = 6$ €, a masa $m_1 + m_2 = 12 + 1 = 13$ kg. Tako lagano dobivamo:

$V(2, 1) = V(2, 2) = \ldots = V(2, 11) = 2$ (u svim slučajevima dobivena optimalna vrijednost 2 se postiže izborom podskupa $\{N_2\}$),
$V(2, 12) = 4$ (biramo između prvoga i drugoga podskupa),
$V(2, 13) = V(2, 14) = V(2, 15) = 6$ (biramo između sva tri podskupa).

Tehniku računanja pomoću rekurzivnih relacija pokazat ćemo na slučaju $i = 3$. Imamo redom:

$V(3, 1) = \max\{V(3 - 1, 1), c_3 + V(3 - 1, 1 - m_3)\} = \max\{V(2, 1), 10 + V(2, 1 - 4)\} =$
$= \max\{V(2, 1), 10 + V(2, -3)\} = \max\{2, 10 + (-\infty)\} = 2;$
$V(3, 2) = \max\{V(3 - 1, 2), c_3 + V(3 - 1, 2 - m_3)\} = \max\{V(2, 2), 10 + V(2, 2 - 4)\} =$
$= \max\{V(2, 2), 10 + V(2, -2)\} = \max\{2, 10 + (-\infty)\} = 2;$
$V(3, 3) = \max\{V(3 - 1, 3), c_3 + V(3 - 1, 3 - m_3)\} = \max\{V(2, 3), 10 + V(2, 3 - 4)\} =$
$= \max\{V(2, 3), 10 + V(2, -1)\} = \max\{2, 10 + (-\infty)\} = 2;$
$V(3, 4) = \max\{V(3 - 1, 4), c_3 + V(3 - 1, 4 - m_3)\} = \max\{V(2, 4), 10 + V(2, 4 - 4)\} =$
$= \max\{V(2, 4), 10 + V(2, 0)\} = \max\{2, 10 + 0\} = 10$ (u naprtnjaču možemo staviti namirnicu N_3 čija je ukupna vrijednost 10 €!);
$V(3, 5) = \max\{V(3 - 1, 5), c_3 + V(3 - 1, 5 - m_3)\} = \max\{V(2, 5), 10 + V(2, 5 - 4)\} =$
$= \max\{V(2, 5), 10 + V(2, 1)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 6) = \max\{V(3 - 1, 6), c_3 + V(3 - 1, 6 - m_3)\} = \max\{V(2, 6), 10 + V(2, 6 - 4)\} =$
$= \max\{V(2, 6), 10 + V(2, 2)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 7) = \max\{V(3 - 1, 7), c_3 + V(3 - 1, 7 - m_3)\} = \max\{V(2, 7), 10 + V(2, 7 - 4)\} =$
$= \max\{V(2, 7), 10 + V(2, 3)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 8) = \max\{V(3 - 1, 8), c_3 + V(3 - 1, 8 - m_3)\} = \max\{V(2, 8), 10 + V(2, 8 - 4)\} =$
$= \max\{V(2, 8), 10 + V(2, 4)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 9) = \max\{V(3 - 1, 9), c_3 + V(3 - 1, 9 - m_3)\} = \max\{V(2, 9), 10 + V(2, 9 - 4)\} =$
$= \max\{V(2, 9), 10 + V(2, 5)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 10) = \max\{V(3 - 1, 10), c_3 + V(3 - 1, 10 - m_3)\} = \max\{V(2, 10), 10 + V(2, 10 - 4)\} =$
$= \max\{V(2, 10), 10 + V(2, 6)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 11) = \max\{V(3 - 1, 11), c_3 + V(3 - 1, 11 - m_3)\} = \max\{V(2, 11), 10 + V(2, 11 - 4)\} =$
$= \max\{V(2, 11), 10 + V(2, 7)\} = \max\{2, 10 + 2\} = 12;$
$V(3, 12) = \max\{V(3 - 1, 12), c_3 + V(3 - 1, 12 - m_3)\} = \max\{V(2, 12), 10 + V(2, 12 - 4)\} =$
$= \max\{V(2, 12), 10 + V(2, 8)\} = \max\{4, 10 + 2\} = 12;$
$V(3, 13) = \max\{V(3 - 1, 13), c_3 + V(3 - 1, 13 - m_3)\} = \max\{V(2, 13), 10 + V(2, 13 - 4)\} =$
$= \max\{V(2, 13), 10 + V(2, 9)\} = \max\{6, 10 + 2\} = 12;$
$V(3, 14) = \max\{V(3 - 1, 14), c_3 + V(3 - 1, 14 - m_3)\} = \max\{V(2, 14), 10 + V(2, 14 - 4)\} =$
$= \max\{V(2, 14), 10 + V(2, 10)\} = \max 6, 10 + 2\} = 12;$
$V(3, 15) = \max\{V(3 - 1, 15), c_3 + V(3 - 1, 15 - m_3)\} = \max\{V(2, 15), 10 + V(2, 15 - 4)\} =$
$= \max\{V(2, 15), 10 + V(2, 11)\} = \max\{6, 10 + 2\} = 12.$

Kako bismo donekle skratili postupak izračunavanja, pogledajmo koje su nam točno vrijednosti funkcije V nužne za izračun vrijednosti $V(5, 15)$:

$V(5, 15) = \max\{V(5 - 1, 15), c_5 + V(5 - 1, 15 - m_5)\} = \max\{V(4, 15), 2 + V(4, 15 - 2)\} =$
$= \max\{V(4, 15), 2 + V(4, 13)\}$

Vrijednosti $V(4, 15)$ i $V(4, 13)$ možemo izračunati jer za odgovarajuće izračune znamo sve potrebne vrijednosti:

$V(4, 15) = \max\{V(4 - 1, 15), c_4 + V(4 - 1, 15 - m_4)\} = \max\{V(3, 15), 1 + V(3, 15 - 1)\} =$
$= \max\{V(3, 15), 1 + V(3, 14)\} = \max\{12, 1 + 12\} = 13;$
$V(4, 13) = \max\{V(4 - 1, 13), c_4 + V(4 - 1, 13 - m_4)\} = \max\{V(3, 13), 1 + V(3, 13 - 1)\} =$
$= \max\{V(3, 13), 1 + V(3, 12)\} = \max\{12, 1 + 12\} = 13.$

Tako je konačno:

$V(5, 15) = \max\{V(4, 15), 2 + V(4, 13)\} = \max\{13, 2 + 13\} = 15.$

Dakle, *optimalna vrijednost svih namirnica stavljenih u naprtnjaču iznosi* 15 €.

Preostaje odrediti pripadni optimalan izbor namirnica, tj. koje točno namirnice treba staviti u naprtnjaču da se dobije gornja optimalna vrijednost. Zaključujemo na sljedeći način:

U svakom je određivanju vrijednosti $V(i, m)$ prvi član unutar vitičaste zagrade dobiven kao posljedica odluke "*vrijednost namirnice N_i ne ulazi u izračun optimalne vrijednosti*", a drugi kao posljedica suprotne odluke "*vrijednost namirnice N_i ulazi u izračun optimalne vrijednosti*". Vrijednost $V(5, 15) = 15$ jednaka je drugom članu unutar pripadne vitičaste zagrade, tj. vrijedi jednakost:

$$V(5, 15) = 2 + 13 = 2 + V(4, 13).$$

To znači da vrijednost namirnice N_5 ulazi u izračun optimalne vrijednosti, pa *namirnicu N_5 stavljamo u naprtnjaču.*

Vrijednost $2 + V(4, 13)$ dobivena je na temelju izračuna vrijednosti $V(4, 13) = 13$ koja je također jednaka drugom članu unutar pripadne vitičaste zagrade, tj. vrijedi jednakost:

$$V(4, 13) = 1 + 12 = 1 + V(3, 12)$$

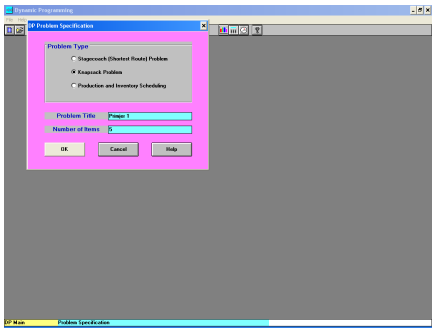
To znači da i vrijednost namirnice N_4 ulazi u izračun optimalne vrijednosti, pa *i namirnicu N_4 stavljamo u naprtnjaču.*

Vrijednost $V(4, 13) = 13$ dobivena je na temelju izračuna vrijednosti $V(3, 12) = 12$, a ova je jednaka drugom članu unutar pripadne vitičaste zagrade. Stoga *i namirnicu N_3 stavljamo u naprtnjaču.*
Nadalje, vrijednost $V(3, 12) = = 12$ dobivena je na temelju vrijednosti $V(2, 8) = 2$, a iz razmatranja navedenih neposredno prije izračuna vrijednosti $V(2, i)$ izravno slijedi da smo vrijednost $V(2, 8)$ dobili na temelju izbora samo namirnice N_2. Prema tome, *namirnicu N_2 stavljamo u naprtnjaču, a namirnicu N_1 ne stavljamo u naprtnjaču.*

Zaključimo: *optimalan izbor namirnica je* $\{N_2, N_3, N_4, N_5\}$, *a pripadna optimalna ukupna vrijednost izabranih namirnica iznosi* 15 €.

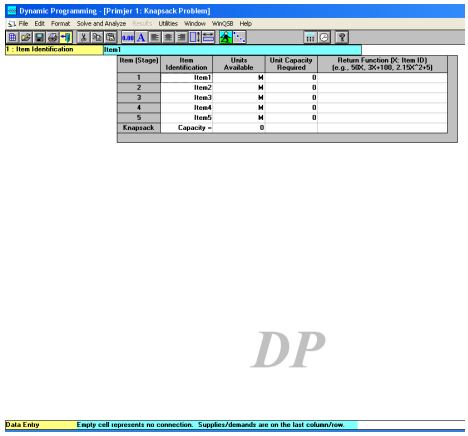
<u>*Rješenje pomoću programa Winqsb*</u>: Pokrenimo potprogram *DP*. Kliknimo lijevom tipkom miša na natpis *File* i odaberimo opciju *New Problem* (slika 9.) Kliknimo lijevom tipkom miša na prazan kružić pored natpisa *Knapsack Problem*, pa u prazne pravokutnike upišimo:

Problem Title (naziv problema): Primjer 1
Number of Items (broj predmeta): 5



Slika 9. Dijaloški okvir potprograma *DP*

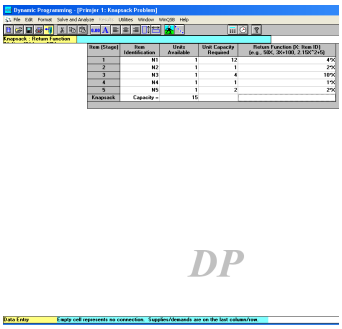
Potom kliknimo lijevom tipkom miša na OK. Pojavljuje se tablica (slika 10.) u koju je potrebno unijeti ulazne podatke.



Slika 10. Tablica za unos ulaznih podataka.

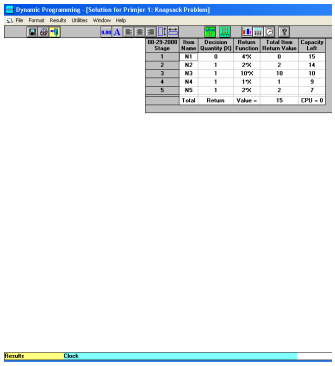
U prvom su stupcu (*Item* (*Stage*)) navedeni redni brojevi predmeta, odnosno, u ovom slučaju, namirnica.
U drugom stupcu (*Item* (*Identification*)) traži se unos naziva svake pojedine namirnice, pa redom upišimo N1 (umjesto *Item*1), N2 (umjesto *Item*2) itd.
U trećem stupcu (*Units Available*) traži se unos raspoložive količine svake pojedine namirnice. Prema pretpostavci, na raspolaganju nam je točno jedan komad svake pojedine vrste namirnica, pa umjesto svakoga od slova *M* u trećem stupcu upišimo 1.
U četvrtom stupcu (*Units Capacity Required*) traži se unos jediničnoga kapaciteta svake pojedine namirnice. U ovome je slučaju riječ o masama namirnica, pa u redak koji odgovara namirnici N_1 upišimo 12, u redak koji odgovara namirnici N_2 upišimo 1, u redak koji odgovara namirnici N_3 upišimo 4, u redak koji odgovara namirnici N_4 upišimo 1, a u redak koji odgovara namirnici N_5 upišimo 2.
U petom stupcu (*Return Function* (*X*: *Item ID*)) traži se unos funkcije cilja koja odgovara svakoj pojedinoj namirnici. Pritom umjesto x_i trebamo upisati jedinstvenu oznaku *X*, a program će sam "prepoznati" na koju se namirnicu odnosi pojedina oznaka *X*. Dakle, u redak koji odgovara namirnici N_1 upišimo 4*X, u redak koji odgovara namirnici N_2 upišimo 2*X, u redak koji odgovara namirnici N_3 upišimo 10*X, u redak koji odgovara namirnici N_4 upišimo 1*X, a u redak koji odgovara namirnici N_5 upišimo 2*X.

Preostaje još popuniti posljednji redak tablice (*Knapsack*) u kojemu se traži unos kapaciteta naprtnjače. U našem je slučaju on jednak 15, pa u polje pored natpisa *Capacity* upišimo 15. Ovime je unos ulaznih podataka završen (slika 11).



Slika 11. Tablica s ulaznim podacima iz Primjera 1.

Na glavnom izborniku kliknimo lijevom tipkom miša na izbornik *Solve and Analyze* i odaberimo opciju *Solve the Problem*. Dobivamo novu tablicu (slika 12.) u kojoj je prikazano rješenje navedenoga problema, pa ćemo ukratko prikazati pojedine njezine dijelove.



Slika 12. Rezultat Primjera 1.

U prvom stupcu nove tablice (*Item Name*) navedeni su nazivi svih namirnica: N_1, N_2, N_3, N_4 i N_5.

U drugom stupcu (*Decision Quantity* (*X*)) navedene su vrijednosti varijabli x_i sukladno našoj definiciji prigodom modeliranja problema. Redom su ispisane vrijednosti 0, 1, 1, 1, 1, što znači da u naprtnjaču trebamo staviti po jedan komad namirnica N_2, N_3, N_4 i N_5, dok namirnicu N_1 ne stavljamo u naprtnjaču. Iz ovoga stupca, dakle, možemo "očitati" optimalan izbor namirnica.

U trećem stupcu (*Return Function*) ispisane su komponente funkcije cilja koje se odnose na svaku pojedinu namirnicu: 4*X, 2*X, 10*X, 1*X i 2*X.

U četvrtom je stupcu (*Total Item Return Value*) za svaku pojedinu vrstu namirnica stavljenu u naprtnjaču ispisana pripadna optimalna vrijednost. Vrijednost namirnice N_1 jednaka je nuli jer nju nismo stavili u naprtnjaču, dok je svaka od ostalih vrijednosti dobivena uvrštavanjem $X = 1$ u odgovarajuću komponentu funkcije cilja. Dakle, vrijednosti namirnica N_2 i N_5 iznose 2 €, vrijednost namirnice N_3 10 €, a vrijednost namirnice N_4 1 €.

U i – tom retku petoga stupca (*Capacity Left*) prikazani su kapaciteti ostatka naprtnjače nakon što u nju stavimo namirnice N_1, N_2, …, N_i. Ovaj stupac možemo shvatiti kao svojevrstan analogon kumulativnom nizu apsolutnih učestalosti "veće od" poznatoga iz opisne (deskriptivne) statistike. Budući da namirnicu N_1 ne stavljamo u naprtnjaču, odgovarajući kapacitet ostatka naprtnjače jednak je polaznom kapacitetu naprtnjače, tj. 15. Namirnicu N_2 stavljamo u naprtnjaču, pa je kapacitet ostatka naprtnjače 15 – 1 = 14 kg. Namirnicu N_3 također stavljamo u naprtnjaču, pa je kapacitet ostatka naprtnjače 14 – 4 = 10 kg itd. Primijetimo da nakon što stavimo posljednju namirnicu N_5 kapacitet ostatka naprtnjače iznosi 7 kg, tj. u naprtnjaču možemo eventualno staviti još neki predmet mase najviše 7 kg.

U posljednjem retku navedena je ukupna vrijednost (*Total Return value*) svih vrsta namirnica stavljenih u naprtnjaču i ona iznosi 15 €.

Primjer 2. Na raspolaganju nam je po jedan komad svakoga od sljedećih 8 predmeta: P_1, P_2, …, P_8. Za svaki su predmet u donjoj tablici navedeni podaci o jediničnoj masi i jediničnoj vrijednosti.

predmet	*jedinična masa* [kg]	*jedinična vrijednost* [000 €]
P_1	1	2
P_2	3	1
P_3	4	3
P_4	3	1
P_5	3	5
P_6	1	4
P_7	5	3
P_8	10	7

Kapacitet naprtnjače iznosi 15 kg. Treba izabrati predmete što veće ukupne vrijednosti tako da njihova ukupna masa ne premaši kapacitet naprtnjače. Formirajmo odgovarajući matematički model, pa odredimo optimalan izbor predmeta i pripadnu optimalnu vrijednost.

Matematički model: Za svaki $i \in [8]$ definirajmo varijablu x_i s:

$$x_i = \begin{cases} 0, & \text{ako predmet } P_i \text{ ne stavljamo u naprtnjaču;} \\ 1, & \text{ako predmet } P_i \text{ stavljamo u naprtnjaču.} \end{cases}$$

Tada odgovarajući *matematički model* glasi:

maksimizirati $f(x_1, x_2, \ldots, x_8) = 2 \cdot x_1 + 1 \cdot x_2 + 3 \cdot x_3 + 1 \cdot x_4 + 5 \cdot x_5 + 4 \cdot x_6 + 3 \cdot x_7 + 7 \cdot x_8$

pod uvjetima

$$1 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 + 3 \cdot x_4 + 3 \cdot x_5 + 1 \cdot x_6 + 5 \cdot x_7 + 10 \cdot x_8 \leq 15$$

$$x_i \in \{0, 1\}, \text{ za svaki } i \in [8].$$

"Klasično" rješenje: Analogno kao i u rješenju Primjera 1., za svaki $i \in [8]$ označimo sa S_i skup koji tvore predmeti P_1, …, P_i uz dodatnu oznaku $S_0 := \varnothing$. Za svaki $i \in [8]$ označimo s c_i jediničnu vrijednost, a m_i jediničnu masu predmeta P_i. Za svaki $m \in [15]$ definiramo:

S_i^m = skup svih nepraznih podskupova skupa S_i takvih da je ukupna masa elemenata svakoga podskupa najviše m kg,

te realnu funkciju dvije realne varijable $V : \mathbf{R}^2 \to \mathbf{R}$:

$$V(i, m) = \max_{S_1 \in S_i^m} \left\{ \sum_j c_j : N_j \in S_1 \right\}.$$

Vrijednosti funkcije V za svaki $i \in [8]$ i svaki $m \in [15] \cup \{0\}$ računamo iz rekurzivne relacije:

$$V(i, m) = \max\{V(i-1, m), c_i + V(i-1, m-m_i)\}$$

uz početne uvjete (koji vrijede za sve dopustive i i m)

$$V(0, m) = V(i, 0) = 0,$$
$$(V(i, m) = -\infty) \Leftrightarrow (m < 0).$$

Postupak računanja je potpuno analogan kao u Primjeru 1., pa u donjoj tablici navodimo samo numeričke rezultate.

i	m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2		0	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3
3		0	2	2	2	3	5	5	5	6	6	6	6	6	6	6	6
4		0	2	2	2	3	5	5	5	6	6	6	7	7	7	7	7
5		0	2	2	2	7	7	7	8	10	10	10	11	11	11	12	12
6		0	4	6	6	7	11	11	11	12	14	14	14	15	15	15	16
7		0	4	6	6	7	11	11	11	12	14	14	14	15	15	17	17
8		0	4	6	6	7	11	11	11	12	14	14	14	15	15	15	**18**

Iz tablice "očitavamo" optimalnu vrijednost svih predmeta koji tvore optimalni izbor: ona iznosi 18 000 €. Preostaje odrediti optimalan izbor predmeta.

Vrijednost $V(8, 15) = 18$ dobivena je kao zbroj $c_8 + V(7, 5)$, tj. donošenjem odluke "*predmet P_8 staviti u naprtnjaču*". Stoga predmet P_8 stavljamo u naprtnjaču.
Vrijednost $V(7, 5) = 11$ dobivena je kao $V(6, 5)$, tj. donošenjem odluke "*predmet P_7 ne staviti u naprtnjaču*". Stoga predmet P_7 ne stavljamo u naprtnjaču.
Vrijednost $V(6, 5) = 11$ dobivena je kao zbroj $c_6 + V(5, 4)$, tj. donošenjem odluke "*predmet P_6 staviti u naprtnjaču*". Stoga predmet P_6 stavljamo u naprtnjaču.
Vrijednost $V(5, 4) = 7$ dobivena je kao zbroj $c_5 + V(4, 1)$, tj. donošenjem odluke "*predmet P_5 staviti u naprtnjaču*". Stoga predmet P_5 stavljamo u naprtnjaču.
Lako je vidjeti da je $V(4, 1) = V(3, 1) = V(2, 1) = V(1, 1)$, što izravno znači da predmete P_4, P_3 i P_2 ne stavljamo u naprtnjaču.
Vrijednost $V(1, 1)$ dobivena je kao $c_1 + V(0, 0)$, tj. donošenjem odluke "*predmet P_1 staviti u naprtnjaču*". Stoga predmet P_1 stavljamo u naprtnjaču.

Dakle, traženi optimalan izbor predmeta je $S = \{P_1, P_5, P_6, P_8\}$.

<u>*Rješenje pomoću programa Winqsb*</u>: Pokrenimo potprogram *DP* i nazovimo problem koji ćemo rješavati *Primjer* 2. Ukupan broj predmeta iznosi 8, pa pored natpisa *Number of Items* upišimo 8. Potom kliknimo na *OK*.

U dobivenu tablicu upisujemo:

- u drugi stupac (*Item* (*Identification*)) redom nazive predmeta: P1, P2, P3 itd.
- u treći stupac (*Units Available*) raspoložive količine svakoga predmeta - budući da raspolažemo s po jednim komadom svake vrste, stupac popunjavamo s jedinicama;
- u četvrti stupac (*Units Capacity Required*) upisujemo redom masu predmeta iz odgovarajućega retka: u prvi redak upišimo 1,u drugi 3, u treći 4 itd.
- u peti stupac (*Return Function* (*X*: *Item ID*)) upišimo komponentu funkcije cilja koja se odnosi na predmet iz odgovarajućega retka: konkretno, u prvi redak upišimo 2*X, u drugi 1*X, u treći 3*X itd.
- u posljednji redak tablice (*Knapsack*) u polje pokraj natpisa *Capacity* upišimo kapacitet naprtnjače: 15.

Time smo unijeli sve potrebne ulazne podatke, pa možemo pokrenuti proceduru *Solve and Analyze* i potproceduru *Solve the Problem*. Dobivamo izlaznu tablicu sa svim potrebnim rezultatima.

Iz drugoga stupca (*Decision Quantity* (*X*)) očitavamo optimalan izbor predmeta kojega tvore predmeti čije su pripadne vrijednosti varijable odlučivanja (x_i) jednake 1. Dakle, u naprtnjaču stavljamo po jedan komad predmeta P_1, P_5, P_6 i P_8. Iz posljednjega polja u četvrtom stupcu očitavamo pripadnu optimalnu vrijednost predmeta koji tvore optimalan izbor: 18 000 € Kapacitet ostatka naprtnjače jednak je 0, što znači da u naprtnjaču više ne možemo staviti niti jedan predmet.

<u>Primjer 3.</u> Riješimo Primjer 2. uz dodatni uvjet da u naprtnjaču obavezno moramo staviti predmet P_2. Masa predmeta P_2 je 3 kg, pa je kapacitet ostatka naprtnjače 15 – 3 = 12 kg. Svakom od preostalih predmeta kao oznaku bijektivno pridružimo točno jedan element skupa [7]:

predmet	*oznaka* (*i*)	*masa* m_i [kg]	*vrijednost* c_i [000 €]
P_1	1	1	2
P_3	2	4	3
P_4	3	3	1
P_5	4	3	5
P_6	5	1	4
P_7	6	5	3
P_8	7	10	7

Uz definiranje varijable x_i s:

$$x_i = \begin{cases} 0, \text{ ako predmet označen brojem } i \text{ ne stavljamo u naprtnjaču;} \\ 1, \text{ ako predmet označen brojem } i \text{ stavljamo u naprtnjaču.} \end{cases}$$

novi <u>*matematički model*</u> glasi:

maksimizirati $f(x_1, x_2, \ldots, x_7) = 2 \cdot x_1 + 3 \cdot x_2 + 1 \cdot x_3 + 5 \cdot x_4 + 4 \cdot x_5 + 3 \cdot x_6 + 7 \cdot x_7$

pod uvjetima

$$1 \cdot x_1 + 4 \cdot x_2 + 3 \cdot x_3 + 3 \cdot x_4 + 1 \cdot x_5 + 5 \cdot x_6 + 10 \cdot x_7 \leq 12$$

$$x_i \in \{0, 1\}, \text{ za svaki } i \in [7].$$

<u>"*Klasično*" *rješenje*</u>: Potpuno analogno kao i u rješenjima prethodnih primjera, za svaki $i \in [7]$ i $m \in [12]$ definiramo skupove S_i i S_i^m, te računamo vrijednosti funkcije $V(i, m)$ koristeći istu rekurzivnu relaciju i iste početne uvjete. Dobiva se sljedeća tablica:

i	*m*	0	1	2	3	4	5	6	7	8	9	10	11	12
0		0	0	0	0	0	0	0	0	0	0	0	0	0
1		0	2	2	2	2	2	2	2	2	2	2	2	2
2		0	2	2	2	3	5	5	5	5	5	5	5	5
3		0	2	2	2	3	5	5	5	6	6	6	6	6
4		0	2	2	5	7	7	7	8	10	10	10	11	11
5		0	4	6	6	9	11	11	11	12	14	14	14	15
6		0	4	6	6	9	11	11	11	12	14	14	14	15
7		0	4	6	6	7	11	11	11	12	14	14	14	**15**

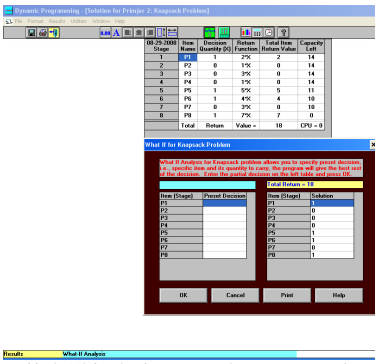
Optimalna vrijednost svih predmeta koji tvore optimalan izbor jednaka je 1 000 (vrijednost predmeta P_2) + 15 000 (optimalna vrijednost svih ostalih predmeta) = 16 000 €. Odredimo sve predmete koji tvore pripadni optimalan izbor (predmet P_2 već smo unaprijed uvrstili u taj izbor).

Očito je $V(7, 12) = V(6, 12) = V(5, 12)$, pa predmete označene brojevima 6 i 7, tj. predmete P_7 i P_8 ne stavljamo u naprtnjaču.
Vrijednost $V(5, 12) = 15$ dobivena je kao zbroj $c_5 + V(4, 11)$, tj. donošenjem odluke "*predmet s oznakom* 5 *staviti u naprtnjaču*", pa predmet P_6 stavljamo u naprtnjaču.
Vrijednost $V(4, 11) = 11$ dobivena je kao zbroj $c_4 + V(3, 8)$, tj. donošenjem odluke "*predmet s oznakom* 4 *staviti u naprtnjaču*", pa predmet P_5 stavljamo u naprtnjaču.
Vrijednost $V(3, 8) = 6$ dobivena je kao zbroj $c_3 + V(2, 5)$, tj. donošenjem odluke "*predmet s oznakom* 3 *staviti u naprtnjaču*", pa predmet P_4 stavljamo u naprtnjaču.
Vrijednost $V(2, 5) = 5$ dobivena je kao zbroj $c_2 + V(1, 1)$, tj. donošenjem odluke "*predmet s oznakom* 2 *staviti u naprtnjaču*", pa predmet P_3 stavljamo u naprtnjaču.
Lako se vidi da je vrijednost $V(1, 1) = 2$ dobivena donošenjem odluke "*predmet s oznakom* 1 *staviti u naprtnjaču*", pa predmet P_1 stavljamo u naprtnjaču.

Uz *iste* oznake iz Primjera 2. slijedi da je optimalan izbor predmeta S_6 (tj. u naprtnjaču treba staviti po jedan komad predmeta $P_1, P_2, \ldots, P_6$), a pripadna optimalna vrijednost 16 000 €.

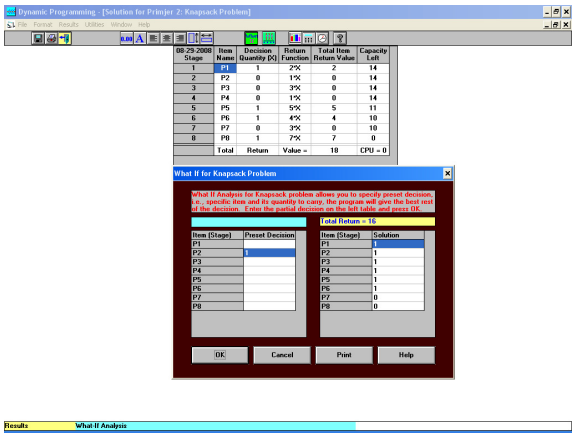
<u>*Rješenje pomoću programa Winqsb*</u>: Iskoristit ćemo rezultat prethodnoga primjera. Potprogram *DP* omogućuje tzv. *što–ako* analizu (*What–If Analysis*) koja rješava isti problem uz dodavanje barem jednoga novoga uvjeta. Ta se procedura nalazi na izborniku *Results* kao opcija *Perform What If Analysis* ili na ikonici *What If* smještenoj na traci s ikonicama.

Pokretanjem navedene procedure dobivamo dijaloški okvir *What If for Knapsack Problem* (slika 12.)



Slika 13. Dijaloški okvir procedure *što – ako* (*What If*).

Na lijevoj strani toga dijaloškoga okvira nalazi se dvostupčana tablica. U prvom su stupcu natpisi predmeta, a u drugom stupcu polja u koja se unose unaprijed definirane vrijednosti varijabli odlučivanja (*Preset Decision*). Budući da se u zadatku zahtijeva da u naprtnjaču obavezno moramo staviti predmet P_2, u drugi stupac trećega retka tablice (polje pored natpisa P2) upišimo 1 i pritisnimo *OK*. Rezultat se pojavljuje u tablici desno (slika 13).



Slika 14. Rezultat primjene procedure *što – ako* (*What If*).

Varijabla odlučivanja ima vrijednost 0 za namirnice P_7 i P_8. Stoga te predmete u ovom slučaju ne stavljamo u naprtnjaču, a svaki od preostalih 6 predmeta stavljamo u naprtnjaču. Time se ukupna vrijednost svih predmeta stavljenih u naprtnjaču s 18 000 € smanjila na 16 000 €. Podatak o kapacitetu ostatka naprtnjače nije naveden, ali ga nije teško izračunati: ukupna masa prvih šest predmeta jednaka je 15 kg, pa je naprtnjača opet potpuno ispunjena.

Istaknimo da potprogram *DP ne* dozvoljava mogućnost *što–ako* analize u slučajevima kada se definira koji predmet *nećemo* staviti u naprtnjaču. To je i logično jer je prirodna polazna pretpostavka da svaki od navedenih predmeta želimo staviti u naprtnjaču.

U nastavku na primjerima ilustriramo postupak rješavanja otvorenoga, odnosno zatvorenoga problema naprtnjače. Pritom napomenimo da se, prigodom rješavanja pomoću potprograma *DP*, otvoreni problem naprtnjače obavezno mora svesti na njemu *izomorfni* zatvoreni problem naprtnjače jer postavke potprograma *DP* (stupac *Units Available*) zahtijevaju nužno definiranje količine (raspoloživi broj komada) svakoga pojedinoga predmeta.

Primjer 4. Na raspolaganju nam je proizvolja količina svake od četiriju različitih vrsta namirnica: N_1, N_2, N_3 i N_4, te naprtnjača kapaciteta 25 kg. Podaci o jediničnoj masi i jediničnoj cijeni svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*jedinična cijena* [USD]	*jedinična masa* [kg]
N_1	11	6
N_2	7	4
N_3	5	3
N_4	1	1

Treba odabrati namirnice što veće ukupne vrijednost tako da njihova ukupna masa ne premaši kapacitet naprtnjače. Formirajmo odgovarajući matematički model, pa odredimo optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.

Budući da nam je na raspolaganju proizvoljna količina svake vrste namirnica, riječ je o *otvorenom problemu naprtnjače*. Za svaki $i \in [4]$ označimo s x_i količinu (broj komada) namirnice N_i koje ćemo staviti u naprtnjaču. Budući da niti jedan komad niti jedne namirnice ne možemo podijeliti na manje dijelove, vrijednosti varijable x_i pripadaju skupu $\mathbf{N}_0$. Stoga je odgovarajući *matematički model*:

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 11 \cdot x_1 + 7 \cdot x_2 + 5 \cdot x_3 + x_4$$

pod uvjetima

$$6 \cdot x_1 + 4 \cdot x_2 + 3 \cdot x_3 + x_4 \leq 25$$

$$x_i \in \mathbf{N}_0, \text{ za svaki } i \in [4]$$

"*Klasično" rješenje*: Za svaki $i \in [4]$ označimo s c_i jediničnu cijenu, a s m_i jediničnu masu namirnice N_i. Definirajmo realnu funkciju $F : \mathbf{R} \to \mathbf{R}$ rekurzivno s:

$$F(s) = \max_i \left[c_i + f(s - m_i) \right]$$

uz početne uvjete

$$F(0) = 0$$
$$F(s) = -\infty, \text{ za svaki } s < 0.$$

Vrijednosti funkcije $F(s)$ su *optimalne ukupne vrijednosti namirnica čija ukupna masa nije veća od s*. Te se vrijednosti za fiksiranu vrijednost broja s određuju primjenom Bellmanova načela: ako u naprtnjaču želimo staviti namirnice ukupne mase najviše s kg i ako najprije stavimo *točno jedan* komad namirnice N_i čija je masa m_i, onda ćemo moći staviti još najviše $s - m_i$ kg namirnica (zbog proizvoljnosti broja komada svake vrste namirnica tu uključujemo i namirnicu N_i čiji smo jedan komad već stavili u naprtnjaču). Jedinična vrijednost namirnice N_i iznosi c_i USD, a *optimalna ukupna vrijednost* svih ostalih komada namirnica, prema definiciji funkcije F, iznosi $F(s - m_i)$. Stoga ukupna vrijednost u slučaju stavljanja jednoga komada namirnice N_i iznosi $c_i + F(s - m_i)$ USD. Budući da namirnicu N_i možemo birati između N_1, N_2, N_3 i N_4, optimalnu ukupnu vrijednost svih namirnica čija je ukupna masa najviše s kg dobit

ćemo uzimanjem najveće od četiriju mogućih vrijednosti navedenoga zbroja (među tim vrijednostima mogu se pojaviti i međusobno jednake).

Uvjet $F(0) = 0$ je prirodna posljedica činjenice da je optimalna ukupna vrijednost namirnica čija ukupna masa nije veća od 0 kg jednaka nuli jer u tom slučaju u naprtnjaču ne stavljamo niti jednu namirnicu.

Uvjet $F(s) = -\infty$ je dogovornoga tipa i poslužit će nam za efektivne izračune vrijednosti funkcije $F(s)$.

Tražena optimalna vrijednost očito je jednaka vrijednosti $F(25)$. Stoga za svaki $s \in [25]$ treba izračunati vrijednosti $F(s)$. Detaljan izračun ilustrativno navodimo za $s = 1, 2, \ldots, 6$:

$F(1) = \max_i \left[c_i + F(1-m_i)\right] = \max\{c_1 + F(1 - m_1), c_2 + F(1 - m_2), c_3 + F(1 - m_3), c_4 + F(1 - m_4)\} = \max\{11 + F(1 - 6), 7 + F(1 - 4), 5 + F(1 - 3), 1 + F(1 - 1)\} = \max\{11 + F(-5), 7 + F(-3), 5 + F(-2), 1 + F(0)\} = \max\{11 + (-\infty), 7 + (-\infty), 5 + (-\infty), 1 + 0\} = 1;$
$F(2) = \max_i \left[c_i + F(2-m_i)\right] = \max\{c_1 + F(2 - m_1), c_2 + F(2 - m_2), c_3 + F(2 - m_3), c_4 + F(2 - m_4)\} = \max\{11 + F(2 - 6), 7 + F(2 - 4), 5 + F(2 - 3), 1 + F(2 - 1)\} = \max\{11 + F(-4), 7 + F(-2), 5 + F(-1), 1 + F(1)\} = \max\{11 + (-\infty), 7 + (-\infty), 5 + (-\infty), 1 + 1\} = 2;$
$F(3) = \max_i \left[c_i + F(3-m_i)\right] = \max\{c_1 + F(3 - m_1), c_2 + F(3 - m_2), c_3 + F(3 - m_3), c_4 + F(3 - m_4)\} = \max\{11 + F(3 - 6), 7 + F(3 - 4), 5 + F(3 - 3), 1 + F(3 - 1)\} = \max\{11 + F(-3), 7 + F(-1), 5 + F(0), 1 + F(2)\} = \max\{11 + (-\infty), 7 + (-\infty), 5 + 0, 1 + 2\} = 5;$
$F(4) = \max_i \left[c_i + F(4-m_i)\right] = \max\{c_1 + F(4 - m_1), c_2 + F(4 - m_2), c_3 + F(4 - m_3), c_4 + F(4 - m_4)\} = \max\{11 + F(4 - 6), 7 + F(4 - 4), 5 + F(4 - 3), 1 + F(4 - 1)\} = \max\{11 + F(-2), 7 + f(0), 5 + F(1), 1 + F(3)\} = \max\{11 + (-\infty), 7 + 0, 5 + 1, 1 + 5\} = 7;$
$F(5) = \max_i \left[c_i + F(5-m_i)\right] = \max\{c_1 + F(5 - m_1), c_2 + F(5 - m_2), c_3 + F(5 - m_3), c_4 + F(5 - m_4)\} = \max\{11 + F(5 - 6), 7 + F(5 - 4), 5 + F(5 - 3), 1 + F(5 - 1)\} = \max\{11 + F(-1), 7 + F(1), 5 + F(2), 1 + F(4)\} = \max\{11 + (-\infty), 7 + 1, 5 + 2, 1 + 7\} = 8;$
$F(6) = \max_i \left[c_i + F(6-m_i)\right] = \max\{c_1 + F(6 - m_1), c_2 + F(6 - m_2), c_3 + F(6 - m_3), c_4 + F(6 - m_4)\} = \max\{11 + F(6 - 6), 7 + F(6 - 4), 5 + F(6 - 3), 1 + F(6 - 1)\} = \max\{11 + F(0), 7 + f(2), 5 + F(3), 1 + F(5)\} = \max\{11 + 0, 7 + 2, 5 + 5, 1 + 8\} = 11$ itd.

Sve vrijednosti funkcije $F(s)$ navedene su u donjoj tablici.

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$F(s)$	0	1	2	5	7	8	11	12	14	16	18	19	22	23	25	27	29	30	33	34

s	20	21	22	23	24	25
$F(s)$	36	38	40	41	44	45

Dakle, $F(25) = 45$, što znači da *optimalna ukupna vrijednost namirnica iznosi* 45 USD. Pripadni optimalan izbor namirnica određujemo slično kao i u prethodnim primjerima.

$F(25) = 45$ možemo dobiti na čak četiri različita načina: ili kao zbroj $c_1 + F(19)$ ili kao zbroj $c_2 + F(21)$ ili kao zbroj $c_3 + F(22)$ ili kao zbroj $c_4 + F(24)$. Stoga imamo četiri različite

mogućnosti izbora, pa se opredijelimo npr. za posljednju od njih, tj. u naprtnjaču stavimo jedan komad namirnice N_4. (Za vježbu razmotrite svaku od ostalih triju mogućnosti.)
$F(24) = 44$ dobiva se kao zbroj $c_1 + F(18)$, pa u naprtnjaču stavljamo jedan komad namirnice N_1.
$F(18) = 33$ dobiva se kao zbroj $c_1 + F(12)$, pa u naprtnjaču stavljamo još jedan komad namirnice N_1.
$F(12) = 22$ dobiva se kao zbroj $c_1 + F(6)$, pa u naprtnjaču stavljamo još jedan komad namirnice N_1.
$F(6) = 11$ dobiva se kao zbroj $c_1 + F(0)$, pa u naprtnjaču stavljamo još jedan komad namirnice N_1.

Dakle, optimalan izbor namirnica tvore 4 komada namirnice N_1 i 1 komad namirnice N_4.

<u>*Rješenje pomoću programa Winqsb*</u>: Budući da program *Winqsb* zahtijeva zadavanje najveće dozvoljene količine namirnica, otvoreni problem naprtnjače svodimo na zatvoreni problem naprtnjače (vidjeti Primjer 5.). Zbog uvjeta

$$x_i \in \mathbf{N}_0, \text{ za svaki } i \in [4],$$

uočimo da iz uvjeta

$$6 \cdot x_1 + 4 \cdot x_2 + 3 \cdot x_3 + x_4 \leq 25$$

slijedi da vrijednost varijable x_1 može biti najviše jednaka $\left\lfloor \frac{25}{6} \right\rfloor = 4$,[15] vrijednost varijable x_2 najviše jednaka $\left\lfloor \frac{25}{4} \right\rfloor = 6$, vrijednost varijable x_3 najviše jednaka $\left\lfloor \frac{25}{3} \right\rfloor = 8$, a vrijednost varijable x_4 najviše jednaka 25. Pokrenimo potprogram *DP*, nazovimo primjer koji rješavamo *Primjer* 4, a kao ukupan broj različitih vrsta namirnica upišimo 4. Potom u dobivenoj tablici upišimo redom:

- u drugi stupac (*Item* (*Identification*)) redom nazive namirnica: N1, N2, N3, N4;
- u treći stupac (*Units Available*) redom upravo izračunate najveće *dopustive* količine: u prvi redak upišimo 4, u drugi 6, u treći 8, a u četvrti 25;
- u četvrti stupac (*Units Capacity Required*) redom jediničnu masu namirnice iz odgovarajućega retka: u prvi redak upišimo 6, u drugi 4, u treći 3, a u četvrti 1;
- u peti stupac (*Return Function* (*X*: *Item ID*)) upišimo komponentu funkcije cilja koja se odnosi na namirnicu iz odgovarajućega retka: u prvi redak upišimo 11*X, u drugi 7*X, u treći 5*X, a u četvrti 1*X.
- u posljednji redak tablice (*Knapsack*) u polje pokraj natpisa *Capacity* upišimo kapacitet naprtnjače: 25.

Time smo unijeli sve potrebne ulazne podatke, pa možemo pokrenuti proceduru *Solve and Analyze* i potproceduru *Solve the Problem*. Dobivamo izlaznu tablicu sa svim potrebnim rezultatima.

[15] S $\lfloor x \rfloor$ označavamo funkciju koja bilo kojem realnom broju x pridružuje najveći cijeli broj koji je manji ili jednak broju x.

U drugom stupcu (*Decision Quantity* (*X*)) navedeni su brojevi komada svake pojedine namirnice, pa "očitavamo" optimalan izbor namirnica: 3 komada namirnice N_3, te po jedan komad svake od namirnica N_2 i N_3. Pripadna optimalna ukupna vrijednost iznosi 45 USD.

Primijetimo da smo na opisani način dobili optimalan izbor različit od izbora dobivena "klasičnim" načinom, ali s jednakom optimalnom ukupnom vrijednošću. Kako smo i ranije napomenuli, potprogram *DP* ne ispisuje sve optimalne izbore, već točno jedan od njih.

Primjer 5. Na raspolaganju su nam po točno 2 komada namirnica N_1 i N_2, točno 3 komada namirnice N_3 i naprtnjača kapaciteta 10 kg. Podaci o jediničnoj masi i jediničnoj cijeni svake namirnice navedeni su u sljedećoj tablici.

namirnica	*jedinična masa* [kg]	*jedinična cijena* [kn]
N_1	2.5	48.00
N_2	1.5	51.00
N_3	2	45.00

Treba odrediti namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne premaši kapacitet naprtnjače. Formirajmo odgovarajući matematički model, pa odredimo optimalan izbor namirnica, pripadnu optimalnu ukupnu vrijednost, te hoćemo li dobivenim optimalnim izborom namirnica u potpunosti ispuniti naprtnjaču.

Matematički model: Zbog ograničenosti raspoloživih količina namirnica riječ je o *zatvorenu problemu naprtnjače*. Za svaki $i \in [3]$ definiramo varijablu x_i kao količinu namirnice i koju stavljamo u naprtnjaču. Tada je traženi *matematički model*:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 48 \cdot x_1 + 51 \cdot x_2 + 45 \cdot x_3$$

pod uvjetima

$$2.5 \cdot x_1 + 1.5 \cdot x_2 + 2 \cdot x_3 \leq 10$$
$$0 \leq x_1 \leq 2;\ 0 \leq x_2 \leq 2;\ 0 \leq x_3 \leq 3$$
$$x_1, x_2, x_3 \in \mathbf{Z}.$$

"*Klasično*" *rješenje*: Opisat ćemo način rješavanja koji ćemo primijeniti i u točki 1.5., tj. na rješavanje problema složene razdiobe ulaganja. Osnovna ideja je sljedeća:

U *prvoj fazi* razmatramo slučaj kad u naprtnjaču možemo staviti samo namirnicu N_1, u *drugoj fazi* slučaj kad u naprtnjaču možemo staviti samo namirnice N_1 i N_2, a u posljednjoj (*trećoj*) *fazi* slučaj kad u naprtnjaču žemo staviti sve tri namirnice. Odmah istaknimo da ukoliko na raspolaganju imamo k različitih vrsta namirnica: $N_1, \ldots, N_k$, onda imamo ukupno k faza rješavanja problema, pri čemu, za svak $i \in [k]$, u i – toj fazi razmatramo slučaj kad u naprtnjaču možemo staviti samo prvih i vrsta namirnica, tj. namirnice $N_1, \ldots, N_i$.

Primijetimo i da je, za svaki $i = 1, 2, 3$, dopustiva odluka "*u naprtnjaču staviti svu količinu namirnice* N_i". Ekvivalentno, jedna dopustiva odluka je "*u naprtnjaču staviti svu količinu namirnice* N_1" (pritom zanemarujemo eventualno stavljene količine ostalih namirnica), druga dozvoljena odluka je "*u naprtnjaču staviti svu količinu namirnice* N_2" itd. U praksi se, međutim, može dogoditi da svu količinu neke namirnice ne možemo odjednom staviti u

naprtnjaču. Npr. imamo li ukupno 10 komada namirnice N_1 čija je ukupna masa $10 \cdot 2.5 = 25$ kg, svih 10 komada ne možemo staviti u naprtnjaču kapaciteta 10 kg. U takvom se slučaju u matematičkom modelu nužno mora *predefinirati gornja granica* raspoložive količine dotične namirnice tako da se za gornju granicu uzme najveći cijeli broj manji ili jednak omjeru kapaciteta naprtnjače i jedinične mase (obujma i sl.) namirnice. To je i logično jer ako imamo npr. namirnicu mase 2.5 kg i naprtnjaču kapaciteta 10 kg, onda u tu naprtnjaču stane najviše $\left\lfloor \frac{10}{2.5} \right\rfloor = 4$ komada te namirnice.

Kako smo rekli, u prvoj fazi rješavanja problema razmatramo slučaj kad u naprtnjaču kapaciteta s možemo staviti samo raspoložive komade namirnice N_1. Za svaki $s \in \{0.5 \cdot k : k \in [20]\}$ računamo vrijednosti realne funkcije f_1 definirane s

$$f_1(s) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{s}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1),$$

Pojasnimo zadavanje funkcije f_1. Varijabla x_1 navedena u uvjetima prema kojima određujemo maksimum zapravo "mjeri" količinu namirnice N_1 stavljenu u naprtnjaču kapaciteta s. Budući da jedinična masa te namirnice iznosi 2.5 kg, u naprtnjaču možemo staviti najviše $\frac{s}{2.5}$ komada namirnice N_1. Prema uvjetu zadatka, neovisno o kapacitetu naprtnjače možemo staviti najviše 2 komada namirnice N_1 jer upravo toliko komada te namirnice imamo na raspolaganju. Stoga je najveća moguća količina namirnice N_1 koju možemo staviti u naprtnjaču kapaciteta s jednaka manjem od brojeva $\frac{s}{2.5}$ i 2, tj. $\min\{\frac{s}{2.5}, 2\}$. Dodatni uvjet $x_1 \in \mathbf{Z}$ izriče činjenicu da su vrijednosti varijable x_1 cijeli brojevi, što je izravna posljedica pretpostavke da niti jedan komad namirnice N_1 ne smijemo dijeliti na manje dijelove. Napokon, vrijednost x_1 komada namirnice N_1 iznosi $48 \cdot x_1$ kn, pa su vrijednosti funkcije f_1, zapravo, *optimalne ukupne vrijednosti* dobivene ukoliko se u naprtnjaču kapaciteta s stavljaju isključivo komadi namirnice N_1.

Budući da su zadane jedinične mase namirnica višekratnici broja 0.5, vrijednosti funkcije f računamo za sve višekratnike toga broja koji nisu veći od kapaciteta naprtnjače, tj. 10. Tako redom imamo:

$$f_1(0) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{0}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le \min\{0,2\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 0 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = 48 \cdot 0 = 0;$$

$$f_1(0.5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{0.5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 0.2 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = 48 \cdot 0 = 0;$$

$$f_1(1) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{1}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 0.4 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = 48 \cdot 0 = 0;$$

$$f_1(1.5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{1.5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 0.6 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = 48 \cdot 0 = 0;$$

$$f_1(2) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{2}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 0.8 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = 48 \cdot 0 = 0;$$

$$f_1(2.5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{2.5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 1 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1\} = 48.$$

$$f_1(3) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{3}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 1.2 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1\} = 48;$$

$$f_1(3.5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{3.5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 1.4 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1\} = 48;$$

$$f_1(4) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{4}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 1.6 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1\} = 48;$$

$$f_1(4.5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{4.5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 1.8 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1\} = 48;$$

$$f_1(5) = \max_{\substack{0 \le x_1 \le \min\left\{\frac{5}{2.5}, 2\right\} \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max_{\substack{0 \le x_1 \le 2 \\ x_1 \in \mathbb{Z}}} (48 \cdot x_1) = \max\{48 \cdot 0, 48 \cdot 1, 48 \cdot 2\} = 96.$$

Budući da za svaki $s \in \{5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\}$ vrijedi jednakost

$$\min\{\frac{s}{2.5}, 2\} = 2,$$

odatle slijedi da za svaki $s \in \{5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\}$ vrijedi jednakost

$$f_1(s) = f_1(5) = 96.$$

U drugoj fazi rješavanja razmatramo slučaj kad u naprtnjaču kapaciteta s možemo staviti samo raspoložive vrijednosti namirnica N_1 i N_2. Razmišljamo na sljedeći način:

Pretpostavimo da u naprtnjaču kapaciteta s stavljamo x_2 komada namirnice N_2. Analogno kao i u prvoj fazi zaključujemo da je najveća moguća vrijednost varijable x_2 manji od brojeva $\frac{s}{1.5}$ i 2, tj. $\min\{\frac{s}{1.5}, 2\}$. Ukupna masa x_2 komada namirnice N_2 iznosi $1.5 \cdot x_2$ kg, pa u naprtnjaču možemo staviti još najviše $s - 1.5 \cdot x_2$ kg namirnice N_1. *Optimalna* vrijednost $s - 1.5 \cdot x_2$ namirnice N_1, prema definiciji funkcije f_1, jednaka je $f_1(s - 1.5 \cdot x_2)$, pa ukupna vrijednost svih stavljenih namirnica iznosi $51 \cdot x_2 + f_1(s - 1.5 \cdot x_2)$ kn. Definiramo realnu funkciju f_2 s:

$$f_2(s) = \max_{\substack{0 \le x_2 \le \min\left\{\frac{s}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(s - 1.5 \cdot x_2)\right].$$

Vrijednosti funkcije f_2 su *optimalne ukupne vrijednosti* namirnica N_1 i N_2 stavljenih u naprtnjaču kapaciteta s. I vrijednosti ove funkcije računamo za svaki $s \in \{0.5 \cdot k : k \in [20]\}$, pa redom dobivamo:

$f_2(0) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{0}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(0 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 0 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(0 - 1.5 \cdot x_2)\right] = 51 \cdot 0 + f_1(0 - 1.5 \cdot 0) = 0 + f_1(0) = 0 + 0 = 0;$

$f_2(0.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{0.5}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(0.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le \frac{1}{3} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(0.5 - 1.5 \cdot x_2)\right] = 51 \cdot 0 + f_1(0.5 - 1.5 \cdot 0) = 0 + f_1(0.5) = 0 + 0 = 0;$

$f_2(1) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{1}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(1 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le \frac{2}{3} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(1 - 1.5 \cdot x_2)\right] = 51 \cdot 0 + f_1(1 - 1.5 \cdot 0) = 0 + f_1(1) = 0 + 0 = 0;$

$f_2(1.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{1.5}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(1.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 1 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(1.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(1.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(1.5 - 1.5 \cdot 1)\} = \max\{0 + f_1(1.5), 51 + f_1(0)\} = \max\{0 + 0, 51 + 0\} = 51;$

$f_2(2) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{2}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(2 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le \frac{4}{3} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(2 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(2 - 1.5 \cdot 0), 51 \cdot 1 + f_1(2 - 1.5 \cdot 1)\} = \max\{0 + f_1(2), 51 + f_1(0.5)\} = \max\{0 + 0, 51 + 0\} = 51;$

$f_2(2.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{2.5}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(2.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le \frac{5}{3} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(2.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(2.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(2.5 - 1.5 \cdot 1)\} = \max\{0 + f_1(2.5), 51 + f_1(1)\} = \max\{0 + 48, 51 + 0\} = 51;$

$f_2(3) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{3}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(3 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(3 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(3 - 1.5 \cdot 0), 51 \cdot 1 + f_1(3 - 1.5 \cdot 1), 51 \cdot 2 + f_1(3 - 1.5 \cdot 2)\} = \max\{0 + f_1(3), 51 + f_1(1.5), 102 + f_1(0)\} = \max\{0 + 48, 51 + 0, 102 + 0\} = 102;$

$f_2(3.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{3.5}{1.5}, 2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(3.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(3.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(3.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(3.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(3.5 - 1.5 \cdot 2)\} = \max\{0 + f_1(3.5), 51 + f_1(2), 102 + f_1(0.5)\} = \max\{0 + 48, 51 + 0, 102 + 0\} = 102;$

$f_2(4) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{4}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(4 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(4 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(4 - 1.5 \cdot 0), 51 \cdot 1 + f_1(4 - 1.5 \cdot 1), 51 \cdot 2 + f_1(4 - 1.5 \cdot 2)\} = \max\{0 + f_1(4), 51 + f_1(2.5), 102 + f_1(1)\} = \max\{0 + 48, 51 + 48, 102 + 0\} = 102;$

$f_2(4.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{4.5}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(4.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(4.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(4.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(4.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(4.5 - 1.5 \cdot 2)\} = \max\{0 + f_1(4.5), 51 + f_1(3), 102 + f_1(1)\} = \max\{0 + 48, 51 + 48, 102 + 0\} = 102;$

$f_2(5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{5}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(5 - 1.5 \cdot 2)\} = \max\{0 + f_1(5), 51 + f_1(3.5), 102 + f_1(2)\} = \max\{0 + 96, 51 + 48, 102 + 0\} = 102;$

$f_2(5.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{5.5}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(5.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(5.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(5.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(5.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(5.5 - 1.5 \cdot 2)\} = \max\{0 + f_1(5.5), 51 + f_1(4), 102 + f_1(2.5)\} = \max\{0 + 96, 51 + 48, 102 + 48\} = 150;$

$f_2(6) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{6}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(6 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(6 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(6 - 1.5 \cdot 0), 51 \cdot 1 + f_1(6 - 1.5 \cdot 1), 51 \cdot 2 + f_1(6 - 1.5 \cdot 2)\} = \max\{0 + f_1(6), 51 + f_1(4.5), 102 + f_1(3)\} = \max\{0 + 96, 51 + 48, 102 + 48\} = 150;$

$f_2(6.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{6.5}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(6.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(6.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(6.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(6.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(6.5 - 1.5 \cdot 2)\} = \max\{0 + f_1(6.5), 51 + f_1(5), 102 + f_1(3.5)\} = \max\{0 + 96, 51 + 96, 102 + 48\} = 150;$

$f_2(7) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{7}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(7 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(7 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(7 - 1.5 \cdot 0), 51 \cdot 1 + f_1(7 - 1.5 \cdot 1), 51 \cdot 2 + f_1(7 - 1.5 \cdot 2)\} = \max\{0 + f_1(7), 51 + f_1(5.5), 102 + f_1(4)\} = \max\{0 + 96, 51 + 96, 102 + 48\} = 150;$

$f_2(7.5) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{7.5}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(7.5 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(7.5 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(7.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(7.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(7.5 - 1.5 \cdot 2)\} = \max\{0 + f_1(7.5), 51 + f_1(6), 102 + f_1(4.5)\} = \max\{0 + 96, 51 + 96, 102 + 48\} = 150;$

$f_2(8) = \max\limits_{\substack{0 \le x_2 \le \min\left\{\frac{8}{1.5},2\right\} \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(8 - 1.5 \cdot x_2)\right] = \max\limits_{\substack{0 \le x_2 \le 2 \\ x_2 \in \mathbb{Z}}} \left[51 \cdot x_2 + f_1(8 - 1.5 \cdot x_2)\right] = \max\{51 \cdot 0 + f_1(8 - 1.5 \cdot 0), 51 \cdot 1 + f_1(8 - 1.5 \cdot 1), 51 \cdot 2 + f_1(8 - 1.5 \cdot 2)\} = \max\{0 + f_1(8), 51 + f_1(6.5), 102 + f_1(5)\} = \max\{0 + 96, 51 + 96, 102 + 96\} = 198;$

$f_2(8.5) = \max\limits_{\substack{0\le x_2\le \min\left\{\frac{8}{1.5},2\right\}\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(8-1.5\cdot x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(8-1.5\cdot x_2)\right] = \max\ \{51 \cdot 0 +$
$+ f_1(8.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(8.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(8.5 - 1.5 \cdot 2)\} = \max\ \{0 + f_1(8.5), 51 +$
$+ f_1(7), 102 + f_1(5.5)\} = \max\ \{0 + 96, 51 + 96, 102 + 96\} = 198;$

$f_2(9) = \max\limits_{\substack{0\le x_2\le \min\left\{\frac{9}{1.5},2\right\}\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(9-1.5\cdot x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(9-1.5\cdot x_2)\right] = \max\ \{51 \cdot 0 +$
$+ f_1(9 - 1.5 \cdot 0), 51 \cdot 1 + f_1(9 - 1.5 \cdot 1), 51 \cdot 2 + f_1(9 - 1.5 \cdot 2)\} = \max\ \{0 + f_1(9), 51 + f_1(7.5),$
$102 + f_1(6)\} = \max\ \{0 + 96, 51 + 96, 102 + 96\} = 198;$

$f_2(9.5) = \max\limits_{\substack{0\le x_2\le \min\left\{\frac{9.5}{1.5},2\right\}\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(9.5-1.5\cdot x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(9.5-1.5\cdot x_2)\right] = \max\ \{51 \cdot 0 +$
$+ f_1(9.5 - 1.5 \cdot 0), 51 \cdot 1 + f_1(9.5 - 1.5 \cdot 1), 51 \cdot 2 + f_1(9.5 - 1.5 \cdot 2)\} = \max\ \{0 + f_1(9.5), 51 +$
$+ f_1(8), 102 + f_1(6)\} = \max\ \{0 + 96, 51 + 96, 102 + 96\} = 198;$

$f_2(10) = \max\limits_{\substack{0\le x_2\le \min\left\{\frac{10}{1.5},2\right\}\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(10-1.5\cdot x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[51\cdot x_2 + f_1(10-1.5\cdot x_2)\right] = \max\ \{51 \cdot 0 +$
$+ f_1(10 - 1.5 \cdot 0), 51 \cdot 1 + f_1(10 - 1.5 \cdot 1), 51 \cdot 2 + f_1(10 - 1.5 \cdot 2)\} = \max\ \{0 + f_1(10), 51 +$
$+ f_1(8.5), 102 + f_1(6.5)\} = \max\ \{0 + 96, 51 + 96, 102 + 96\} = 198.$

Napokon, u trećoj fazi razmatramo slučaj kad u naprtnjaču kapaciteta s stavljamo sve tri raspoložive vrste namirnica. Analogno kao i u drugoj fazi razmišljamo na sljedeći način:

Pretpostavimo da smo u naprtnjaču kapaciteta s stavili ukupno x_3 komada namirnice N_3. Najveća moguća vrijednost varijable x_3 je manji od brojeva $\frac{s}{2}$ i 3, tj. $\min\{\frac{s}{2}, 3\}$. Ukupna masa x_3 komada namirnice N_3 je $2 \cdot x_3$ kg, pa u naprtnjaču možemo staviti još najviše $s - 2 \cdot x_3$ kg namirnica N_1 i N_2. Prema definiciji funkcije f_2, *optimalna vrijednost* $s - 2 \cdot x_3$ kg namirnica N_1 i N_2 iznosi $f_2(s - 2 \cdot x_3)$, pa ukupna vrijednost svih stavljenih namirnica iznosi $45 \cdot x_3 + f_2(s - 2 \cdot x_3)$. Definiramo realnu funkciju

$$f_3(s) = \max\limits_{\substack{0\le x_3\le \min\left\{\frac{s}{2},2\right\}\\ x_3\in\mathbb{Z}}} \left[45\cdot x_3 + f_2(s-2\cdot x_3)\right]$$

čije su vrijednosti *optimalne ukupne vrijednosti* svih triju vrsta namirnica stavljenih u naprtnjaču kapaciteta s. Optimalna vrijednost koja je rješenje postavljenoga problema jednaka je $f_3(10)$, pa je:

$f_3(10) = \max\limits_{\substack{0\le x_3\le \min\left\{\frac{10}{2},2\right\}\\ x_3\in\mathbb{Z}}} \left[45\cdot x_3 + f_2(10-2\cdot x_3)\right] = \max\limits_{\substack{0\le x_3\le 2\\ x_3\in\mathbb{Z}}} \left[45\cdot x_3 + f_2(10-2\cdot x_3)\right] = \max\ \{45 \cdot 0 + f_2(10$
$- 2 \cdot 0), 45 \cdot 1 + f_2(10 - 2 \cdot 1), 45 \cdot 2 + f_2(10 - 2 \cdot 2)\} = \max\ \{0 + f_2(10), 45 + f_2(8), 90 + f_2(6)\}$
$= \max\ \{0 + 198, 45 + 198, 90 + 150\} = 243.$

(Za vježbu izračunajte vrijednosti funkcije f_3 za svaki $s \in \{0.5 \cdot k : k \in [19]\}$.)

Time smo izračunali pripadnu optimalnu vrijednost namirnica koje tvore optimalan izbor i ona iznosi 243 kn. Pogledajmo sada kako iz dobivenih rezultata možemo "očitati" pripadni optimalan izbor namirnica.

Vrijednost $f_3(10) = 243$ dobivena je kao zbroj 45 + 198, odnosno kao zbroj $45 + f_2(8)$, odnosno kao zbroj $45 \cdot 1 + f_2(10 - 2 \cdot 1)$. Odavde "očitavamo" da je pripadna *optimalna* vrijednost varijable x_3 jednaka $x_3^* = 1$, tj. u naprtnjaču trebamo staviti jedan komad namirnice N_3.
Vrijednost $f_2(8) = 198$ dobivena je kao zbroj 102 + 96, odnosno kao zbroj $102 + f_1(5)$, odnosno kao zbroj $51 \cdot 2 + f_1(8 - 1.5 \cdot 2)$. Odavde očitavamo da je pripadna *optimalna* vrijednost varijable x_2 jednaka $x_2^* = 2$, tj. u naprtnjaču trebamo staviti dva komada namirnice N_2.
Vrijednost $f_1(5) = 102$ dobivena je kao umnožak $48 \cdot 2$, pa je pripadna *optimalna* vrijednost varijable x_1 jednaka $x_1^* = 2$, tj. u naprtnjaču trebamo staviti dva komada namirnice N_1.

Dakle, traženi optimalan izbor namirnica je: po 2 komada svake od namirnica N_1 i N_2, te jedan komad namirnice N_3. Pripadna optimalna vrijednost tih namirnica iznosi 243 kn, a njihova ukupna masa $2 \cdot 2.5 + 2 \cdot 1.5 + 1 \cdot 2 = 10$ kg. Toliki je i kapacitet naprtnjače, pa smo optimalnim izborom namirnica potpuno ispunili naprtnjaču.

<u>*Rješenje pomoću programa Winqsb*</u>: Postupak rješavanja analogan je onima opisanima u rješenjima prethodnih primjera. Nazovimo problem koji rješavamo *Primjer* 5, a kao ukupan broj različitih vrsta namirnica upišimo 3.
U drugi stupac tablice za unos polaznih podataka upišimo redom nazive namirnica: N1, N2 i N3.
U treći stupac upišimo raspoložive količine namirnica: 2, 2 i 3.
U četvrti stupac upišimo jedinične mase namirnica: 2.5, 1.5 i 2.
U peti stupac upišimo odgovarajuće komponente funkcije cilja: 48*X, 51*X i 45*X.
Kao kapacitet naprtnjače (*Capacity*) u posljednji redak upišimo 10.

Pokrenimo proceduru *Solve and Analyze*.

U drugom stupcu dobivene tablice nalazi se količina svake pojedine namirnice koja tvori optimalan izbor. Ti su brojevi redom jednaki 2, 2 i 1, pa zaključujemo da traženi optimalan izbor tvore po dva komada svake od namirnica N_1 i N_2, te jedan komad namirnice N_3. Pripadna ukupna masa iznosi točno 10 kg, pa je kapacitet ostatka naprtnjače jednak 0, tj. naprtnjača je potpuno ispunjena. Optimalna ukupna vrijednost svih stavljenih namirnica iznosi 243 kn.

Zadaci za vježbu:

1. Na raspolaganju nam je vrećica kapaciteta 2 kg, te po točno jedan komad svake od sljedećih namirnica: kruh, pašteta od tune, keksi, čokoladno mlijeko i sok od naranče. Podaci o jediničnoj masi i jediničnoj cijeni svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*jedinična masa* [g]	*jedinična cijena* [kn]
kruh	500	4
pašteta	100	17
keksi	350	12
čokoladno mlijeko	1000	10
sok od naranče	1500	8

Treba odabrati namirnice što veće vrijednosti tako da njihova ukupna masa ne bude veća od kapaciteta vrećice.
a) Formirajte odgovarajući matematički model, pa "klasično" i pomoću programa *Winqsb* odredite optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.
b) Riješite **a)** podzadatak uz dodatan uvjet da u vrećicu obavezno treba staviti sok od naranče.

2. Snjeguljica se sprema otići na plažu *Raj na zemlji*, pa bira što će sve ponijeti. Na raspolaganju ima po jedan komad svakoga od sljedeća četiri predmeta:

predmet	*jedinična masa* [kg]	*jedinična vrijednost* [n.j]
sunčane naočale	0.05	50
kupaći kostim	0.3	450
ručnik	0.4	520
termos-boca	0.25	100

Budući da će na plažu obavezno ponijeti i neke druge predmete (npr. međunarodno priznate popularno-znanstvene časopise *Extra* i *Story*), Snjeguljica je unaprijed odlučila da će ponijeti najviše 0.5 kg svih gore navedenih predmeta.
a) Formirajte odgovarajući matematički model, pa "klasično" i pomoću programa *Winqsb* odredite optimalan izbor predmeta i pripadnu optimalnu ukupnu vrijednost.
b) Riješite **a)** podzadatak uz dodatan uvjet da Snjeguljica na plažu obavezno mora ponijeti kupaći kostim.

3. Na raspolaganju nam je proizvoljan broj svake od triju različitih vrsta namirnica: N_1, N_2 i N_3, te torba kapaciteta 20 kg. Podaci o jediničnoj cijeni i jediničnoj masi svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*jedinična cijena* [n.j.]	*jedinična masa* [kg]
N_1	15	7
N_2	5	5
N_3	6	4

Treba odabrati namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne bude veća od kapaciteta torbe.
a) Formirajte odgovarajući matematički model, pa "klasično" i pomoću programa *Winqsb* odredite optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.

b) Riješite **a)** podzadatak uz dodatan uvjet da u torbu treba staviti točno jedan komad namirnice N_2.
c) Koristeći program *Winqsb* riješite **a)** podzadatak uz dodatan uvjet da u torbu treba staviti *barem jedan* komad namirnice N_2.

4. Na raspolaganju su nam točno 2 komada svake od namirnica N_1 i N_3, točno 3 komada svake od namirnica N_2 i N_4, te naprtnjača kapaciteta 20 kg. Podaci o jediničnoj masi i jediničnoj cijeni svake pojedine vrste namirnica zadani su u sljedećoj tablici.

namirnica	*jedinična masa* [kg/kom.]	*jedinična cijena* [kn/kom.]
N_1	6	6
N_2	8	8
N_3	5	5
N_4	7	7

Treba odabrati namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne bude veća od kapaciteta naprtnjače.

a) Formirajte odgovarajući matematički model, pa "klasično" i pomoću programa *Winqsb* odredite optimalan izbor namirnica, te pripadnu optimalnu ukupnu vrijednost.
b) Riješite **a)** podzadatak uz dodatni uvjet da u naprtnjači mora biti točno jedan komad namirnice N_1.
c) Riješite **a)** podzadatak uz dodatni uvjet da u naprtnjači mora biti sva raspoloživa količina namirnice N_3.

5. Na raspolaganju nam je vozilo maksimalne nosivosti 2 t. U vozilo treba utovariti ukupno 10 različitih vrsta namirnica tako da njihova ukupna vrijednost bude što veća. Niti jedan komad bilo koje namirnice nije dozvoljeno dijeliti na manje dijelove. Podaci o jedinčnoj cijeni i jediničnoj masi svake pojedine namirnice navedeni su u donjoj tablici.

namirnica	*komada*	*jedinična masa* [kg]	*jedinična cijena* [kn]
N_1	200	2	10.5
N_2	400	1	4.5
N_3	150	1.5	8.75
N_4	300	0.5	9.5
N_5	400	2.5	5.25
N_6	500	1.5	9.75
N_7	300	2.5	9.25
N_8	450	0.5	7.5
N_9	350	2	11.25
N_{10}	250	1	6.75

a) Formirajte odgovarajući matematički model, pa pomoću programa *Winqsb* odredite optimalan izbor namirnica, njihovu pripadnu optimalnu ukupnu vrijednost, te postotak iskorištenosti nosivosti vozila nakon utovara optimalnoga izbora namirnica.
b) Riješite podzadatak **a)** uz dodatni uvjet da se u vozilo moraju utovariti svi raspoloživi komadi namirnice N_2.
c) Riješite podzadatak **a)** uz dodatni uvjet da se u vozilo mora utovariti točno 100 komada namirnice N_4 i točno 200 komada namirnice N_8.

6. Pet različitih datoteka treba presnimiti s čvrstoga diska na praznu disketu memorijskoga kapaciteta 1.44 MB. Podaci o veličini svake datoteke i najkraćem vremenu potrebnom za njezino presnimavanje na disketu zadani su u sljedećoj tablici.

datoteka	*veličina* [kB]	*najkraće vrijeme presnimavanja* [s]
D_1	468	7
D_2	106	3
D_3	570	9
D_4	757	11
D_5	816	13

Treba naći izbor datoteka čija ukupna veličina neće premašiti memorijski kapacitet diskete tako da vrijeme presnimavanja svih izabranih datoteka bude što dulje. Nije dozvoljeno presnimavanje dijela bilo koje datoteke.

a) Formulirajte odgovarajući matematički model, pa odredite optimalan izbor datoteka, pripadno optimalno ukupno vrijeme presnimavanja i koliko će memorijskoga prostora ostati na disketi nakon presnimavanja optimalno izabranih datoteka.
b) Riješite **a)** podzadatak uz dodatni uvjet da se na disketu obavezno mora presnimiti datoteka D_1.
c) Riješite **a)** podzadatak uz dodatni uvjet da se na disketu obavezno mora presnimiti datoteka najveće veličine.

7. Interpretirajte sljedeće optimizacijske probleme kao odgovarajuće probleme naprtnjače pa ih riješite "klasično" i pomoću programa *Winqsb*:

a) maksimizirati $f(x_1, x_2, x_3, x_4) = 10 \cdot x_1 + 7 \cdot x_2 + 25 \cdot x_3 + 24 \cdot x_4$

pod uvjetima

$$2 \cdot x_1 + x_2 + 6 \cdot x_3 + 5 \cdot x_4 \leq 7,$$

$$x_1, x_2, x_3, x_4 \in \{0, 1\};$$

b) maksimizirati $f(x_1, x_2, x_3, x_4) = 10 \cdot x_1 + 7 \cdot x_2 + 25 \cdot x_3 + 24 \cdot x_4$

pod uvjetima

$$2 \cdot x_1 + x_2 + 6 \cdot x_3 + 5 \cdot x_4 \leq 7,$$

$$x_1, x_2, x_3, x_4 \in \mathbf{N};$$

c) maksimizirati $f(x_1, x_2, x_3, x_4) = 10 \cdot x_1 + 7 \cdot x_2 + 25 \cdot x_3 + 24 \cdot x_4$

pod uvjetima

$$2 \cdot x_1 + x_2 + 6 \cdot x_3 + 5 \cdot x_4 \leq 7$$

$$0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 3,$$

$$x_1, x_2, x_3, x_4 \in \mathbf{N};$$

1.3. Problem jednostavne razdiobe ulaganja

Radi kraćega i jednostavnijega zapisa, označimo:
$[n]_0 = \{0, 1, 2, \ldots, n\}$ = prvih $n + 1$ elemenata skupa $\mathbf{N}_0 = \{0, 1, 2, \ldots\}$;
$[n] = \{1, 2, \ldots, n\}$ = skup prvih n prirodnih brojeva.

Problem jednostavne razdiobe ulaganja općenito podrazumijeva određivanje razdiobe određenoga novčanoga iznosa na određeni broj privrednih grana tako da ukupna očekivana neto–dobit nastala tim ulaganjima bude maksimalna. U ovu kategoriju pripada i problem minimizacije troškova proizvodnje: ako u jednakim vremenskim intervalima treba proizvesti određenu količinu nekoga proizvoda i ako su poznati troškovi proizvodnje u svakom pojedinom intervalu, napraviti raspored proizvodnje tako da ukupni troškovi budu minimalni. Navedene probleme, njihovo matematičko modeliranje i rješavanje dinamičkim programiranjem opisujemo na sljedećim primjerima.

Primjer 1. Uprava tvrtke "*Lopužić* d.o.o." želi uložiti najviše 8 n.j. u tri odjela tvrtke tako da iznos svakoga pojedinoga uloga bude cjelobrojan, te da se u svaki odjel uloži najviše 5 n.j. (Dakle, moguće je da se u neki odjel ne uloži ništa, kao i da se ne uloži sav predviđeni kapital.) Kriterij za razdiobu ulaganja su ostvarene neto–dobiti u pojedinim odjelima koje linearno ovise o iznosima uloženoga kapitala. Uloži li se po 1 n.j. u svaki odjel, prvi odjel ostvarit će neto–dobit od 4 n.j., drugi odjel neto–dobit od 5 n.j., a treći odjel neto–dobit od 2 n.j. Formirajmo odgovarajući matematički model, pa odredimo optimalan iznos ulaganja u svaki odjel tako da ukupna neto–dobit bude maksimalna.

Matematički model: Za svaki $i \in [3]$ označimo s x_i iznos ulaganja u i – ti odjel. Ukupna neto–dobit $D(x_1, x_2, x_3)$ nastala tim ulaganjima iznosi

$$D(x_1, x_2, x_3) = 4 \cdot x_1 + 5 \cdot x_2 + 2 \cdot x_3.$$

Stoga je *matematički model* promatranoga problema:

$$\text{maksimizirati } D(x_1, x_2, x_3) = 4 \cdot x_1 + 5 \cdot x_2 + 2 \cdot x_3$$

pod uvjetima

$$x_1 + x_2 + x_3 \leq 8$$
$$x_1, x_2, x_3 \in [5]_0.$$

"Klasično" rješenje: Osnovna ideja je podijeliti problem u tri faze tako da se, za svaki $i \in [3]$, u i – toj fazi rješavanja promatra ulaganje kapitala u odjele 1, …, i tako da se u izračunima u svakoj pojedinoj fazi koriste optimalne vrijednosti dobivene u prethodnim fazama. Radi kraćega zapisa, označimo $C = [8]_0$.

Dakle, *u prvoj fazi* određujemo optimalnu neto–dobit kad se sav kapital ulaže samo u prvi odjel. Uz ulaganje 1 n.j. u prvi odjel ostvaruje se neto–dobit od 4 n.j., pa će za ulog od x_1 n.j. ostvarena neto–dobit biti $4 \cdot x_1$ n.j. Primijetimo da, zbog uvjeta zadatka, ne znamo točan iznos ukupno uloženoga kapitala, nego znamo jedino da je taj iznos neki prirodan broj najviše jednak 8. Ekvivalentno, ukupni uloženi kapital je *neki* element skupa C, ali ne znamo koji je

to točno element. Stoga za *svaki* element $c \in C$ računamo vrijednosti realne funkcije f_1 definirane s:

$$f_1(c) = 4 \cdot c.$$

Te su vrijednosti, zapravo, *optimalne* neto–dobiti u slučaju ukupan kapital od c n.j. ulažemo samo u prvi odjel. Odmah napomenimo da za vrijednosti $c \in C$ takve da je $c \geq 5$ stavljamo $f_1(s) = f_1(5)$ jer ukupno ulaganje u prvi odjel ne može biti strogo veće od 5 n.j. Tako redom dobivamo:

$$\begin{aligned}
f_1(0) &= 4 \cdot 0 = 0;\\
f_1(1) &= 4 \cdot 1 = 4,\\
f_1(2) &= 4 \cdot 2 = 8,\\
f_1(3) &= 4 \cdot 3 = 12,\\
f_1(4) &= 4 \cdot 4 = 16,\\
f_1(5) &= 4 \cdot 5 = 20,\\
f_1(6) &= 4 \cdot 5 = 20,\\
f_1(7) &= 4 \cdot 5 = 20,\\
f_1(8) &= 4 \cdot 5 = 20,
\end{aligned}$$

U *drugoj fazi* razmatramo slučaj ulaganja ukupnoga kapitala samo u prvi i drugi odjel. Razmišljamo ovako:

Pretpostavimo da ukupno ulažemo c n.j., pri čemu je $c \in C$. Za svaki $i \in [2]$ označimo s x_i iznos koji ulažemo u i – ti odjel. Tada mora vrijediti jednakost

$$x_1 + x_2 = c,$$

iz koje je

$$x_1 = c - x_2.$$

Neto–dobit koju će ostvariti drugi odjel iznosi $5 \cdot x_2$ n.j., a *optimalna* neto–dobit koju će ostvariti prvi odjel, prema definiciji funkcije f_1, iznosi $f_1(x_1)$, tj. $f_1(c - x_2)$ n.j. Stoga za svaki element $c \in C$ računamo vrijednosti realne funkcije f_2 definirane s:

$$f_2(c) = \max_{\substack{0 \leq x_2 \leq \min\{c,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(c - x_2)\right].$$

Te su vrijednosti, zapravo, optimalne neto–dobiti ukoliko ukupni kapital od c n.j. uložimo samo u prvi i drugi odjel. Uvjeti pri kojima određujemo maksimum posljedica su činjenica da iznos x_2 mora biti cjelobrojan i ne smije premašiti niti ukupni kapital od c n.j., niti 5. n.j. Imamo redom:

$f_2(0) = \max\limits_{\substack{0 \le x_2 \le \min\{0,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(0 - x_2)\right] = \max\limits_{0 \le x_2 \le 0} \left[5 \cdot x_2 + f_1(0 - x_2)\right] = 5 \cdot 0 + f_1(0) = 0 + 0 = 0;$

$f_2(1) = \max\limits_{\substack{0 \le x_2 \le \min\{1,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(1 - x_2)\right] = \max\limits_{0 \le x_2 \le 1} \left[5 \cdot x_2 + f_1(1 - x_2)\right] = \max\{5 \cdot 0 + f_1(1 - 0), 5 \cdot 1 + f_1(1 - 1)\} = \max\{0 + f_1(1), 5 + f_1(0)\} = \max\{0 + 4, 5 + 0\} = 5;$

$f_2(2) = \max\limits_{\substack{0 \le x_2 \le \min\{2,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(2 - x_2)\right] = \max\limits_{0 \le x_2 \le 2} \left[5 \cdot x_2 + f_1(2 - x_2)\right] = \max\{5 \cdot 0 + f_1(2 - 0), 5 \cdot 1 + f_1(2 - 1), 5 \cdot 2 + f_1(2 - 2)\} = \max\{0 + f_1(2), 5 + f_1(1), 10 + f_1(0)\} = \max\{0 + 8, 5 + 4, 10 + 0\} = 10;$

$f_2(3) = \max\limits_{\substack{0 \le x_2 \le \min\{3,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(3 - x_2)\right] = \max\limits_{0 \le x_2 \le 3} \left[5 \cdot x_2 + f_1(3 - x_2)\right] = \max\{5 \cdot 0 + f_1(3 - 0), 5 \cdot 1 + f_1(3 - 1), 5 \cdot 2 + f_1(3 - 2), 5 \cdot 3 + f_1(3 - 3)\} = \max\{0 + f_1(3), 5 + f_1(2), 10 + f_1(1), 15 + f_1(0)\} = \max\{0 + 12, 5 + 8, 10 + 4, 15 + 0\} = 15;$

$f_2(4) = \max\limits_{\substack{0 \le x_2 \le \min\{4,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(4 - x_2)\right] = \max\limits_{0 \le x_2 \le 4} \left[5 \cdot x_2 + f_1(4 - x_2)\right] = \max\{5 \cdot 0 + f_1(4 - 0), 5 \cdot 1 + f_1(4 - 1), 5 \cdot 2 + f_1(4 - 2), 5 \cdot 3 + f_1(4 - 3), 5 \cdot 4 + f_1(4 - 4)\} = \max\{0 + f_1(4), 5 + f_1(3), 10 + f_1(2), 15 + f_1(1), 20 + f_1(0)\} = \max\{0 + 16, 5 + 12, 10 + 8, 15 + 4, 20 + 0\} = 20;$

$f_2(5) = \max\limits_{\substack{0 \le x_2 \le \min\{5,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(5 - x_2)\right] = \max\limits_{0 \le x_2 \le 5} \left[5 \cdot x_2 + f_1(5 - x_2)\right] = \max\{5 \cdot 0 + f_1(5 - 0), 5 \cdot 1 + f_1(5 - 1), 5 \cdot 2 + f_1(5 - 2), 5 \cdot 3 + f_1(5 - 3), 5 \cdot 4 + f_1(5 - 4), 5 \cdot 5 + f_1(5 - 5)\} = \max\{0 + f_1(5), 5 + f_1(4), 10 + f_1(3), 15 + f_1(2), 20 + f_1(1), 25 + f_1(0)\} = \max\{0 + 20, 5 + 16, 10 + 12, 15 + 8, 20 + 4, 25 + 0\} = 25;$

$f_2(6) = \max\limits_{\substack{0 \le x_2 \le \min\{6,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(6 - x_2)\right] = \max\limits_{0 \le x_2 \le 5} \left[5 \cdot x_2 + f_1(6 - x_2)\right] = \max\{5 \cdot 0 + f_1(6 - 0), 5 \cdot 1 + f_1(6 - 1), 5 \cdot 2 + f_1(6 - 2), 5 \cdot 3 + f_1(6 - 3), 5 \cdot 4 + f_1(6 - 4), 5 \cdot 5 + f_1(6 - 5)\} = \max\{0 + f_1(6), 5 + f_1(5), 10 + f_1(4), 15 + f_1(3), 20 + f_1(2), 25 + f_1(1)\} = \max\{0 + 20, 5 + 20, 10 + 16, 15 + 12, 20 + 8, 25 + 4\} = 29;$

$f_2(7) = \max\limits_{\substack{0 \le x_2 \le \min\{7,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(7 - x_2)\right] = \max\limits_{0 \le x_2 \le 5} \left[5 \cdot x_2 + f_1(7 - x_2)\right] = \max\{5 \cdot 0 + f_1(7 - 0), 5 \cdot 1 + f_1(7 - 1), 5 \cdot 2 + f_1(7 - 2), 5 \cdot 3 + f_1(7 - 3), 5 \cdot 4 + f_1(7 - 4), 5 \cdot 5 + f_1(7 - 5)\} = \max\{0 + f_1(7), 5 + f_1(6), 10 + f_1(5), 15 + f_1(4), 20 + f_1(3), 25 + f_1(2)\} = \max\{0 + 20, 5 + 20, 10 + 20, 15 + 16, 20 + 12, 25 + 8\} = 33;$

$f_2(8) = \max\limits_{\substack{0 \le x_2 \le \min\{8,5\} \\ x_2 \in \mathbb{Z}}} \left[5 \cdot x_2 + f_1(8 - x_2)\right] = \max\limits_{0 \le x_2 \le 5} \left[5 \cdot x_2 + f_1(8 - x_2)\right] = \max\{5 \cdot 0 + f_1(8 - 0), 5 \cdot 1 + f_1(8 - 1), 5 \cdot 2 + f_1(8 - 2), 5 \cdot 3 + f_1(8 - 3), 5 \cdot 4 + f_1(8 - 4), 5 \cdot 5 + f_1(8 - 5)\} = \max\{0 +$

$+ f_1(8)$, $5 + f_1(7)$, $10 + f_1(6)$, $15 + f_1(5)$, $20 + f_1(4)$, $25 + f_1(3)\} = \max\{0 + 20, 5 + 20, 10 + 20, 15 + 20, 20 + 16, 25 + 12\} = 37$.

Dobiveni rezultat smo mogli i unaprijed predvidjeti tzv. "*pohlepnim*" *principom*: uzmemo najveći mogući kapital i uložimo najviše što možemo u profitabilniji od dvaju odabranih odjela, a ostatak potom u manje profitabilan odjel. U ovom slučaju to znači da ulažemo svih 8 n.j. tako da 5 n.j. uložimo u drugi (profitabilniji) odjel, a ostatak od 8 – 5 = 3 n.j. u prvi odjel, pa će ukupna optimalna neto–dobit biti $5 \cdot 5 + 4 \cdot 3 = 37$ n.j.

U *trećoj fazi* razmatramo ulaganje kapitala u sva tri odjela. Razmišljamo analogno kao u drugoj fazi:

Pretpostavimo da ulažemo c n.j. pri čemu je $c \in C$. Za svaki $i \in [3]$ označimo s x_i iznos uložen u i – ti odjel. Tada mora vrijediti jednakost

$$x_1 + x_2 + x_3 = c,$$

iz koje je

$$x_1 + x_2 = c - x_3.$$

Neto–dobit koju će ostvariti treći odjel iznosi $2 \cdot x_3$ n.j., a *optimalna* neto–dobit koju će ostvariti prvi i drugi odjel zajedno, prema definiciji funkcije f_2, iznosi $f_2(x_1 + x_2)$ n.j., odnosno $f_2(s - x_3)$ n.j. Stoga za svaki element $c \in C$ računamo vrijednosti realne funkcije f_3 definirane s

$$f_3(c) = \max_{\substack{0 \le x_3 \le \min\{c,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(0 - x_3)\right]$$

Te su vrijednosti, zapravo, optimalne neto–dobiti ukoliko ukupni kapital od c n.j. uložimo u sva tri odjela. Uvjeti pri kojima određujemo maksimum posljedica su činjenica da iznos x_3 mora biti cjelobrojan i ne smije premašiti niti ukupni kapital od c n.j., niti 5. n.j. Imamo redom:

$$f_3(0) = \max_{\substack{0 \le x_3 \le \min\{0,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(0 - x_3)\right] = \max_{0 \le x_3 \le 0} \left[2 \cdot x_3 + f_2(0 - x_3)\right] = 2 \cdot 0 + f_2(0) = 0 + 0 = 0;$$

$$f_3(1) = \max_{\substack{0 \le x_3 \le \min\{1,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(1 - x_3)\right] = \max_{0 \le x_3 \le 1} \left[2 \cdot x_3 + f_2(1 - x_3)\right] = \max\{2 \cdot 0 + f_2(1 - 0), 2 \cdot 1 + f_2(1 - 1)\} = \max\{0 + f_2(1), 2 + f_2(0)\} = \max\{0 + 5, 2 + 0\} = 5;$$

$$f_3(2) = \max_{\substack{0 \le x_3 \le \min\{2,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(2 - x_3)\right] = \max_{0 \le x_3 \le 2} \left[2 \cdot x_3 + f_2(2 - x_3)\right] = \max\{2 \cdot 0 + f_2(2 - 0), 2 \cdot 1 + f_2(2 - 1), 2 \cdot 2 + f_2(2 - 2)\} = \max\{0 + f_2(2), 2 + f_2(1), 4 + f_2(0)\} = \max\{0 + 10, 2 + 5, 4 + 0\} = 10;$$

$f_3(3) = \max\limits_{\substack{0 \le x_3 \le \min\{3,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(3 - x_3)\right] = \max\limits_{0 \le x_3 \le 3} \left[2 \cdot x_3 + f_2(3 - x_3)\right] = \max\{2 \cdot 0 + f_2(3 - 0), 2 \cdot 1 + f_2(3 - 1), 2 \cdot 2 + f_2(3 - 2), 2 \cdot 3 + f_2(3 - 3)\} = \max\{0 + f_2(3), 2 + f_2(2), 4 + f_2(1), 6 + f_2(0)\} = \max\{0 + 15, 2 + 10, 4 + 5, 6 + 0\} = 15;$

$f_3(4) = \max\limits_{\substack{0 \le x_3 \le \min\{4,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(4 - x_3)\right] = \max\limits_{0 \le x_3 \le 4} \left[2 \cdot x_3 + f_2(4 - x_3)\right] = \max\{2 \cdot 0 + f_2(4 - 0), 2 \cdot 1 + f_2(4 - 1), 2 \cdot 2 + f_2(4 - 2), 2 \cdot 3 + f_2(4 - 3), 2 \cdot 4 + f_2(4 - 4)\} = \max\{0 + f_2(4), 2 + f_2(3), 4 + f_2(2), 6 + f_2(1), 8 + f_2(0)\} = \max\{0 + 20, 2 + 15, 4 + 10, 6 + 5, 8 + 0\} = 20;$

$f_3(5) = \max\limits_{\substack{0 \le x_3 \le \min\{5,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(5 - x_3)\right] = \max\limits_{0 \le x_3 \le 5} \left[2 \cdot x_3 + f_2(5 - x_3)\right] = \max\{2 \cdot 0 + f_2(5 - 0), 2 \cdot 1 + f_2(5 - 1), 2 \cdot 2 + f_2(5 - 2), 2 \cdot 3 + f_2(5 - 3), 2 \cdot 4 + f_2(5 - 4), 2 \cdot 5 + f_2(5 - 5)\} = \max\{0 + f_2(5), 2 + f_2(4), 4 + f_2(3), 6 + f_2(2), 8 + f_2(1), 10 + f_2(0)\} = \max\{0 + 25, 2 + 20, 4 + 15, 6 + 10, 8 + 5, 10 + 0\} = 25;$

$f_3(6) = \max\limits_{\substack{0 \le x_3 \le \min\{6,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(6 - x_3)\right] = \max\limits_{0 \le x_3 \le 5} \left[2 \cdot x_3 + f_2(6 - x_3)\right] = \max\{2 \cdot 0 + f_2(6 - 0), 2 \cdot 1 + f_2(6 - 1), 2 \cdot 2 + f_2(6 - 2), 2 \cdot 3 + f_2(6 - 3), 2 \cdot 4 + f_2(6 - 4), 2 \cdot 5 + f_2(6 - 5)\} = \max\{0 + f_2(6), 2 + f_2(5), 4 + f_2(4), 6 + f_2(3), 8 + f_2(2), 10 + f_2(1)\} = \max\{0 + 29, 2 + 25, 4 + 20, 6 + 15, 8 + 10, 10 + 5\} = 29;$

$f_3(7) = \max\limits_{\substack{0 \le x_3 \le \min\{7,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(7 - x_3)\right] = \max\limits_{0 \le x_3 \le 5} \left[2 \cdot x_3 + f_2(7 - x_3)\right] = \max\{2 \cdot 0 + f_2(7 - 0), 2 \cdot 1 + f_2(7 - 1), 2 \cdot 2 + f_2(7 - 2), 2 \cdot 3 + f_2(7 - 3), 2 \cdot 4 + f_2(7 - 4), 2 \cdot 5 + f_2(7 - 5)\} = \max\{0 + f_2(7), 2 + f_2(6), 4 + f_2(5), 6 + f_2(4), 8 + f_2(3), 10 + f_2(2)\} = \max\{0 + 33, 2 + 29, 4 + 25, 6 + 20, 8 + 15, 10 + 10\} = 33;$

$f_3(8) = \max\limits_{\substack{0 \le x_3 \le \min\{8,5\} \\ x_3 \in \mathbb{Z}}} \left[2 \cdot x_3 + f_2(8 - x_3)\right] = \max\limits_{0 \le x_3 \le 5} \left[2 \cdot x_3 + f_2(8 - x_3)\right] = \max\{2 \cdot 0 + f_2(8 - 0), 2 \cdot 1 + f_2(8 - 1), 2 \cdot 2 + f_2(8 - 2), 2 \cdot 3 + f_2(8 - 3), 2 \cdot 4 + f_2(8 - 4), 2 \cdot 5 + f_2(8 - 5)\} = \max\{0 + f_2(8), 2 + f_2(7), 4 + f_2(6), 6 + f_2(5), 8 + f_2(4), 10 + f_2(3)\} = \max\{0 + 37, 2 + 33, 4 + 29, 6 + 25, 8 + 20, 10 + 15\} = 37.$

Optimalna vrijednost funkcije f_3 jednaka je 37 i postiže se za $c = 8$, što znači da ćemo najveću ukupnu neto–dobit ostvariti ulaganjem cjelokupnoga raspoloživoga kapitala od 8 n.j. Odredimo pripadne optimalne iznose ulaganja u svaki odjel.

Vrijednost $f_3(8) = 37$ postiže se kao zbroj $0 + 37$, odnosno kao zbroj $0 + f_2(8)$, tj. kao zbroj $2 \cdot 0 + f_2(8 - 0)$. Odatle "očitavamo" da je optimalna vrijednost varijable x_3 jednaka $x_3^* = 0$, pa zaključujemo da u treći odjel ne ulažemo ništa.

Vrijednost $f_2(8) = 37$ postiže se kao zbroj 25 + 12, odnosno kao zbroj $25 + f_1(3)$, tj. kao zbroj $5 \cdot 5 + f_1(8 - 5)$. Odatle "očitavamo" da je optimalna vrijednost varijable x_2 jednaka $x_2^* = 5$, pa u drugi odjel ulažemo ukupno 5 n.j.

Vrijednost $f_1(3) = 12$ postiže se kao umnožak $4 \cdot 3$, pa je optimalna vrijednost varijable x_1 jednaka $x_1^* = 3$, pa u prvi odjel ulažemo ukupno 3 n.j.

Prema tome, optimalan plan ulaganja je: *uložiti svih* 8 *n.j. tako da se u prvi odjel ulože* 3 *n.j.*, *a u drugi odjel* 5 *n.j.* (U treći odjel nema nikakvih ulaganja). *Pripadna optimalna neto–dobit iznosi* 37 *n.j.*

Dobivene rezultate možemo pregledno prikazati sljedećom tablicom:

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*	$f_3(s)$	x_3^*
0	0	0	0	0	0	0
1	4	1	5	1	5	0
2	8	2	10	2	10	0
3	12	3	15	3	13	0
4	16	4	20	4	20	0
5	20	5	25	5	25	0
6	20	5	29	5	29	0
7	20	5	33	5	33	0
8	20	5	37	5	37	0

Rješenje pomoću programa Winqsb: Navedeni problem možemo shvatiti i kao sljedeći *zatvoreni problem naprtnjače*:

Na raspolaganju nam je po 5 *komada svake od triju vrsta namirnica*: N_1, N_2 i N_3, *te naprtnjača kapaciteta* 8 kg. *Podaci o jediničnoj cijeni i jediničnoj masi svake namirnice navedeni su u donjoj tablici*.

namirnica	*jedinična cijena* [n.j.]	*jedinična masa* [kg]
N_1	4	1
N_2	5	1
N_3	2	1

Treba odabrati namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne bude veća od kapaciteta naprtnjače. Formirajmo odgovarajući matematički model, pa odredimo optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.

Ovako definiran problem ima *isti matematički model* kao i polazni problem, ali s drugačijim *interpretacijama* varijabli, funkcije cilja i uvjeta. Takvi se problemi nazivaju *izomorfnima* (slično kao izomorfni grafovi u teoriji grafova): njihovi matematički modeli su jednaki (i rješavaju se istim metodama), a eventualne razlike su u numeričkim vrijednostima

koeficijenata koji se pojavljuju u modelu[16], te interpretaciji ulaznih i izlaznih podataka. Jedan od osnovnih zadataka operacijskih istraživanja je *klasificirati* razmatrane probleme u *klase* međusobno izomorfnih problema i naći algoritme za rješavanje problema svake pojedine klase.

Riješimo, dakle, navedeni problem pomoću programa *Winqsb*. Pokretanjem potprograma *DP* odaberimo tip *Knapsack Problem*. Nazovimo problem koji ćemo rješavati *Primjer* 1, tj. upišimo:

Problem Title: Primjer 1
Number of Items: 3

Potom kliknimo na *OK*. U sljedećem koraku unosimo ulazne podatke.

U drugi stupac (*Item Identification*) upišimo redom O1, O2 i O3 jer će nam takve oznake na kraju "otkriti" svotu uloženu u svaki pojedini odjel.
U treći stupac (*Units Available*) upisujemo najveći mogući iznos koji može biti uložen u svaki pojedini odjel. Ti iznosi su jednaki 5, pa upisujemo: 5, 5, 5.
U četvrti stupac (*Unit Capacity Required*) treba upisati koeficijente uz nepoznanice x_1, x_2 i x_3 u uvjetu $x_1 + x_2 + x_3 \leq 8$. Oni su jednaki 1, pa u četvrti stupac upišimo: 1, 1, 1. Istaknimo odmah da su prigodom modeliranja problema jednostavne razdiobe ulaganja koeficijenti u ovom stupcu *uvijek* jednaki 1, dok kod problema složene razdiobe ulaganja (obrađenoga u točki 1.4.) ti koeficijenti općenito mogu biti i nenegativni realni brojevi različiti od 1.
U peti stupac (*Return Function* (*X*: *Item ID*)) upišimo redom komponente funkcije cilja koje se odnose na odjel iz pripadnoga retka. Tako u prvi redak toga stupca upišimo 4*X, u drugi 5*X, a u treći 2*X.
U posljednji *redak* pored natpisa *Capacity*= upišimo ukupan iznos kapitala, tj. 8.

Ovime je unos ulaznih podataka završen. Kliknimo lijevom tipkom miša na izbornik *Solve and Analyze* i odaberimo opciju *Solve the Problem.* Dobivamo izlaznu tablicu sa svim potrebnim rezultatima.

Iz trećega stupca (*Decision Quantity* (*X*)) očitavamo optimalan iznos ulaganja u svaki pojedini odjel: u prvi odjel (O_1) treba uložiti 3 n.j., u drugi odjel (O_2) 5 n.j., dok se u treći odjel (O_3) ne ulaže ništa. Optimalnu neto–dobit očitavamo iz posljednjega retka (*Total Return Value*) i ona iznosi 37. n.j.

<u>Primjer 2.</u> Riješimo prethodni primjer uz dodatni uvjet da se u treći odjel unaprijed mora uložiti 1 n.j., a ostatak kapitala optimalno raspodijeliti na sva tri odjela.

Unaprijed smo, dakle, donijeli odluku da u treći odjel uložimo 1 n.j., pa nam na raspolaganju ostaje još najviše 8 – 1 = 7 n.j. Stoga je novi <u>*matematički model*</u>:

[16] Izomorfni modeli se mogu razlikovati u vrijednostima koeficijenata, ali ne i u *tipovima* vrijednosti varijabli. Npr. ako su u modelu *A* varijable nužno cjelobrojne, a u modelu *B* nužno realne, onda takvi modeli nisu izomorfni. Slično vrijedi i za tipove (ne)jednadžbi koje opisuju uvjete.

$$\text{maksimizirati } D(x_1, x_2, x_3) = 4 \cdot x_1 + 5 \cdot x_2 + 2 \cdot x_3$$

pod uvjetima

$$x_1 + x_2 + x_3 \leq 7$$
$$x_1, x_2, x_3 \in [5]_0.$$

"*Klasično" rješenje*: U ovom slučaju nema razloga ponovno provoditi sva razmatranja iz rješenja Primjera 1. Naime, budući da preostalih 7 n.j. trebamo razdijeliti na sva tri odjela, pripadna optimalna vrijednost jednaka je $f_3(7)$, gdje je f_3 realna funkcija definirana u "klasičnom" rješenju Primjera 1. Vrijednost $f_3(7) = 33$ postiže se za $x_1^* = 5$, $x_2^* = 2$ i $x_3^* = 0$, pa je optimalan plan ulaganja: *uložiti svih* 8 n.j., *i to* 5 n.j. *u prvi odjel*, 2 n.j. *u drugi odjel i* (unaprijed određenih) 1 n.j. *u treći odjel*. Optimalna ukupna neto–dobit iznosi $f_3(7) + 1 \cdot 2 =$ $= 33 + 2 = 35$ n.j.

Rješenje pomoću programa Winqsb: Iako se na prvi pogled čini da bismo ovdje mogli primijeniti *što–ako* analizu i odmah dobiti rezultat, to nije točno. Naime, zadavanje podataka u okviru *što – ako* analize podrazumijeva *fiksiranje* vrijednosti (barem) jedne varijable i promjenu vrijednosti preostalih varijabli. Ukoliko bismo pokrenuli *što – ako* analizu i u retku lijeve tablice koji odgovara trećem odjelu (O_3) upisali jedinicu, preostali iznos od 7 n.j. raspoređivao bi se isključivo na prvi i drugi odjel, što ne odgovara polaznim uvjetima problema (kapital od 8 – 1 = 7 n.j. treba rasporediti na sva tri odjela, a ne samo na prvi i drugi). Stoga ovaj problem ne možemo riješiti pomoću *što – ako* analize, nego izmjenom kapaciteta u posljednjemu retku tablice s ulaznim podacima: 8 treba zamijeniti sa 7.

Pokretanjem procedure *Solve and Analyze* dobiva se tablica iz čijega trećega stupca (*Decision Quantity* (*X*)) očitavamo optimalan iznos ulaganja kapitala od 7 n.j.: u prvi odjel (O_1) treba uložiti 2 n.j., u drugi odjel (O_2) 5 n.j., dok se u treći odjel (O_3) ne ulaže ništa. Pripadna optimalna neto–dobit (*Total Return Value*) iznosi 33. n.j., pa je optimalan plan ulaganja: *uložiti svih* 8 n.j., *i to* 5 n.j. *u prvi odjel*, 2 n.j. *u drugi odjel i* (unaprijed određenih) 1 n.j. *u treći odjel*. Pripadna optimalna ukupna neto–dobit iznosi 33 + 2 = 35 n.j.

Napomena: Isključivo zbog zadanih numeričkih vrijednosti pokretanje *što – ako* analize daje isti rezultat kao i gornje rješenje. Navedeni je rezultat, zapravo, rješenje sljedećega matematičkoga modela:

$$\text{maksimizirati } D(x_1, x_2, x_3) = 4 \cdot x_1 + 5 \cdot x_2 + 2 \cdot x_3$$

pod uvjetima

$$x_1 + x_2 + x_3 \leq 8,$$
$$x_3 = 1,$$
$$x_1, x_2 \in [5]_0$$

koji je izomorfan modelu

$$\text{maksimizirati } D(x_1, x_2) = 4 \cdot x_1 + 5 \cdot x_2 + 2$$

pod uvjetima

$$x_1 + x_2 \leq 7,$$
$$x_1, x_2 \in [5]_0.$$

Primjer 3. Tijekom tri mjeseca treba proizvesti točno 12 komada nekoga proizvoda, pri čemu je najveći mogući kapacitet proizvodnje 6 komada proizvoda mjesečno. Troškovi proizvodnje razmjerni su kvadratima broja proizvedenih proizvoda, te za svaki $i \in [3]$ vrijedi: ako se u i – tom mjesecu proizvede x_i komada proizvoda, troškovi proizvodnje iznose $w_i \cdot x_i^2$ n.j., gdje su w_i strogo pozitivni realni brojevi. Treba odrediti plan proizvodnje takav da ukupni troškovi proizvodnje budu minimalni.

a) Odredimo opći matematički model promatranoga problema.
b) Riješimo problem za konkretne vrijednosti: $w_1 = 10$, $w_2 = 24$ i $w_3 = 32$.

Matematički model: U postavci problema je već naznačeno da s x_i označavamo količinu (broj komada) proizvoda proizvedenu u i – tom mjesecu. Ukupan zbroj svih količina mora biti jednak 12, pa vrijedi jednakost:

$$x_1 + x_2 + x_3 = 12.$$

Troškovi proizvodnje u prvom mjesecu su $w_1 \cdot x_1^2$, troškovi proizvodnje u drugom mjesecu $w_2 \cdot x_2^2$, a troškovi proizvodnje u trećem mjesecu $w_3 \cdot x_3^2$. Stoga ukupni troškovi proizvodnje u navedena tri mjeseca iznose

$$f(x_1, x_2, x_3) = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + w_3 \cdot x_3^2 .$$

Nadalje, iz tipa promatranoga problema slijedi da brojevi x_i nužno moraju biti elementi skupa $\mathbf{N}_0$. Također, brojevi x_i ne smiju biti strogo veći od 6 jer je najveći mogući kapacitet proizvodnje jednak 6. Prema tome, *matematički model* promatranoga problema glasi:

$$\text{minimizirati } f(x_1, x_2, x_3) = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + w_3 \cdot x_3^2$$

pod uvjetima

$$x_1 + x_2 + x_3 = 12,$$
$$x_1, x_2, x_3 \in [6]_0.$$

"Klasično" rješenje: Za $w_1 = 10$, $w_2 = 24$ i $w_3 = 32$ dobivamo sljedeći problem:

$$\text{minimizirati } f(x_1, x_2, x_3) = 10 \cdot x_1^2 + 24 \cdot x_2^2 + 32 \cdot x_3^2$$

pod uvjetima

$$x_1 + x_2 + x_3 = 12,$$
$$x_1, x_2, x_3 \in [6]_0.$$

I ovdje bismo mogli razmišljati "pohlepno", tj. da u prvom mjesecu treba proizvesti točno 6 komada proizvoda (jer su najmanji troškovi proizvodnje upravo u tom mjesecu), a u drugom preostalih 6 (jer su troškovi proizvodnje u drugom mjesecu manji nego u trećem). Stoga bi traženi plan mogao biti $x_1 = x_2 = 6$, $x_3 = 0$, a minimalni troškovi $f(6, 6, 0) = 10 \cdot 6^2 + 24 \cdot 6^2 + 32 \cdot 0^2 = 1224$ n.j. Pokazat ćemo, međutim, da ovakav "pohlepni" pristup u ovom slučaju ne daje optimalno rješenje.

Analogno kao i u prethodnom primjeru, stavimo $S = [12]_0$. Primijetimo da u ovom primjeru znamo točan ukupan broj komada proizvoda koje treba proizvesti (to je 12), ali radi uvježbavanja tehnike dinamičkoga programiranja u svakoj ćemo fazi računati 13 odgovarajućih vrijednosti.

Dakle, *u prvoj fazi* za svaki $s \in S$ računamo vrijednosti realne funkcije

$$f_1(s) = 10 \cdot s^2.$$

Vrijednosti funkcije f_1 su optimalni troškovi proizvodnje ukoliko se u prvom mjesecu proizvede točno s komada proizvoda. Pritom za svaki $s \in S$ takav da je $s \geq 7$ dogovorno stavljamo da su troškovi proizvodnje beskonačni jer je najveći proizvodni kapacitet, prema uvjetima zadatka, jednak 6. Tako dobivamo sljedeću tablicu:

s	$f_1(s)$	x_1^*
0	0	0
1	10	1
2	40	2
3	90	3
4	160	4
5	250	5
6	360	6
7	∞	6
8	∞	6
9	∞	6
10	∞	6
11	∞	6
12	∞	6

Vrijednost x_1^* označava optimalnu proizvodnju u prvom mjesecu za koju se postiže pripadna optimalna vrijednost $f_1(s)$. U prvoj je fazi ovaj zapis praktički nepotreban, ali će nam korisno poslužiti u sljedećim fazama. Pritom smo u retcima u kojima je $f_1(s) = \infty$ dogovorno pisali $x_1^* = 6$ zbog već istaknute činjenice da je najveći proizvodni kapacitet jednak 6.

U *drugoj fazi* računamo vrijednosti realne funkcije

$$f_2(s) = \min_{\substack{0 \leq x_2 \leq \min\{s,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(s - x_2) \right].$$

Naime, u ovoj fazi pretpostavljamo da se proizvodnja ukupno s komada proizvoda odvija tijekom prva dva mjeseca. Ukoliko se u drugom mjesecu proizvede x_2 proizvoda, u prvom se mora proizvesti ukupno $s - x_2$ proizvoda. Troškovi proizvodnje u drugom mjesecu iznose $24 \cdot x_2^2$ n.j., dok *optimalni* troškovi proizvodnje u prvom mjesecu, sukladno definiciji funkcije $f_1(s)$, iznose $f_1(s - x_2)$. Zbroj tih troškova jednak je ukupnim troškovima proizvodnje u prva dva mjeseca, pa taj zbroj treba minimizirati uvažavajući uvjete da vrijednost varijable x_2 mora

biti cjelobrojna, ne veća od ukupne količine proizvoda koju treba proizvesti i ne veća od najvećega mjesečnoga kapaciteta proizvodnje.

Računamo redom:

$f_2(0) = \min\limits_{\substack{0 \le x_2 \le \min\{0,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(0 - x_2)\right] = \min\limits_{0 \le x_2 \le 0} \left[24 \cdot x_2^2 + f_1(0 - x_2)\right] = 24 \cdot 0^2 + f_1(0 - 0) = 0 + 0 =$
$= 0;$

$f_2(1) = \min\limits_{\substack{0 \le x_2 \le \min\{1,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(1 - x_2)\right] = \min\limits_{0 \le x_2 \le 1} \left[24 \cdot x_2^2 + f_1(1 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(1 - 0),$
$24 \cdot 1^2 + f_1(1 - 1)\} = \min\{0 + 10, 24 + 0\} = 10;$

$f_2(2) = \min\limits_{\substack{0 \le x_2 \le \min\{2,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(2 - x_2)\right] = \min\limits_{0 \le x_2 \le 2} \left[24 \cdot x_2^2 + f_1(2 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(2 - 0),$
$24 \cdot 1^2 + f_1(2 - 1), 24 \cdot 2^2 + f_1(2 - 2)\} = \min\{0 + 40, 24 + 10, 96 + 0\} = 34;$

$f_2(3) = \min\limits_{\substack{0 \le x_2 \le \min\{3,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(3 - x_2)\right] = \min\limits_{0 \le x_2 \le 3} \left[24 \cdot x_2^2 + f_1(3 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(3 - 0),$
$24 \cdot 1^2 + f_1(3 - 1), 24 \cdot 2^2 + f_1(3 - 2), 24 \cdot 3^2 + f_1(3 - 3)\} = \min\{0 + 90, 24 + 40, 96 + 10,$
$216 + 0\} = 64;$

$f_2(4) = \min\limits_{\substack{0 \le x_2 \le \min\{4,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(4 - x_2)\right] = \min\limits_{0 \le x_2 \le 4} \left[24 \cdot x_2^2 + f_1(4 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(4 - 0),$
$24 \cdot 1^2 + f_1(4 - 1), 24 \cdot 2^2 + f_1(4 - 2), 24 \cdot 3^2 + f_1(4 - 3), 24 \cdot 4^2 + f_1(4 - 4)\} = \min\{0 + 160,$
$24 + 90, 96 + 40, 216 + 10, 384 + 0\} = 114;$

$f_2(5) = \min\limits_{\substack{0 \le x_2 \le \min\{5,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(5 - x_2)\right] = \min\limits_{0 \le x_2 \le 0} \left[24 \cdot x_2^2 + f_1(5 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(5 - 0),$
$24 \cdot 1^2 + f_1(5 - 1), 24 \cdot 2^2 + f_1(5 - 2), 24 \cdot 3^2 + f_1(5 - 3), 24 \cdot 4^2 + f_1(5 - 4), 24 \cdot 5^2 + f_1(5 - 5)\}$
$= \min\{0 + 250, 24 + 160, 96 + 90, 216 + 40, 384 + 10, 600 + 0\} = 184;$

$f_2(6) = \min\limits_{\substack{0 \le x_2 \le \min\{6,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(6 - x_2)\right] = \min\limits_{0 \le x_2 \le 6} \left[24 \cdot x_2^2 + f_1(6 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(6 - 0),$
$24 \cdot 1^2 + f_1(6 - 1), 24 \cdot 2^2 + f_1(6 - 2), 24 \cdot 3^2 + f_1(6 - 3), 24 \cdot 4^2 + f_1(6 - 4), 24 \cdot 5^2 + f_1(6 - 5),$
$24 \cdot 6^2 + f_1(6 - 6)\} = \min\{0 + 360, 24 + 250, 96 + 160, 216 + 90, 384 + 40, 600 + 10, 864 +$
$+ 0\} = 256.$

U nastavku pri izračunima $f_2(7), f_2(8), \ldots, f_2(12)$ koristimo rezultat da $f_1(7), f_1(8), \ldots, f_1(12)$ ne postoje jer se u prvom mjesecu ne može proizvesti više od 6 komada proizvoda.

$f_2(7) = \min\limits_{\substack{0 \le x_2 \le \min\{7,6\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(7 - x_2)\right] = \min\limits_{0 \le x_2 \le 6} \left[24 \cdot x_2^2 + f_1(7 - x_2)\right] = \min\{24 \cdot 0^2 + f_1(7 - 0),$
$24 \cdot 1^2 + f_1(7 - 1), 24 \cdot 2^2 + f_1(7 - 2), 24 \cdot 3^2 + f_1(7 - 3), 24 \cdot 4^2 + f_1(7 - 4), 24 \cdot 5^2 + f_1(7 - 5),$
$24 \cdot 6^2 + f_1(7 - 6)\} = \min\{24 \cdot 1^2 + f_1(6), 24 \cdot 2^2 + f_1(5), 24 \cdot 3^2 + f_1(4), 24 \cdot 4^2 + f_1(3), 24 \cdot 5^2$
$+ f_1(2), 24 \cdot 6^2 + f_1(1)\} = \min\{384, 346, 376, 474, 640, 874\} = 346;$

$f_2(8) = \min\limits_{\substack{0\le x_2\le \min\{8,6\}\\ x_2\in\mathbb{Z}}}\left[24\cdot x_2^2 + f_1(8-x_2)\right] = \min\limits_{0\le x_2\le 6}\left[24\cdot x_2^2 + f_1(8-x_2)\right] = \min\{24\cdot 0^2 + f_1(8-0),$ $24\cdot 1^2 + f_1(8-1), 24\cdot 2^2 + f_1(8-2), 24\cdot 3^2 + f_1(8-3), 24\cdot 4^2 + f_1(8-4), 24\cdot 5^2 + f_1(8-5),$ $24\cdot 6^2 + f_1(8-6)\} = \min\{24\cdot 2^2 + f_1(6), 24\cdot 3^2 + f_1(5), 24\cdot 4^2 + f_1(4), 24\cdot 5^2 + f_1(3), 24\cdot 6^2 + f_1(2)\} = \min\{456, 466, 544, 690, 904\} = 456;$

$f_2(9) = \min\limits_{\substack{0\le x_2\le \min\{9,6\}\\ x_2\in\mathbb{Z}}}\left[24\cdot x_2^2 + f_1(9-x_2)\right] = \min\limits_{0\le x_2\le 6}\left[24\cdot x_2^2 + f_1(9-x_2)\right] = \min\{24\cdot 0^2 + f_1(9-0),$ $24\cdot 1^2 + f_1(9-1), 24\cdot 2^2 + f_1(9-2), 24\cdot 3^2 + f_1(9-3), 24\cdot 4^2 + f_1(9-4), 24\cdot 5^2 + f_1(9-5),$ $24\cdot 6^2 + f_1(9-6)\} = \min\{24\cdot 3^2 + f_1(6), 24\cdot 4^2 + f_1(5), 24\cdot 5^2 + f_1(4), 24\cdot 6^2 + f_1(3)\} =$ $= \min\{576, 634, 760, 954\} = 576;$

$f_2(10) = \min\limits_{\substack{0\le x_2\le \min\{10,6\}\\ x_2\in\mathbb{Z}}}\left[24\cdot x_2^2 + f_1(10-x_2)\right] = \min\limits_{0\le x_2\le 6}\left[24\cdot x_2^2 + f_1(10-x_2)\right] = \min\{24\cdot 0^2 + f_1(10-$ $-0), 24\cdot 1^2 + f_1(10-1), 24\cdot 2^2 + f_1(10-2), 24\cdot 3^2 + f_1(10-3), 24\cdot 4^2 + f_1(10-4), 24\cdot 5^2 +$ $+ f_1(10-5), 24\cdot 6^2 + f_1(10-6)\} = \min\{24\cdot 4^2 + f_1(6), 24\cdot 5^2 + f_1(5), 24\cdot 6^2 + f_1(4)\} =$ $= \min\{744, 850, 1024\} = 744;$

$f_2(11) = \min\limits_{\substack{0\le x_2\le \min\{11,6\}\\ x_2\in\mathbb{Z}}}\left[24\cdot x_2^2 + f_1(11-x_2)\right] = \min\limits_{0\le x_2\le 6}\left[24\cdot x_2^2 + f_1(11-x_2)\right] = \min\{24\cdot 0^2 + f_1(11-$ $-0), 24\cdot 1^2 + f_1(11-1), 24\cdot 2^2 + f_1(11-2), 24\cdot 3^2 + f_1(11-3), 24\cdot 4^2 + f_1(11-4), 24\cdot 5^2 +$ $+ f_1(11-5), 24\cdot 6^2 + f_1(11-6)\} = \min\{24\cdot 5^2 + f_1(6), 24\cdot 6^2 + f_1(5)\} = \min\{960, 1114\} =$ $= 960;$

$f_2(12) = \min\limits_{\substack{0\le x_2\le \min\{12,6\}\\ x_2\in\mathbb{Z}}}\left[24\cdot x_2^2 + f_1(12-x_2)\right] = \min\limits_{0\le x_2\le 6}\left[24\cdot x_2^2 + f_1(12-x_2)\right] = \min\{24\cdot 0^2 + f_1(12-$ $-0), 24\cdot 1^2 + f_1(12-1), 24\cdot 2^2 + f_1(12-2), 24\cdot 3^2 + f_1(12-3), 24\cdot 4^2 + f_1(12-4), 24\cdot 5^2 +$ $+ f_1(12-5), 24\cdot 6^2 + f_1(12-6)\} = \min\{24\cdot 6^2 + f_1(6)\} = 1224.$

Tako dobivamo sljedeću tablicu:

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*
0	0	0	0	0
1	10	1	10	0
2	40	2	34	1
3	90	3	64	1
4	160	4	114	1
5	250	5	184	1
6	360	6	256	2
7	∞	6	346	2
8	∞	6	456	2
9	∞	6	576	3
10	∞	6	744	4
11	∞	6	960	5
12	∞	6	1224	6

Vrijednost x_2^* označava optimalnu proizvodnju u drugom mjesecu za koju se postiže pripadna optimalna vrijednost $f_2(s)$, a lagano se "očitava" iz odgovarajućega izračuna.

U *trećoj fazi* rješavanja problema promatramo realnu funkciju:

$$f_3(s) = \min_{\substack{0 \le x_3 \le \min\{s,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(s - x_3) \right].$$

U ovoj fazi pretpostavljamo da se proizvodnja ukupno s komada proizvoda odvija tijekom sva tri mjeseca. Ukoliko se u trećem mjesecu proizvede x_3 proizvoda, u prva dva mjeseca mora se proizvesti ukupno $s - x_3$ proizvoda. Ukupni troškovi proizvodnje u trećem mjesecu iznose $32 \cdot x_3^2$, dok *optimalni* troškovi proizvodnje u prva dva mjeseca, sukladno definiciji funkcije f_2, iznose $f_2(s - x_3)$ n.j. Stoga ukupni troškovi proizvodnje u sva tri mjeseca iznose $32 \cdot x_3^2 + f_2(s - x_3)$ n.j, pa taj zbroj treba minimizirati uvažavajući početne uvjete na vrijednost varijable x_3 (cijeli broj ne veći od ukupne količine proizvoda koju treba proizvesti i najvećega mjesečnoga kapaciteta proizvodnje). Računamo redom:

$f_3(0) = \min_{\substack{0 \le x_3 \le \min\{0,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(0 - x_3) \right] = \min_{\substack{0 \le x_3 \le 0 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(0 - x_3) \right] = 32 \cdot 0^2 + f_2(0 - 0) = 0 + 0 = 0;$

$f_3(1) = \min_{\substack{0 \le x_3 \le \min\{1,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(1 - x_3) \right] = \min_{\substack{0 \le x_3 \le 1 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(1 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(1 - 0), 32 \cdot 1^2 + f_2(1 - 1)\} = \min \{0 + 10, 32 + 0\} = 10;$

$f_3(2) = \min_{\substack{0 \le x_3 \le \min\{2,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(2 - x_3) \right] = \min_{\substack{0 \le x_3 \le 2 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(2 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(2 - 0), 32 \cdot 1^2 + f_2(2 - 1), 32 \cdot 2^2 + f_2(2 - 2)\} = \min \{34, 42, 128\} = 34;$

$f_3(3) = \min_{\substack{0 \le x_3 \le \min\{3,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(3 - x_3) \right] = \min_{\substack{0 \le x_3 \le 3 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(3 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(3 - 0), 32 \cdot 1^2 + f_2(3 - 1), 32 \cdot 2^2 + f_2(3 - 2), 32 \cdot 3^2 + f_2(3 - 3)\} = \min \{64, 66, 138, 288\} = 64;$

$f_3(4) = \min_{\substack{0 \le x_3 \le \min\{4,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(4 - x_3) \right] = \min_{\substack{0 \le x_3 \le 4 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(4 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(4 - 0), 32 \cdot 1^2 + f_2(4 - 1), 32 \cdot 2^2 + f_2(4 - 2), 32 \cdot 3^2 + f_2(4 - 3), 32 \cdot 3^2 + f_2(4 - 4)\} = \min \{114, 96, 162, 298, 512\} = 96;$

$f_3(5) = \min_{\substack{0 \le x_3 \le \min\{5,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(5 - x_3) \right] = \min_{\substack{0 \le x_3 \le 5 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(5 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(5 - 0), 32 \cdot 1^2 + f_2(5 - 1), 32 \cdot 2^2 + f_2(5 - 2), 32 \cdot 3^2 + f_2(5 - 3), 32 \cdot 4^2 + f_2(5 - 4), 32 \cdot 5^2 + f_2(5 - 5)\} = \min \{184, 146, 192, 322, 522, 800\} = 146;$

$f_3(6) = \min_{\substack{0 \le x_3 \le \min\{6,6\} \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(6 - x_3) \right] = \min_{\substack{0 \le x_3 \le 6 \\ x_3 \in \mathbb{Z}}} \left[32 \cdot x_3^2 + f_2(6 - x_3) \right] = \min \{32 \cdot 0^2 + f_2(6 - 0), 32 \cdot 1^2 + f_2(6 - 1), 32 \cdot 2^2 + f_2(6 - 2), 32 \cdot 3^2 + f_2(6 - 3), 32 \cdot 4^2 + f_2(6 - 4), 32 \cdot 5^2 + f_2(6 - 5), 32 \cdot 6^2 + f_2(6 - 6)\} = \min \{256, 216, 242, 352, 546, 810, 1152\} = 216;$

$f_3(7) = \min\limits_{\substack{0\le x_3\le \min\{7,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(7-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(7-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(7-0),$
$32\cdot 1^2 + f_2(7-1),\ 32\cdot 2^2 + f_2(7-2),\ 32\cdot 3^2 + f_2(7-3),\ 32\cdot 4^2 + f_2(7-4),\ 32\cdot 5^2 + f_2(7-5),$
$32\cdot 6^2 + f_2(7-6)\} = \min\{346, 288, 312, 402, 576, 834, 1162\} = 288;$

$f_3(8) = \min\limits_{\substack{0\le x_3\le \min\{8,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(8-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(8-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(8-0),$
$32\cdot 1^2 + f_2(8-1),\ 32\cdot 2^2 + f_2(8-2),\ 32\cdot 3^2 + f_2(8-3),\ 32\cdot 4^2 + f_2(8-4),\ 32\cdot 5^2 + f_2(8-5),$
$32\cdot 6^2 + f_2(8-6)\} = \min\{456, 378, 384, 472, 626, 864, 1186\} = 378;$

$f_3(9) = \min\limits_{\substack{0\le x_3\le \min\{9,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(9-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(9-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(9-0),$
$32\cdot 1^2 + f_2(9-1),\ 32\cdot 2^2 + f_2(9-2),\ 32\cdot 3^2 + f_2(9-3),\ 32\cdot 4^2 + f_2(9-4),\ 32\cdot 5^2 + f_2(9-5),$
$32\cdot 6^2 + f_2(9-6)\} = \min\{576, 488, 474, 544, 696, 914, 1216\} = 474;$

$f_3(10) = \min\limits_{\substack{0\le x_3\le \min\{10,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(10-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(10-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(10-$
$-0),\ 32\cdot 1^2 + f_2(10-1),\ 32\cdot 2^2 + f_2(10-2),\ 32\cdot 3^2 + f_2(10-3),\ 32\cdot 4^2 + f_2(10-4),\ 32\cdot 5^2 +$
$+ f_2(10-5),\ 32\cdot 6^2 + f_2(10-6)\} = \min\{744, 608, 584, 634, 768, 984, 1266\} = 584;$

$f_3(11) = \min\limits_{\substack{0\le x_3\le \min\{11,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(1-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(11-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(11-0),$
$32\cdot 1^2 + f_2(11-1),\ 32\cdot 2^2 + f_2(11-2),\ 32\cdot 3^2 + f_2(11-3),\ 32\cdot 4^2 + f_2(11-4),\ 32\cdot 5^2 +$
$+ f_2(11-5),\ 32\cdot 6^2 + f_2(11-6)\} = \min\{960, 776, 704, 744, 858, 1056, 1336\} = 704;$

$f_3(12) = \min\limits_{\substack{0\le x_3\le \min\{12,6\}\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(12-x_3)\right] = \min\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[32\cdot x_3^2 + f_2(12-x_3)\right] = \min\ \{32\cdot 0^2 + f_2(12-$
$-0),\ 32\cdot 1^2 + f_2(12-1),\ 32\cdot 2^2 + f_2(12-2),\ 32\cdot 3^2 + f_2(12-3),\ 32\cdot 4^2 + f_2(12-4),\ 32\cdot 5^2 +$
$+ f_2(12-5),\ 32\cdot 6^2 + f_2(12-6)\} = \min\{1224, 992, 872, 864, 968, 1146, 1408\} = 864.$

Tako dobivamo sljedeću tablicu:

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*	$f_3(s)$	x_3^*
0	0	0	0	0	0	0
1	10	1	10	0	10	0
2	40	2	34	1	34	0
3	90	3	64	1	64	0
4	160	4	114	1	96	1
5	250	5	184	1	146	1
6	360	6	256	2	216	1
7	∞	6	346	2	288	1
8	∞	6	456	2	378	1
9	∞	6	576	3	474	2
10	∞	6	744	4	584	2
11	∞	6	960	5	704	2
12	∞	6	1224	6	864	3

Vrijednost x_3^* označava optimalnu proizvodnju u trećem mjesecu za koju se postiže pripadna optimalna vrijednost $f_3(s)$, a lagano se "očitava" iz odgovarajućega izračuna.

Iz tablice nam preostaje unatrag "pročitati" traženo rješenje. Za s = 12 minimalni ukupni troškovi proizvodnje iznose 864 n.j. Ti se troškovi postižu proizvodnjom točno x_3^* = 3 komada proizvoda u trećem mjesecu, te 12 – 3 = 9 proizvoda u prva dva mjeseca. Za s = 9 odgovarajuća optimalna količina proizvodnje u drugom mjesecu je x_2^* = 3, pa u drugom mjesecu treba proizvesti točno 3 proizvoda. Stoga u prvom mjesecu treba proizvesti točno 12 – (3 + 3) = 6 proizvoda.

Dakle, traženi optimalni plan proizvodnje je:

mjesec	*optimalan broj proizvoda*
1.	6
2.	3
3.	3

a pripadni optimalni troškovi iznose $f_3(12)$ = 864 n.j.

<u>*Rješenje pomoću programa Winqsb*</u>: U ovome ćemo primjeru postupiti analogno kao u Primjeru 1., ali uz jednu modifikaciju. Naime, problem naprtnjače je *maksimizacijski* problem, tj. traži se maksimizacija vrijednosti svih namirnica koje stavljamo u naprtnjaču. Stoga potprogram *DP* prigodom rješavanja problema naprtnjače traži *maksimum* funkcije cilja. No, problem koji ovdje promatramo je *minimizacijski* problem jer zahtijevamo minimizacija ukupnih troškova. Na prvi se pogled čini da ta dva različita optimizacijska problema ne možemo riješiti pomoću potprograma *DP*. Nasreću, takav je zaključak pogrešan jer nas "spašava" standardna veza između problema minimizacije i problema maksimizacije:

$$\text{minimizirati } f(x) = \text{maksimizirati } (-f(x)),$$

gdje je $f(x)$ funkcija cilja. Koristeći ovu jednakost, polazni matematički model možemo zapisati u obliku:

$$\text{maksimizirati } f(x_1, x_2, x_3) = -10 \cdot x_1^2 - 24 \cdot x_2^2 - 32 \cdot x_3^2$$

pod uvjetima

$$x_1 + x_2 + x_3 = 12,$$
$$x_1, x_2, x_3 \in [6]_0.$$

<u>*Oprez*</u>: Promjena predznaka odnosi se <u>isključivo</u> na funkciju cilja, ali nikako i na postavljene uvjete!

Ovaj problem možemo riješiti potpuno analogno kao i zatvoreni problem naprtnjače, pri čemu modificiranu funkciju cilja npr. možemo shvatiti kao ukupne troškove prijevoza svih vrsta namirnica (svaki negativni predznak uobičajeno ekonomski interpretiramo kao trošak).

Dakle, pokrenimo potprogram *DP*, nazovimo problem koji ćemo rješavati *Primjer* 3 i, kao ukupan broj predmeta, ponovno upišimo 3. Potom lijevom tipkom miša kliknimo na *OK*.

U sljedećem koraku unesimo ulazne podatke.

U drugi stupac (*Item Identification*) upišimo M1, M2 i M3 jer će nam te oznake na kraju sugerirati na koji se mjesec (prvi, drugi ili treći) odnosi odgovarajući rezultat.
U treći stupac (*Units Available*) upišimo najveće moguće kapacitete mjesečne proizvodnje. Sva tri kapaciteta su jednaka 6, pa u stupac unosimo tri "šestice"[17].
U četvrti stupac (*Unit Capacity Required*), kako smo već istakli u rješenju Primjera 1., upisujemo tri "jedinice".
U peti stupac (*Return Function* (*X*: *Item ID*)) upisujemo pojedine komponente *modificirane* funkcije cilja, tj. funkcije cilja koja se odnosi na odgovarajući problem maksimizacije. Ta funkcija je $f(x_1, x_2, x_3) = -10 \cdot x_1^2 - 24 \cdot x_2^2 - 32 \cdot x_3^2$, pa u peti stupac upisujemo redom: –10*X^2, -24*X^2 i -32*X^2 (analogno kao i pri unosu npr. u MS Excelu).
U posljednji redak tablice (*Capacity*=) upišimo ukupan broj komada koje treba proizvesti, tj. 12.

Time je unos ulaznih podataka završen. Lijevom tipkom miša kliknimo na izbornik *Solve and Analyze*, a potom na opciju *Solve the Problem*. Dobivamo tablicu u kojoj je navedeno rješenje promatranoga problema.

Spomenuto rješenje očitavamo u trećem stupcu tablice (*Decision Quantity*). Iz toga stupca proizlazi da u prvom mjesecu treba proizvesti 6 proizvoda, a u drugom i trećem po 3 proizvoda. Ukupne minimalne troškove proizvodnje "očitamo" iz posljednjega retka (*Total Return Value*): ondje je naveden broj –864. Predznak –, kako smo istakli, označava da se radi o troškovima, pa zaključujemo da su optimalni ukupni troškovi 864 n.j.

Primjer 4. Riješimo prethodni primjer uz dodatan uvjet da se u prvom mjesecu, zbog planiranoga remonta nekih strojeva, može proizvesti najviše 4 komada proizvoda.

Uz iste oznake kao u Primjeru 3., odgovarajući *matematički model* je:

$$\text{minimizirati } f(x_1, x_2, x_3) = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + w_3 \cdot x_3^2$$

pod uvjetima

$$x_1 + x_2 + x_3 = 12,$$
$$x_1 \in [4]_0,$$
$$x_2, x_3 \in [6]_0,$$

odnosno za $w_1 = 10$, $w_2 = 24$ i $w_3 = 32$

$$\text{minimizirati } f(x_1, x_2, x_3) = 10 \cdot x_1^2 + 24 \cdot x_2^2 + 32 \cdot x_3^2$$

pod uvjetima

$$x_1 + x_2 + x_3 = 12,$$
$$x_1 \in [4]_0,$$
$$x_2, x_3 \in [6]_0.$$

[17] Ovaj unos je sasvim slučajan i nema nikakve veze s đavoljim kultovima ili nekim njima srodnim više-manje paranormalnim pojavama.

"*Klasično*" *rješenje* je potpuno analogno onome iz Primjera 3., s tim da su vrijednosti funkcije $f_1(s)$ dogovorno jednake ∞ za svaki $s \in S$ takav da je $s \geq 5$. Dobiva se sljedeći optimalan plan proizvodnje (provjerite!):

mjesec	*optimalan broj proizvoda*
1.	4
2.	5
3.	3

Pripadni optimalni ukupni troškovi proizvodnje iznose 1 048 n.j.

Rješenje pomoću programa Winqsb: U rješenju prethodnoga primjera u treći stupac treba upisati brojeve 4, 6 i 6, a svi ostali ulazni podaci ostaju nepromijenjeni. Pokretanjem procedure *Solve and Analyze* iz stupca *Decision Quantity* "očitavamo" optimalan plan proizvodnje: *u prvom mjesecu proizvesti* 4 *proizvoda*, *u drugom* 5, *a u trećem* 3 *proizvoda*. Pripadni optimalni ukupni troškovi (*Total Return Value*) iznose 1 048 n.j.

Primjer 5. U tijeku jednoga mjeseca (4 tjedna) treba proizvesti ukupno 11 komada nekoga proizvoda. Troškovi proizvodnje x_i komada proizvoda u i – tom tjednu zadani su izrazom $w_i \cdot x_i^2$. Tijekom jednoga tjedna moguće je proizvesti najviše 5 komada proizvoda. Treba izraditi plan proizvodnje tako da njezini ukupni troškovi budu minimalni.

a) Odredimo matematički model promatranoga problema.
b) Riješimo model za $w_1 = 12$, $w_2 = 24$, $w_3 = 40$ i $w_4 = 30$.

Matematički model: Ukupne troškove proizvodnje računamo prema formuli

$$f(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} w_i \cdot x_i^2 = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + w_3 \cdot x_3^2 + w_4 \cdot x_4^2 .$$

Ukupan broj proizvedenih komada mora biti jednak 11, što znači da mora vrijediti jednakost

$$\sum_{i=1}^{4} x_i = x_1 + x_2 + x_3 + x_4 = 11.$$

Budući da se tijekom jednoga tjedna može proizvesti najviše 5 komada proizvoda i da ukupan broj proizvedenih komada nužno mora biti prirodan broj ili nula, za svaki $i \in [4]$ vrijedi uvjet:

$$x_i \in [5]_0.$$

Stoga odgovarajući *matematički model* glasi:

$$\text{minimizirati } f(x_1, x_2, x_3, x_4) = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + w_3 \cdot x_3^2 + w_4 \cdot x_4^2$$

pod uvjetima

$$x_1 + x_2 + x_3 + x_4 = 11;$$
$$x_1, x_2, x_3, x_4 \in [5]_0.$$

b) U konkretnom je slučaju matematički model sljedeći:

$$\text{minimizirati } f(x_1, x_2, x_3, x_4) = 12 \cdot x_1^2 + 24 \cdot x_2^2 + 40 \cdot x_3^2 + 30 \cdot x_4^2$$

pod uvjetima

$$x_1 + x_2 + x_3 + x_4 = 11;$$
$$x_1, x_2, x_3, x_4 \in [5]_0.$$

"*Klasično" rješenje*: I ovaj model rješavamo potpuno analogno kao i prethodni. Stavimo $S = [11]_0$, pa za svaki $s \in S$ računamo vrijednosti sljedećih funkcija:

$$f_1(s) = 12 \cdot s^2;$$
$$f_2(s) = \min_{\substack{0 \le x_2 \le \min\{s,5\} \\ x_2 \in \mathbb{Z}}} \left[24 \cdot x_2^2 + f_1(s - x_2) \right];$$
$$f_3(s) = \min_{\substack{0 \le x_3 \le \min\{s,5\} \\ x_3 \in \mathbb{Z}}} \left[40 \cdot x_3^2 + f_2(s - x_3) \right];$$
$$f_4(s) = \min_{\substack{0 \le x_4 \le \min\{s,5\} \\ x_4 \in \mathbb{Z}}} \left[30 \cdot x_4^2 + f_3(s - x_4) \right].$$

Pri izračunu vrijednosti ovih funkcija treba imati na umu da zbog polaznih uvjeta *ne* postoje sljedeće vrijednosti:

- $f_1(6), f_1(7), \ldots, f_1(11)$ jer se u prvom (ali i *bilo kojem*!) tjednu može proizvesti najviše 5 proizvoda;
- $f_2(11)$ jer se u prva (ali i *bilo koja*!) dva tjedna može proizvesti najviše $2 \cdot 5 = 10$ proizvoda.

Analogno kao u prethodnim primjerima dobiva se sljedeća tablica (provjerite!):

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*	$f_3(s)$	x_3^*	$f_4(s)$	x_4^*
0	0	0	0	0	0	0	0	0
1	12	1	12	0	12	0	12	0
2	48	2	36	1	36	0	36	0
3	108	3	72	1	72	0	66	1
4	192	4	132	1	112	1	102	1
5	300	5	204	2	172	1	142	1
6	∞	5	288	2	244	1	202	1
7	∞	5	396	2	328	1	274	1
8	∞	5	516	3	436	1	358	1
9	∞	5	684	4	556	1, 2	448	2
10	∞	5	900	5	676	2	556	2
11	∞	5	∞	5	844	2	676	2

(Zapis 1, 2 označava da se optimalna vrijednost odgovarajuće funkcije postiže za $x_3 = 1$ **i** $x_3 = 2$.)

Preostaje nam unatrag "iščitati" rješenje postavljenoga problema. Optimalna minimalna vrijednost ukupnih troškova proizvodnje 11 komada proizvoda jednaka je $f_4(11) = 676$ n.j. Ta se vrijednost postiže za $x_4^* = 2$, što znači da u četvrtom tjednu treba proizvesti točno dva proizvoda. Stoga u prva tri tjedna treba proizvesti ukupno 11 – 2 = 9 proizvoda. Optimalni minimalni troškovi proizvodnje 9 proizvoda u prva tri tjedna iznose $f_3(9) = 556$ n.j. i postižu se za $x_3^* = 1$ ili $x_3^* = 2$. Zbog toga razlikujemo dvije mogućnosti:

1) U trećem tjednu proizvede se točno jedan proizvod (tj. $x_3^* = 1$). Tada u prva dva tjedna treba proizvesti ukupno 9 – 1 = 8 proizvoda. Optimalni minimalni troškovi proizvodnje 8 proizvoda u prva dva tjedna iznose $f_2(8) = 516$ n.j. i postižu se za $x_2^* = 3$. Zbog toga u drugom tjednu treba proizvesti ukupno 3 proizvoda, pa u prvom tjednu preostaje proizvesti ukupno 8 – 3 = 5 proizvoda.

2.) U trećem tjednu proizvedu se točno dva proizvoda (tj. $x_3^* = 2$). Tada u prva dva tjedna treba proizvesti ukupno 9 – 2 = 7 proizvoda. Optimalni minimalni troškovi proizvodnje 7 proizvoda u prva dva tjedna iznose $f_2(7) = 396$ n.j. i postižu se za $x_2^* = 2$. Stoga u drugom tjednu treba proizvesti ukupno 2 proizvoda, pa u prvom tjednu preostaje proizvesti ukupno 7 – 2 = 5 proizvoda.

Tako smo kao rješenje dobili ukupno dva optimalna plana proizvodnje:

Plan 1:

tjedan	1	2	3	4
optimalna proizvodnja [kom.]	5	3	1	2

Plan 2:

tjedan	1	2	3	4
optimalna proizvodnja [kom.]	5	2	2	2

Rješenje pomoću programa Winqsb: Na ovom ćemo primjeru uočiti glavni nedostatak potprograma *DP* programa *Winqsb*, a to je nemogućnost ispisa barem dvaju optimalnih rješenja (ukoliko ona postoje, naravno). Naime, potprogram će – kao izlazni rezultat – ispisati točno jedan optimalni plan i *ne* postoji nikakva mogućnost da nekom od implementiranih procedura dobijemo ispis drugoga optimalnoga plana.

Sâm postupak rješavanja problema pomoću programa *Winqsb* je potpuno analogan onome iz prethodnoga primjera. Najprije problem minimizacije "pretvorimo" u problem maksimizacije:

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = -12 \cdot x_1^2 - 24 \cdot x_2^2 - 40 \cdot x_3^2 - 30 \cdot x_4^2$$

pod uvjetima

$$x_1 + x_2 + x_3 + x_4 = 11;$$

$$x_1, x_2, x_3, x_4 \in [5]_0.$$

Potom pokrenemo potprogram *DP*, nazovimo problem koji ćemo rješavati *Primjer* 5, a u pravokutnik pored natpisa *Number of Items* upišimo 4. Potom kliknimo na *OK*.

Unesimo ulazne podatke za razmatrani primjer na sljedeći način:

- u drugi stupac (*Item Identification*) upišimo: T1, T2, T3 i T4 jer ćemo tako na kraju odmah moći "očitati" opseg proizvodnje u svakom pojedinom tjednu;
- u treći stupac (*Units Available*) upišimo maksimalne tjedne kapacitete proizvodnje: oni su jednaki 5, pa upišimo ukupno četiri "petice";
- u četvrti stupac (*Unit Capacity Required*) već standardno upišimo onoliko "jedinica" koliko imamo namirnica, odnosno, u ovom slučaju, tjedana: dakle, upišimo četiri "jedinice";
- u peti stupac (*Return Function* (*X*: *Item ID*)) upišimo komponente modificirane funkcije cilja koje se odnose na svaki pojedini tjedan: dakle, upišimo redom –12*X^2, –24*X^2, –40*X^2 i –30*X^2 (na isti način kao što to činimo npr. u MS Excelu);
- u posljednji redak (*Capacity*=) upišimo ukupan broj komada proizvoda koje treba proizvesti, tj. 11.

Ovime je unos ulaznih podataka završen. Lijevom tipkom miša kliknimo na izbornik *Solve and Analyze* i odaberimo opciju *Solve the Problem*. Kao rješenje razmatranoga problema dobivamo *Plan* 2 iz "klasičnoga" rješenja, tj. u prvom tjednu treba proizvesti 5 komada proizvoda, a u svakom od preostala tri tjedna po dva komada proizvoda. Vrijednost *Total Return Value* = –676 znači da ukupni minimalni *troškovi* iznose 676 n.j. "Klasičnim" rješenjem dobiven *Plan* 1 nije moguće dobiti pomoću programa *DP*[18].

Primjer 6. Tvrtka raspolaže s kapitalom od ukupno 6 n.j. koje može uložiti u točno tri vrste djelatnosti: D_1, D_2 i D_3. Očekivana dobit u pojedinoj djelatnosti ovisno o iznosu uloga navedena je u donjoj tablici.

ulog [n.j.]	0	1	2	3	4	5	6
očekivana neto–dobit od D_1	0	4	26	40	45	50	51
očekivana neto–dobit od D_2	0	5	15	40	80	90	95
očekivana neto–dobit od D_3	0	5	15	40	60	70	73

Treba napraviti optimalan plan ulaganja tako da iznos uloga u svaku djelatnost bude cjelobrojan, a očekivana neto–dobit maksimalna. Odredimo matematički model promatranoga problema, pa pomoću njega nađimo optimalan plan ulaganja.

<u>*Matematički model*</u>: Za svaki $i \in [3]$ označimo s x_i iznos ulaganja u djelatnost D_i, a s $f_i(x_i)$ očekivanu neto–dobit nastalu tim ulaganjem. Drugim riječima, funkcija $f_1(x_1)$ određuje očekivanu neto–dobit nastalu ulaganjem iznosa x_1 u djelatnost D_1, funkcija $f_2(x_2)$ očekivanu neto–dobit nastalu ulaganjem iznosa x_2 u djelatnost D_2, a funkcija $f_3(x_3)$ očekivanu neto–dobit nastalu ulaganjem iznosa x_3 u djelatnost D_3. Stoga je ukupna očekivana neto–dobit nastala ulaganjem svih triju iznosa jednaka

$$f(x_1, x_2, x_3) = f_1(x_1) + f_2(x_2) + f_3(x_3).$$

[18] Osim u superposebnim slučajevima primjene *što–ako* analize kao što je npr. dodatni zahtjev da se u trećem tjednu, zbog kolektivnoga godišnjeg odmora, može proizvesti točno jedan proizvod.

Nadalje, ukupan zbroj uloga u navedene djelatnosti ne može biti strogo veći od raspoloživoga iznosa, pa mora vrijediti nejednakost

$$x_1 + x_2 + x_3 \leq 6.$$

Prema uvjetima zadatka, svi uloženi iznosi nužno moraju biti ili prirodni brojevi ne veći od 6 ili nula, pa vrijedi skupovna relacija

$$x_1, x_2, x_3 \in [6]_0.$$

Prema tome, traženi *matematički model* opisanoga problema glasi:

$$\text{maksimizirati } f(x_1, x_2, x_3) = f_1(x_1) + f_2(x_2) + f_3(x_3)$$

pod uvjetima

$$x_1 + x_2 + x_3 \leq 6$$
$$x_1, x_2, x_3 \in [6]_0.$$

<u>*Rješenje*</u>: Odmah istaknimo da rješenje ovoga problema pomoću programa *Winqsb* **nije** moguće jer taj program zahtijeva da funkcija cilja bude definirana analitičkim izrazom, a ne tablično. Stoga je jedino moguće rješenje promatranoga problema "klasično" rješenje. Stavimo $C = [6]_0$, pa definirajmo realne funkcije:

$$d_1(c) = f_1(c),$$
$$d_2(c) = \max_{\substack{0 \leq x_2 \leq \min\{c,6\} \\ x_2 \in \mathbb{Z}}} \left[f_2(x_2) + d_1(c - x_2) \right],$$
$$d_3(c) = \max_{\substack{0 \leq x_3 \leq \min\{c,6\} \\ x_3 \in \mathbb{Z}}} \left[f_3(x_3) + d_2(c - x_3) \right].$$

Funkcija $d_1(c)$ maksimizira očekivanu neto–dobit nastalu ulaganjem c n.j. isključivo u djelatnost D_1, funkcija $d_2(c)$ maksimizira očekivanu neto–dobit nastalu ulaganjem c n.j. isključivo u djelatnosti D_1 i D_2, dok funkcija $d_3(c)$ maksimizira očekivanu neto–dobit nastalu ulaganjem c n.j. u sve tri djelatnosti. Intuitivno možemo naslutiti da će tražena optimalna očekivana neto–dobit biti jednaka vrijednosti $f_3(6)$.

Vrijednosti funkcije $d_1(c)$ već su naznačene u drugom retku početne tablice (redak *očekivana neto–dobit od* D_1), pa ih samo prepišimo na sljedeći način:

c	$d_1(c)$	x_1^*
0	0	0
1	4	1
2	26	2
3	40	3
4	45	4
5	50	5
6	51	6

U nastavku za svaki $c \in C$ računamo vrijednosti funkcije $d_2(s)$:

$d_2(0) = \max\limits_{\substack{0\le x_2\le \min\{0,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(0-x_2)\right] = \max\limits_{\substack{0\le x_2\le 0\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(0-x_2)\right] = f_2(0) + d_1(0) = 0 + 0 = 0;$

$d_2(1) = \max\limits_{\substack{0\le x_2\le \min\{1,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(1-x_2)\right] = \max\limits_{\substack{0\le x_2\le 1\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(1-x_2)\right] = \max\ \{f_2(0) + d_1(1 - 0), f_2(1) + d_1(1 - 1)\} = \max\ \{0 + 4, 5 + 0\} = 5;$

$d_2(2) = \max\limits_{\substack{0\le x_2\le \min\{2,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(2-x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(2-x_2)\right] \max\ \{f_2(0) + d_1(2 - 0), f_2(1) + d_1(2 - 1), f_2(2) + d_1(2 - 2)\} = \max\ \{0 + 26, 5 + 4, 15 + 0\} = 26;$

$d_2(3) = \max\limits_{\substack{0\le x_2\le \min\{3,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(3-x_2)\right] = \max\limits_{\substack{0\le x_2\le 2\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(3-x_2)\right] = \max\ \{f_2(0) + d_1(3 - 0), f_2(1) + d_1(3 - 1), f_2(2) + d_1(3 - 2), f_2(3) + d_1(3 - 3)\} = \max\ \{0 + 40, 5 + 26, 15 + 4, 40 + 0\} = 40;$

$d_2(4) = \max\limits_{\substack{0\le x_2\le \min\{4,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(4-x_2)\right] = \max\limits_{\substack{0\le x_2\le 4\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(4-x_2)\right] = \max\ \{f_2(0) + d_1(4 - 0), f_2(1) + d_1(4 - 1), f_2(2) + d_1(4 - 2), f_2(3) + d_1(4 - 3), f_2(4) + d_1(4 - 4)\} = \max\ \{0 + 45, 5 + 40, 15 + 26, 40 + 4, 80 + 0\} = 80;$

$d_2(5) = \max\limits_{\substack{0\le x_2\le \min\{5,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(5-x_2)\right] = \max\limits_{\substack{0\le x_2\le 5\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(5-x_2)\right] = \max\ \{f_2(0) + d_1(5 - 0), f_2(1) + d_1(5 - 1), f_2(2) + d_1(5 - 2), f_2(3) + d_1(5 - 3), f_2(4) + d_1(5 - 4), f_2(5) + d_1(5 - 5)\} = \max\ \{0 + 50, 5 + 45, 15 + 40, 40 + 26, 80 + 4, 90 + 0\} = 90;$

$d_2(6) = \max\limits_{\substack{0\le x_2\le \min\{6,6\}\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(6-x_2)\right] = \max\limits_{\substack{0\le x_2\le 6\\ x_2\in\mathbb{Z}}} \left[f_2(x_2)+d_1(6-x_2)\right] = \max\ \{f_2(0) + d_1(6 - 0), f_2(1) + d_1(6 - 1), f_2(2) + d_1(6 - 2), f_2(3) + d_1(6 - 3), f_2(4) + d_1(6 - 4), f_2(5) + d_1(6 - 5), f_2(6) + d_1(6 - 6)\} = \max\ \{0 + 51, 5 + 50, 15 + 45, 40 + 40, 80 + 26, 90 + 4, 95 + 0\} = 106.$

Prethodnu tablicu nadopunjujemo s još dvama stupcima:

c	$d_1(c)$	x_1^*	$d_2(c)$	x_2^*
0	0	0	0	0
1	4	1	5	1
2	26	2	26	0
3	40	3	40	0, 3
4	45	4	80	4
5	50	5	90	5
6	51	6	106	4

Preostaje izračunati vrijednosti funkcije $d_3(c)$ za svaki $c \in C$:

$d_3(0) = \max\limits_{\substack{0\le x_3\le \min\{0,6\}\\ x_3\in\mathbb{Z}}} \left[f_3(x_3)+d_2(0-x_3)\right] = \max\limits_{\substack{0\le x_3\le 0\\ x_3\in\mathbb{Z}}} \left[f_3(x_3)+d_2(0-x_3)\right] = f_3(0) + d_2(0) = 0 + 0 = 0;$

$d_3(1) = \max\limits_{\substack{0\le x_3\le \min\{1,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(1-x_3)\right] = \max\limits_{\substack{0\le x_3\le 1\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(1-x_3)\right]$ = max $\{f_3(0) + d_2(1 - 0), f_3(1) + d_2(1 - 1)\}$ = max $\{0 + 5, 5 + 0\} = 5$;

$d_3(2) = \max\limits_{\substack{0\le x_3\le \min\{2,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(2-x_3)\right] = \max\limits_{\substack{0\le x_3\le 2\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(2-x_3)\right]$ = max $\{f_3(0) + d_2(2 - 0), f_3(1) + d_2(2 - 1), f_3(2) + d_2(2 - 2)\}$ = max $\{0 + 26, 5 + 5, 15 + 0\} = 26$;

$d_3(3) = \max\limits_{\substack{0\le x_3\le \min\{3,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(3-x_3)\right] = \max\limits_{\substack{0\le x_3\le 3\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(3-x_3)\right]$ = max $\{f_3(0) + d_2(3 - 0), f_3(1) + d_2(3 - 1), f_3(2) + d_2(3 - 2), f_3(3) + d_2(3 - 3)\}$ = max $\{0 + 40, 5 + 26, 15 + 5, 40 + 0\}$ = = 40;

$d_3(4) = \max\limits_{\substack{0\le x_3\le \min\{4,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(4-x_3)\right] = \max\limits_{\substack{0\le x_3\le 4\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(4-x_3)\right]$ = max $\{f_3(0) + d_2(4 - 0), f_3(1) + d_2(4 - 1), f_3(2) + d_2(4 - 2), f_3(3) + d_2(4 - 3), f_3(4) + d_2(4 - 4)\}$ = max $\{0 + 80, 5 + 40, 15 + 26, 40 + 5, 60 + 0\} = 80$;

$d_3(5) = \max\limits_{\substack{0\le x_3\le \min\{5,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(5-x_3)\right] = \max\limits_{\substack{0\le x_3\le 5\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(5-x_3)\right]$ = max $\{f_3(0) + d_2(5 - 0), f_3(1) + d_2(5 - 1), f_3(2) + d_2(5 - 2), f_3(3) + d_2(5 - 3), f_3(4) + d_2(5 - 4), f_3(5) + d_2(5 - 5)\}$ = = max $\{0 + 90, 5 + 80, 15 + 40, 40 + 26, 60 + 5, 70 + 0\} = 90$;

$d_3(6) = \max\limits_{\substack{0\le x_3\le \min\{6,6\}\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(6-x_3)\right] = \max\limits_{\substack{0\le x_3\le 6\\ x_3\in\mathbb{Z}}}\left[f_3(x_3)+d_2(6-x_3)\right]$ = max $\{f_3(0) + d_2(6 - 0), f_3(1) + d_2(6 - 1), f_3(2) + d_2(6 - 2), f_3(3) + d_2(6 - 3), f_3(4) + d_2(6 - 4), f_3(5) + d_2(6 - 5), f_3(6) + d_2(6 - 6)\}$ = max $\{0 + 106, 5 + 90, 15 + 80, 40 + 40, 60 + 26, 70 + 5, 73 + 0\} = 106$.

U prethodnu tablicu dodajemo još dva stupca:

c	$d_1(c)$	x_1^*	$d_2(c)$	x_2^*	$d_3(c)$	x_3^*
0	0	0	0	0	0	0
1	4	1	5	1	5	0, 1
2	26	2	26	0	26	0
3	40	3	40	0, 3	40	0, 3
4	45	4	80	4	80	0
5	50	5	90	5	90	0
6	51	6	106	4	106	0

Iz dobivene tablice "očitavamo" optimalnu očekivanu neto–dobit od ulaganja. Kako smo intuitivno i naslućivali, ona je jednaka $d_3(6) = 106$ n.j. i postiže se za $x_3^* = 0$. Dakle, u djelatnost D_3 ne treba uložiti ništa. Preostali iznos je 6 – 0 = 6 n.j. i njega ulažemo u djelatnosti D_1 i D_2. Pripadna optimalna vrijednost je $d_2(6) = 106$ n.j., a postiže se za $x_2^* = 4$. Stoga u djelatnost D_2 treba uložiti ukupno 4 n.j., pa u djelatnost D_1 preostaje uložiti ukupno 6 – 4 = 2 n.j. Pregledno to možemo prikazati sljedećom tablicom:

optimalni plan ulaganja:

djelatnost	D_1	D_2	D_3
optimalno ulaganje [n.j.]	2	4	0

Primjer 7. Problem optimalnoga ulaganja kapitala u razvoj određene regije ili određene gospodarske grane može se definirati na sljedeći način: Za ostvarivanje bržega razvoja određene gospodarske grane predviđen je kapital u iznosu od C n.j. Taj kapital se može uložiti u ukupno n različitih tvrtki iz navedene gospodarske grane. Ovisno o uloženom kapitalu, svaka pojedina tvrtka će ostvariti dodatnu neto–dobit (razliku ukupne neto–dobiti i uloženoga kapitala). Treba naći razdiobu raspoloživoga kapitala na navedene tvrtke tako da ukupna dodatna neto–dobit (tj. zbroj dodatnih neto–dobiti svih n tvrtki) bude maksimalna.

Konkretno, za ostvarivanje bržega razvoja određene gospodarske grane predviđen je kapital u iznosu od C = 4.000.000,00 kn. Taj se kapital ulaže u ukupno tri tvrtke: T_1, T_2 i T_3. Očekivana dodatna neto–dobit koju će ostvariti svaka tvrtka navedena je u donjoj tablici.

iznos ulaganja [000 000 kn]	*očekivana dodatna neto–dobit tvrtke* [000 000 kn]		
	T_1	T_2	T_3
0	0	0	0
1	0.3	0.29	0.31
2	0.47	0.45	0.46
3	0.7	0.67	0.74
4	0.83	0.82	0.8

a) Odredimo opći matematički model promatranoga problema uz pretpostavku da svi iznosi ulaganja nužno moraju biti cjelobrojni.
b) Za konkretni slučaj nađimo optimalnu razdiobu predviđenoga kapitala.

Opći matematički model: Za svaki $i \in [n]$ označimo s x_i iznos ulaganja u tvrtku T_i, a $f_i(x_i)$ pripadnu očekivanu dodatnu neto–dobit. Tada je ukupna očekivana dodatna neto–dobit od ulaganja u svih n tvrtki jednaka

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) = \sum_{i=1}^{n} f_i(x_i)$$

pa zahtijevamo da vrijednost te funkcije bude maksimalna. Pogledajmo pripadne uvjete uz koje tražimo maksimalnu vrijednost. Zbroj svih ulaganja ne smije biti strogo veći od ukupnoga iznosa raspoloživoga kapitala C, pa mora vrijediti nejednakost

$$x_1 + x_2 + \ldots + x_n \leq C,$$

tj.

$$\sum_{i=1}^{n} x_i \leq C.$$

Iznos svakoga ulaganja nužno mora biti cjelobrojan, a prema prirodi problema on mora biti i nenegativan i najviše jednak C. Zbog toga vrijedi skupovna relacija:

$$x_i \in [C]_0, \text{ za svaki } i \in [n].$$

Prema tome, traženi *matematički model* je:

$$\text{maksimizirati } f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$$

pod uvjetima

$$\sum_{i=1}^{n} x_i \le C$$

$$x_i \in [C]_0, \text{ za svaki } i \in [n].$$

b) U ovome su slučaju $n = 3$ i $C = 4$, pa je odgovarajući *matematički model*

$$\text{maksimizirati } f(x_1, x_2, x_3) = f_1(x_1) + f_2(x_2) + f_3(x_3)$$

pod uvjetima

$$x_1 + x_2 + x_3 \le 4$$

$$x_i \in [4]_0, \text{ za svaki } i \in [3].$$

Stavimo $S = [4]_0$. Definirajmo realne funkcije $d_1, d_2, d_3 : S \to \mathbf{R}$ s:

$$d_1(s) = f_1(s),$$

$$d_2(s) = \max_{\substack{0 \le x_2 \le \min\{s,4\} \\ x_2 \in \mathbb{Z}}} \left[f_2(x_2) + d_1(s - x_2) \right],$$

$$d_3(c) = \max_{\substack{0 \le x_3 \le \min\{s,6\} \\ x_3 \in \mathbb{Z}}} \left[f_3(x_3) + d_2(s - x_3) \right].$$

i računajmo vrijednost svake od tih funkcija za svaki $s \in S$. Analogno kao u prethodnim primjerima dobiva se sljedeća tablica (provjerite je!):

s	$d_1(s)$	x_1^*	$d_2(s)$	x_2^*	$d_3(s)$	x_3^*
0	0	0	0	0	0	0
1	0.3	1	0.3	0	0.31	1
2	0.47	2	0.59	1	0.61	1
3	0.7	3	0.76	1	0.90	1
4	0.83	4	0.99	1	1.23	1

Vrijednost $d_3(4) = 1.23$ interpretiramo ovako: *maksimalna očekivana dodatna neto–dobit (nastala ulaganjem kapitala* $C = 4.000.000{,}00$ kn *u sve tri tvrtke iznosi* 1.230.000,00 kn. Iz tablice lagano "očitavamo" i pripadni optimalni plan ulaganja: vrijednost $d_3(4) = 1.23$ postiže se za $x_1^* = 2$, $x_2^* = x_3^* = 1$, pa *u tvrtke* T_2 i T_3 *treba uložiti po* 1.000.000,00 kn, *a u tvrtku* T_1 (preostalih) 2.000.000,00 kn.

Zadaci za vježbu:

1. Na raspolaganju nam je ukupno 8.000.000,00 € koje možemo uložiti u tri različite aktivnosti: A_1, A_2 i A_3. Odgovarajuće očekivane neto-dobiti u ovisnosti o iznosu ulaganja za svaku su pojedinu aktivnost navedene u sljedećoj tablici. Pretpostavljamo da je dobit ostvarena ulaganjem u bilo koju aktivnost neovisna o ulaganjima u preostale aktivnosti.

ulaganje [mil. €]	*očekivana neto–dobit od A_1* [mil. €]	*očekivana neto–dobit od A_2* [mil. €]	*očekivana neto–dobit od A_3* [mil. €]
0	0	0	0
1	5	5	4
2	15	15	26
3	40	40	40
4	80	60	45
5	90	70	50
6	95	73	51
7	98	74	52
8	100	75	53

Treba odrediti optimalnu razdiobu (ne nužno cjelokupnoga) kapitala tako da ukupna očekivana neto–dobit bude maksimalna, pri čemu iznos ulaganja u svaku tvrtku nužno mora biti cjelobrojni višekratnik broja 1 000 000.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalnu razdiobu cjelokupnoga kapitala.
c) Koristeći rezultat **b)** podzadatka, nađite razdiobu kapitala od 6.000.000,00 € tako da ukupna očekivana neto–dobit od ulaganja u sve tri aktivnosti bude maksimalna.

2. Pretpostavimo da raspolažemo ukupnim kapitalom od 10 novčanih jedinica kojega želimo uložiti u 4 različita proizvodna područja. Očekivana dobit u svakom pojedinom području iskazana u ovisnosti o ulaganju u to područje navedena je u sljedećoj tablici.

ulaganje [n.j.]	*očekivana dobit na području A*	*očekivana dobit na području B*	*očekivana dobit na području C*	*očekivana dobit na području D*
0	0	0	0	0
1	0.28	0.25	0.15	0.20
2	0.45	0.41	0.25	0.33
3	0.65	0.55	0.40	0.42
4	0.78	0.65	0.50	0.48
5	0.90	0.75	0.62	0.53
6	1.02	0.80	0.73	0.56
7	1.13	0.85	0.82	0.58
8	1.23	0.88	0.90	0.60
9	1.32	0.90	0.96	0.60
10	1.38	0.90	1.00	0.60

Treba odrediti razdiobu ukupnoga kapitala na sva četiri navedena područja tako da ukupna očekivana dobit bude maksimalna, pri čemu iznos svakoga pojedinoga uloga nužno mora biti cjelobrojan.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalnu razdiobu kapitala i pripadnu optimalnu ukupnu očekivanu dobit.
c) Koristeći rezultat **b)** zadatka, nađite razdiobu kapitala u iznosu od 9 n.j. tako da ukupna očekivana dobit bude maksimalna.

3. Kapital od 30 n.j. raspoređuje se na tri industrijske grane G_1, G_2 i G_3. U granu G_1 može se uložiti najviše 10 n.j., a za x uloženih novčanih jedinica ostvaruje se neto–dobit $0.4 \cdot x^2$ n.j. U granu G_2 može se uložiti najviše 8 n.j., a za y uloženih novčanih jedinica ostvaruje se neto–dobit $10 \cdot y$ n.j. U granu G_3 može se uložiti najviše 15 n.j., a za z uloženih novčanih jedinica ostvaruje se neto–dobit $12 \cdot z$ n.j. Treba rasporediti cjelokupan kapital na sve tri grane tako da iznos uložen u svaku granu bude cjelobrojan, a ukupna ostvarena neto–dobit maksimalna.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalnu razdiobu kapitala i pripadnu optimalnu ukupnu neto–dobit.
c) Zbog neplaniranih se troškova predviđeni kapital smanjen za 20%. Kako će se to smanjenje odraziti na predviđeno optimalno ulaganje u svaku pojedinu granu? (*Naputak*: Nađite novu optimalnu razdiobu kapitala, pa za svaku pojedinu granu odredite smjer i postotak promjene optimalnoga uloga u odnosu na optimalnu razdiobu dobivenu u **b)** podzadatku.)

4. Tijekom tri mjeseca treba proizvesti točno 15 komada nekoga proizvoda, pri čemu je najveći mogući kapacitet proizvodnje 7 komada proizvoda mjesečno. Troškovi proizvodnje razmjerni su kvadratima broja proizvedenih proizvoda, te za svaki $i \in [3]$ vrijedi: ako se u i – tom mjesecu proizvede x_i komada proizvoda, troškovi proizvodnje iznose $w_i \cdot x_i^2$ n.j., gdje su w_i strogo pozitivni realni brojevi. Treba odrediti plan proizvodnje takav da ukupni troškovi proizvodnje budu minimalni.

a) Odredite opći matematički model promatranoga problema.
b) Riješite problem za konkretne vrijednosti: $w_1 = 9$, $w_2 = 21$ i $w_3 = 28$.
c) Riješite **b)** podzadatak ako je u drugom mjesecu zbog remonta dijela postrojenja moguće proizvesti najviše 4 komada proizvoda. Za koliko će se postotaka u tom slučaju povećati ukupni troškovi proizvodnje u odnosu na optimalne troškove proizvodnje?

5. U tijeku jednoga mjeseca (4 tjedna) treba proizvesti ukupno 18 komada nekoga proizvoda. Troškovi proizvodnje x_i komada proizvoda u i – tom tjednu zadani su izrazom $w_i \cdot x_i^2$. Tijekom jednoga tjedna moguće je proizvesti najviše 8 komada proizvoda. Treba izraditi plan proizvodnje tako da njezini ukupni troškovi budu minimalni.

a) Odredite matematički model promatranoga problema.
b) Riješite problem za konkretne vrijednosti: $w_1 = 10$, $w_2 = 25$, $w_3 = 20$ i $w_4 = 30$.
c) Riješite **b)** podzadatak ako je u prvom mjesecu zbog remonta dijela postrojenja moguće proizvesti najviše 5 komada proizvoda. Za koliko će se postotaka u tom slučaju povećati ukupni troškovi proizvodnje u odnosu na optimalne troškove proizvodnje?

1.4. Problem složene razdiobe ulaganja

Problem složene razdiobe ulaganja definira se slično problemu jednostavne razdiobe ulaganja. Osnovne postavke i način rješavanja objasnit ćemo na konkretnim primjerima.

Primjer 1. Tvrtka izvodi montažu dva tipa objekata: A i B. Montažom jednoga objekta tipa A ostvaruje se dobit od 8 n.j., a montažom jednoga objekta tipa B ostvaruje se dobit od 12 n.j. Prigodom montaže obaju tipova objekata koristi se posebna oprema čiji je ukupni raspoloživi kapacitet 10 komada. Za montažu jednoga objekta tipa A potrebna su 2 komada, a za montažu jednoga objekta tipa B 3 komada te opreme. Tvrtka želi ostvariti maksimalnu dobit *istodobnom* montažom objekata obaju navedenih tipova.

a) Odredimo matematički model promatranoga problema.
b) Nađimo optimalan plan montaže.
c) Ispitajmo kako na plan dobiven pod **b)** utječe poseban ugovor kojim se tvrtka obvezuje da će montirati najmanje 1, a najviše 3 objekta tipa A.

Matematički model: Označimo s x_1 ukupan broj objekata tipa A koje će tvrtka montirati, a s x_2 ukupan broj objekata tipa B koje će tvrtka montirati. Ukupna dobit od montaže jednoga objekta tipa A je 8 n.j., pa ukupna dobit od montaže x_1 objekata tipa A iznosi $8 \cdot x_1$ n.j. Potpuno analogno, ukupna dobit od montaže x_2 objekata tipa B iznosi $12 \cdot x_2$. Prema tome, ukupna dobit koja se dobije montažom obaju tipova objekata iznosi:

$$f(x_1, x_2) = 8 \cdot x_1 + 12 \cdot x_2.$$

Nadalje, ako za montažu jednoga objekta tipa A tvrtka treba 2 komada posebne opreme, onda za istodobnu montažu x_1 objekata tipa A tvrtka treba $2 \cdot x_1$ komada posebne opreme. Potpuno analogno, za istodobnu montažu x_2 objekata tipa B potrebno je $3 \cdot x_2$ komada posebne opreme. Budući da ukupan broj komada posebne opreme ne smije premašiti raspoloživi kapacitet, mora vrijediti nejednakost

$$2 \cdot x_1 + 3 \cdot x_2 \leq 10.$$

Prema prirodi postavljenoga problema, x_1 i x_2 moraju biti nenegativni cijeli brojevi, jer broj montiranih komada ne može biti niti decimalan broj niti negativan broj, tj.

$$x_i \in \mathbf{N}_0, \text{ za } i = 1, 2.$$

Dakle, traženi matematički model je:

$$\text{maksimizirati } f(x_1, x_2) = 8 \cdot x_1 + 12 \cdot x_2$$

pod uvjetima

$$2 \cdot x_1 + 3 \cdot x_2 \leq 10$$

$$x_i \in \mathbf{N}_0, \text{ za } i = 1, 2.$$

"*Klasično" rješenje*: Stavimo $S = [10]_0$, pa za svaki $s \in S$ računajmo vrijednosti funkcija

$$f_1(s) = \max_{0 \le 2 \cdot x_1 \le s} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{s}{2}} (8 \cdot x_1)$$

$$f_2(s) = \max_{0 \le 3 \cdot x_2 \le s} \left[12 \cdot x_2 + f_1(s - 3 \cdot x_2)\right] = \max_{0 \le x_2 \le \frac{s}{3}} \left[12 \cdot x_2 + f_1(s - 3 \cdot x_2)\right]$$

Prva funkcija računa *optimalnu* dobit u slučaju kad se sva raspoloživa posebna oprema koristi isključivo za montiranje objekata tipa A, a druga funkcija računa *optimalnu* dobit kad se sva raspoloživa posebna oprema koristi za montažu obiju tipova objekata.

Vrijednosti navedenih funkcija računamo potpuno analogno kao i u problemima jednostavne razdiobe ulaganja. U prvom koraku dobivamo:

$f_1(0) = \max_{0 \le 2 \cdot x_1 \le 0} (8 \cdot x_1) = \max_{0 \le x_1 \le 0} (8 \cdot x_1) = 8 \cdot 0 = 0;$

$f_1(1) = \max_{0 \le 2 \cdot x_1 \le 1} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{1}{2}} (8 \cdot x_1) =$ (u segmentu $[0, \frac{1}{2}]$ nalazi se samo jedna cjelobrojna vrijednost: 0) $= 8 \cdot 0 = 0;$

$f_1(2) = \max_{0 \le 2 \cdot x_1 \le 2} (8 \cdot x_1) = \max_{0 \le x_1 \le 1} (8 \cdot x_1) = \max \{8 \cdot 0, 8 \cdot 1\} = 8;$

$f_1(3) = \max_{0 \le 2 \cdot x_1 \le 3} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{3}{2}} (8 \cdot x_1) =$ (u segmentu $[0, \frac{3}{2}]$ nalaze se dvije cjelobrojne vrijednosti: 0 i 1) $= \max \{8 \cdot 0, 8 \cdot 1\} = 8;$

$f_1(4) = \max_{0 \le 2 \cdot x_1 \le 4} (8 \cdot x_1) = \max_{0 \le x_1 \le 2} (8 \cdot x_1) = \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2\} = 16;$

$f_1(5) = \max_{0 \le 2 \cdot x_1 \le 5} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{5}{2}} (8 \cdot x_1) =$ (u segmentu $[0, \frac{5}{2}]$ nalaze se tri cjelobrojne vrijednosti: 0, 1 i 2) $= \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2\} = 16;$

$f_1(6) = \max_{0 \le 2 \cdot x_1 \le 6} (8 \cdot x_1) = \max_{0 \le x_1 \le 3} (8 \cdot x_1) = \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2, 8 \cdot 3\} = 24;$

$f_1(7) = \max_{0 \le 2 \cdot x_1 \le 7} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{7}{2}} (8 \cdot x_1) =$ (u segmentu $[0, \frac{7}{2}]$ nalaze se četiri cjelobrojne vrijednosti: 0, 1, 2 i 3) $= \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2, 8 \cdot 3\} = 24;$

$f_1(8) = \max_{0 \le 2 \cdot x_1 \le 8} (8 \cdot x_1) = \max_{0 \le x_1 \le 4} (8 \cdot x_1) = \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2, 8 \cdot 3, 8 \cdot 4\} = 32;$

$f_1(9) = \max_{0 \le 2 \cdot x_1 \le 9} (8 \cdot x_1) = \max_{0 \le x_1 \le \frac{9}{2}} (8 \cdot x_1) =$ (u segmentu $[0, \frac{9}{2}]$ nalazi se pet cjelobrojnih vrijednosti: 0, 1, 2, 3 i 4) $= \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2, 8 \cdot 3, 8 \cdot 4\} = 32;$

$f_1(10) = \max_{0 \le 2 \cdot x_1 \le 10} (8 \cdot x_1) = \max_{0 \le x_1 \le 5} (8 \cdot x_1) = \max \{8 \cdot 0, 8 \cdot 1, 8 \cdot 2, 8 \cdot 3, 8 \cdot 4, 8 \cdot 5\} = 40;$

Tako dobivamo sljedeću tablicu:

s	$f_1(s)$	x_1^*
0	0	0
1	0	0
2	8	1
3	8	1
4	16	2
5	16	2
6	24	3
7	24	3
8	32	4
9	32	4
10	40	5

U sljedećem koraku računamo vrijednosti funkcije $f_2(x)$ koristeći dobivenu tablicu. Imamo redom:

$f_2(0) = \max\limits_{0\le 3\cdot x_2\le 0}\left[12\cdot x_2 + f_1(0-3\cdot x_2)\right] = \max\limits_{0\le x_2\le 0}\left[12\cdot x_2 + f_1(0-3\cdot x_2)\right] = 12 \cdot 0 + f_1(0 - 3 \cdot 0) = 0 + f_1(0) = 0 + 0 = 0$;

$f_2(1) = \max\limits_{0\le 3\cdot x_2\le 1}\left[12\cdot x_2 + f_1(1-3\cdot x_2)\right] = \max\limits_{0\le x_2\le \frac{1}{3}}\left[12\cdot x_2 + f_1(1-3\cdot x_2)\right]$ = (u segmentu $[0, \frac{1}{3}]$ nalazi se samo jedna cjelobrojna vrijednost: 0) $= 12 \cdot 0 + f_1(1 - 3 \cdot 0) = 0 + f_1(1) = 0 + 0 = 0$;

$f_2(2) = \max\limits_{0\le 3\cdot x_2\le 2}\left[12\cdot x_2 + f_1(2-3\cdot x_2)\right] = \max\limits_{0\le x_2\le \frac{2}{3}}\left[12\cdot x_2 + f_1(2-3\cdot x_2)\right]$ = (u segmentu $[0, \frac{2}{3}]$ nalazi se samo jedna cjelobrojna vrijednost: 0) $= 12 \cdot 0 + f_1(2 - 3 \cdot 0) = 0 + f_1(2) = 0 + 8 = 8$;

$f_2(3) = \max\limits_{0\le 3\cdot x_2\le 3}\left[12\cdot x_2 + f_1(3-3\cdot x_2)\right] = \max\limits_{0\le x_2\le 1}\left[12\cdot x_2 + f_1(2-3\cdot x_2)\right] = \max\{12 \cdot 0 + f_1(3 - 3 \cdot 0), 12 \cdot 1 + f_1(3 - 3 \cdot 1)\} = \max\{0 + f_1(3), 12 + f_1(0)\} = \max\{0 + 8, 12 + 0\} = 12$;

$f_2(4) = \max\limits_{0\le 3\cdot x_2\le 4}\left[12\cdot x_2 + f_1(4-3\cdot x_2)\right] = \max\limits_{0\le x_2\le \frac{4}{3}}\left[12\cdot x_2 + f_1(4-3\cdot x_2)\right]$ = (u segmentu $[0, \frac{4}{3}]$ nalaze se dvije cjelobrojne vrijednosti: 0 i 1) $= \max\{12 \cdot 0 + f_1(4 - 3 \cdot 0), 12 \cdot 1 + f_1(4 - 3 \cdot 1)\} = \max\{0 + f_1(4), 12 + f_1(1)\} = \max\{0 + 16, 12 + 0\} = 16$;

$f_2(5) = \max\limits_{0\le 3\cdot x_2\le 5}\left[12\cdot x_2 + f_1(5-3\cdot x_2)\right] = \max\limits_{0\le x_2\le \frac{5}{3}}\left[12\cdot x_2 + f_1(5-3\cdot x_2)\right]$ = (u segmentu $[0, \frac{5}{3}]$ nalaze se dvije cjelobrojne vrijednosti: 0 i 1) $= \max\{12 \cdot 0 + f_1(5 - 3 \cdot 0), 12 \cdot 1 + f_1(5 - 3 \cdot 1)\} = \max\{0 + f_1(5), 12 + f_1(2)\} = \max\{0 + 16, 12 + 8\} = 20$;

$f_2(6) = \max\limits_{0\le 3\cdot x_2\le 6}\left[12\cdot x_2 + f_1(6-3\cdot x_2)\right] = \max\limits_{0\le x_2\le 2}\left[12\cdot x_2 + f_1(6-3\cdot x_2)\right] = \max\{12 \cdot 0 + f_1(6 - 3 \cdot 0), 12 \cdot 1 + f_1(6 - 3 \cdot 1), 12 \cdot 2 + f_1(6 - 3 \cdot 2)\} = \max\{0 + f_1(6), 12 + f_1(3), 24 + f_1(0)\} = \max\{0 + 24, 12 + 8, 24 + 0\} = 24$;

$f_2(7) = \max\limits_{0 \le 3 \cdot x_2 \le 7} \left[12 \cdot x_2 + f_1(7 - 3 \cdot x_2)\right] = \max\limits_{0 \le x_2 \le \frac{7}{3}} \left[12 \cdot x_2 + f_1(7 - 3 \cdot x_2)\right]$ = (u segmentu [0, $\frac{7}{3}$] nalaze se tri cjelobrojne vrijednosti: 0, 1 i 2) = max {$12 \cdot 0 + f_1(7 - 3 \cdot 0)$, $12 \cdot 1 + f_1(7 - 3 \cdot 1)$, $12 \cdot 2 + f_1(7 - 3 \cdot 2)$} = max {$0 + f_1(7)$, $12 + f_1(4)$, $24 + f_1(1)$} = max {0 + 24, 12 + 16, 24 + 0} = 28;

$f_2(8) = \max\limits_{0 \le 3 \cdot x_2 \le 8} \left[12 \cdot x_2 + f_1(8 - 3 \cdot x_2)\right] = \max\limits_{0 \le x_2 \le \frac{8}{3}} \left[12 \cdot x_2 + f_1(8 - 3 \cdot x_2)\right]$ = (u segmentu [0, $\frac{8}{3}$] nalaze se tri cjelobrojne vrijednosti: 0, 1 i 2) = max {$12 \cdot 0 + f_1(8 - 3 \cdot 0)$, $12 \cdot 1 + f_1(8 - 3 \cdot 1)$, $12 \cdot 2 + f_1(8 - 3 \cdot 2)$} = max {$0 + f_1(8)$, $12 + f_1(5)$, $24 + f_1(2)$} = max {0 + 32, 12 + 16, 24 + 8} = 32;

$f_2(9) = \max\limits_{0 \le 3 \cdot x_2 \le 9} \left[12 \cdot x_2 + f_1(9 - 3 \cdot x_2)\right] = \max\limits_{0 \le x_2 \le 3} \left[12 \cdot x_2 + f_1(9 - 3 \cdot x_2)\right]$ = max {$12 \cdot 0 + f_1(9 - 3 \cdot 0)$, $12 \cdot 1 + f_1(9 - 3 \cdot 1)$, $12 \cdot 2 + f_1(9 - 3 \cdot 2)$, $12 \cdot 3 + f_1(9 - 3 \cdot 3)$} = max {$0 + f_1(9)$, $12 + f_1(6)$, $24 + f_1(3)$, $36 + f_1(0)$} = max {0 + 32, 12 + 24, 24 + 8, 36 + 0} = 36;

$f_2(10) = \max\limits_{0 \le 3 \cdot x_2 \le 10} \left[12 \cdot x_2 + f_1(10 - 3 \cdot x_2)\right] = \max\limits_{0 \le x_2 \le \frac{10}{3}} \left[12 \cdot x_2 + f_1(10 - 3 \cdot x_2)\right]$ = (u segmentu [0, $\frac{10}{3}$] nalaze se četiri cjelobrojne vrijednosti: 0, 1, 2 i 3) = max {$12 \cdot 0 + f_1(10 - 3 \cdot 0)$, $12 \cdot 1 + f_1(10 - 3 \cdot 1)$, $12 \cdot 2 + f_1(10 - 3 \cdot 2)$, $12 \cdot 3 + f_1(10 - 3 \cdot 3)$} = max {$0 + f_1(10)$, $12 + f_1(7)$, $24 + f_1(4)$, $36 + f_1(1)$} = max {0 + 40, 12 + 16, 24 + 16, 36 + 0} = 40.

Tako dobivamo sljedeću tablicu:

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*
0	0	0	0	0
1	0	0	0	0
2	8	1	8	0
3	8	1	12	1
4	16	2	16	0
5	16	2	20	1
6	24	3	24	0, 2
7	24	3	28	1
8	32	4	32	0, 2
9	32	4	36	1, 3
10	40	5	40	0, 2

Dakle, optimalna dobit iznosi $f_2(10) = 40$ n.j. i postiže se za $x_2^* = 0$ ili $x_2^* = 2$. Stoga postoje dva moguća optimalna plana montaže:

1.) $x_2^* = 0$ znači da se ne predviđa montaža niti jednoga objekta tipa *B*. Pripadna optimalna vrijednost funkcije f_1 je $f_1(10 - 3 \cdot 0) = f_1(10) = 40$, a postiže se za $x_1^* = 5$. Prema tome, treba montirati točno 5 objekata tipa *A*.

2.) $x_2^* = 2$ znači da treba montirati dva objekta tipa B. Pripadna optimalna vrijednost funkcije f_1 je $f_1(10 - 3 \cdot 2) = f_1(4) = 16$, a postiže se za $x_1^* = 2$. U ovom slučaju, dakle, treba montirati po dva objekta svakoga tipa.

<u>*Analiza osjetljivosti rješenja na dodatan uvjet*</u>: Činjenicu da prema posebnom ugovoru tvrtka treba montirati najmanje 1, a najviše 3 objekta tipa A možemo zapisati u obliku nejednakosti:

$$1 \le x_1 \le 3.$$

Stoga u ovom slučaju prva tri stupca posljednje tablice izgledaju ovako:

s	$f_1(s)$	x_1^*
0	0	0
1	0	0
2	8	1
3	8	1
4	16	2
5	16	2
6	24	3
7	24	3
8	24	3
9	24	3
10	24	3

Tako vidimo da se rezultati mijenjaju jedino za $s \in \{8, 9, 10\}$, pa se analognim računom dobiva:

s	$f_1(s)$	x_1^*	$f_2(s)$	x_2^*
0	0	0	0	0
1	0	0	0	0
2	8	1	8	0
3	8	1	12	1
4	16	2	16	0
5	16	2	20	1
6	24	3	24	0, 2
7	24	3	28	1
8	24	3	32	2
9	24	3	36	1, 3
10	24	3	40	2

U ovome je slučaju optimalan plan montaže $x_1^* = x_2^* = 2$, tj. treba montirati po dva objekta svakoga tipa.

<u>*Rješenje pomoću programa Winqsb*</u>: Problem složene razdiobe ulaganja svodi se na problem ruksaka u kojemu jedan komad namirnice N_1 ima masu m_1, jedan komad namirnice n_2 masu m_2 itd., a na raspolaganju su nam n_1 komada namirnice N_1, n_2 komada namirnice N_2 itd. Ukoliko, eventualno, broj komada svake pojedine namirnice nije definiran, uvijek možemo

uzeti da su ti brojevi jednaki kapacitetu ruksaka (to je, dakako, prevelika vrijednost koja nema nikakva utjecaja na dobivanje rješenja problema osim, eventualno, nešto sporije brzine izračuna toga rješenja).

Već uobičajeno, pokrenimo potprogram *DP* programa *Winqsb*, lijevom tipkom miša kliknimo na izbornik *File* i odaberimo opciju *New Problem*. U dobivenom dijaloškom okviru lijevom tipkom miša kliknimo na kružić pored natpisa *Knapsack Problem*, pa u pravokutnik pored natpisa *Problem Title* upišimo *Problem* 1-4-1, a u pravokutnik pored natpisa *Number of Items* upišimo 2 (jer imamo 2 tipa objekata).

U sljedećem koraku unosimo ulazne podatke:

- u drugi stupac (*Item Identification*) upisujemo nazive tipova objekata: *A* i *B*;
- u treći stupac (*Units Available*) trebali bismo upisati koliko nam je maksimalno objekata kojega tipa na raspolaganju, ali te brojeve ne znamo. Sukladno ranijoj napomeni, uvijek možemo upisati da imamo na raspolaganju onoliko komada svake vrste objekata koliko iznosi kapacitet raspoložive opreme, no, možemo postupiti i lukavije. Riješimo nejednadžbe $2 \cdot x_1 \leq 10$ i $3 \cdot x_2 \leq 10$ u skupu **N** (zbog zahtjeva $x_1, x_2 \in \mathbf{N}$). Dobivamo $x_1 \leq 5$ i $x_2 \leq 3$. To znači da ne možemo montirati više od 5 objekata tipa *A* i više od 3 objekta tipa *B*, pa u ovaj stupac upisujemo 5 i 3.
- u četvrti stupac (*Unit Capacity Required*) upisujemo težinske koeficijente koji množe veličine x_1 i x_2 u uvjetu iz matematičkoga modela. To su brojevi 2 i 3, pa u četvrti stupac upisujemo redom 2 i 3.
- u peti stupac (*Return Function* (*X*: *Item ID*)) upisujemo komponente funkcije cilja koje odgovaraju svakoj pojedinom objektu, a to je zapravo dobit od montaže svakoga pojedinoga objekta. Dakle, upisujemo redom: 8*X i 12*X.
- u posljednji redak (*Capacity*=) upisujemo kapacitet raspoložive opreme, tj. 10.

Ovime je unos ulaznih podataka završen. Lijevom tipkom miša kliknimo na izbornik *Solve and Analyze*, pa odaberimo opciju *Solve the Problem*. Iz trećega stupca dobivene tablice (*Decision Quantity* (*X*)) očitavamo optimalan plan montaže: treba montirati po 2 objekta svakoga tipa. Dakle, riječ je o drugom od dvaju optimalnih planova montaže koje smo dobili "klasičnim" rješavanjem problema. U ovom slučaju ne treba provesti analizu osjetljivosti dobivenoga rješenja jer optimalno rješenje $x_1^* = 2$ već zadovoljava uvjet $1 \leq x_1 \leq 3$, pa bismo tom analizom dobili isto optimalno rješenje.

<u>Primjer 2.</u> Neka tvrtka treba montirati tri tipa objekata: A_1, A_2 i A_3. Pri montaži svih tipova objekata koristi se posebna oprema čiji je ukupni kapacitet 18 komada. Osnovni podaci o broju potrebnih komada posebne opreme i dobiti za izradu svakoga pojedinoga tipa opreme navedeni su u sljedećoj tablici.

tip objekta	*broj potrebnih komada opreme*	*dobit* [n.j./objekt]
A_1	4	8
A_2	6	12
A_3	3	10

Odlukom uprave tvrtke dozvoljena je istodobna montaža najviše 3 objekta tipa A_1 i 5 objekata tipa A_3. Tvrtka treba napraviti optimalan plan montaže tako da ostvarena dobit bude maksimalna.

a) Odredimo matematički model promatranoga problema.
b) Nađimo optimalan plan montaže.

<u>*Matematički model*</u>: Za svaki $i = 1, 2, 3$ neka je x_i broj objekata tipa A_i koji će se montirati. Tada ukupna dobit od montaže objekata tipa A_1 iznosi $8 \cdot x_1$ n.j., ukupna dobit od montaže objekata tipa A_2 $12 \cdot x_2$ n.j., a ukupna dobit od montaže objekata tipa A_3 $10 \cdot x_3$ n.j. Stoga je ukupna dobit od montaže svih triju tipova objekata

$$f(x_1, x_2, x_3) = 8 \cdot x_1 + 12 \cdot x_2 + 10 \cdot x_3.$$

Za montažu x_1 objekata tipa A_1 treba ukupno $4 \cdot x_1$ komada opreme, za montažu x_2 objekata tipa A_2 treba ukupno $6 \cdot x_2$ komada opreme, a za montažu x_3 objekata tipa A_3 treba ukupno $3 \cdot x_3$ komada opreme. Budući da ukupan broj potrebnih komada opreme ne može premašiti ukupan kapacitet opreme, mora vrijediti nejednakost

$$4 \cdot x_1 + 6 \cdot x_2 + 3 \cdot x_3 \leq 18.$$

Nadalje, budući da je istodobno moguće montirati najviše 3 objekta tipa A_1 i najviše 5 objekata tipa A_2, moraju vrijediti nejednakosti

$$0 \leq x_1 \leq 3 \text{ i } 0 \leq x_3 \leq 5.$$

Prema tome, traženi matematički model je:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 8 \cdot x_1 + 12 \cdot x_2 + 10 \cdot x_3$$

pod uvjetima

$$4 \cdot x_1 + 6 \cdot x_2 + 3 \cdot x_3 \leq 18$$
$$x_1 \in [3]_0,$$
$$x_2 \in \mathbf{N}_0,$$
$$x_3 \in [5]_0.$$

<u>*"Klasično" rješenje*</u>: Stavimo $S = [8]_0$, pa za svaki $s \in S$ računajmo vrijednosti sljedećih funkcija:

$$f_1(s) = \max_{\substack{0 \leq 4 \cdot x_1 \leq s \\ 0 \leq x_1 \leq 3}} (8 \cdot x_1) = \max_{0 \leq x_1 \leq \min\left\{\frac{s}{4},\ 3\right\}} (8 \cdot x_1)$$

$$f_2(s) = \max_{0 \leq 6 \cdot x_2 \leq s} \left[12 \cdot x_2 + f_1(s - 6 \cdot x_2)\right] = \max_{0 \leq x_2 \leq \frac{s}{6}} \left[12 \cdot x_2 + f_1(s - 6 \cdot x_2)\right]$$

$$f_3(s) = \max_{\substack{0 \leq 3 \cdot x_3 \leq s \\ 0 \leq x_3 \leq 5}} \left[10 \cdot x_3 + f_2(s - 3 \cdot x_3)\right] = \max_{0 \leq x_3 \leq \min\left\{\frac{s}{3},\ 5\right\}} \left[10 \cdot x_3 + f_2(s - 3 \cdot x_2)\right]$$

Funkcija f_1 računa optimalnu dobit ukoliko se s komada opreme koristi isključivo za montažu objekata tipa A_1, funkcija f_2 računa optimalnu dobit ukoliko se s komada opreme koristi

isključivo za montažu objekata tipa A_1 i A_2, a funkcija f_3 računa optimalnu dobit ukoliko se s komada opreme koristi za montažu svih triju tipova objekata.

U prvom koraku dobivamo sljedeću tablicu:

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$f_1(s)$	0	0	0	0	8	8	8	8	16	16	16	16	24	24	24	24	24	24	24
x_1^*	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3

U drugom koraku dobivamo sljedeću tablicu:

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$f_1(s)$	0	0	0	0	8	8	8	8	16	16	16	16	24	24	24	24	24	24	24
x_1^*	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3
$f_2(s)$	0	0	0	0	8	8	12	12	16	16	20	20	24	24	28	28	32	32	36
x_2^*	0	0	0	0	0	0	1	1	0	0	1	1	0, 2	0, 2	1	1	2	2	1, 3

U trećem koraku dobivamo sljedeću tablicu:

S	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$f_1(s)$	0	0	0	0	8	8	8	8	16	16	16	16	24	24	24	24	24	24	24
x_1^*	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3
$f_2(s)$	0	0	0	0	8	8	12	12	16	16	20	20	24	24	28	28	32	32	36
x_2^*	0	0	0	0	0	0	1	1	0	0	1	1	0, 2	0, 2	1	1	2	2	1, 3
$f_3(s)$	0	0	0	10	10	10	20	20	20	30	30	30	40	40	40	50	50	50	52
x_3^*	0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	4

Prema tome, optimalna dobit je $f_3(18) = 52$ n.j. i postiže se za $x_3^* = 4$, što znači da treba montirati ukupno 4 objekta tipa A_3. Pripadna optimalna vrijednost funkcije f_2 je $f_2(18 - 3 \cdot 4) = f_2(6) = 12$ koja se postiže za $x_2^* = 1$, pa treba montirati jedan objekt tipa A_2. Napokon, pripadna optimalna vrijednost funkcije f_1 je $f_1(6 - 6 \cdot 1) = f_1(0) = 0$ i postiže se za $x_1^* = 0$, pa ne treba montirati niti jedan objekt tipa A_1. Dakle, optimalan plan montaže je: *montirati jedan objekt tipa A_2 i 4 objekta tipa A_3* uz maksimalnu dobit 52 n.j.

<u>*Rješenje pomoću programa Winqsb*</u>: Navodimo samo parametre prema kojima se rješenje ovoga problema razlikuje od prethodno navedenih rješenja:

- *Problem Title*: Primjer 1-4-2
- *Number of Items*: 3
- *Item Identification*: A1, A2, A3
- *Units Available*: 3, 18, 5 (jer se može montirati najviše 3 objekta tipa A_1 i najviše 5 objekata tipa A_3)

- *Unit Capacity Required*: 4, 6, 3
- *Return Function* (*X*: *Item ID*): 8*X, 12*X, 10*X
- *Capacity* = 18

Rješavanjem problema u drugom stupcu (*Decision Quantity*) dobivamo vrijednosti 0, 1 i 4, pa "očitavamo" da treba montirati točno 1 objekt tipa A_2 i 4 objekta tipa A_3. Ukupna optimalna dobit od montaže svih triju objekata (*Total Return Value*) iznosi 52 n.j.

Primjedba: U ovome je slučaju zgodno provesti *što – ako* analizu, pa npr. istražiti kako će se promijeniti optimalni plan odlučimo li montirati točno jedan objekt tipa A_1. U tu svrhu iskoristimo rezultat dobiven programom *Winqsb*. Kliknemo lijevom tipkom miša na ikonicu *What If*, pa u presjeku retka *A1* i stupca *Preset Decision* upišimo 1. Ovim smo zapravo *unaprijed* definirali odluku da želimo montirati točno jedan objekt tipa A_1.[19] Iz tablice na desnoj strani toga okvira "očitamo" da, uz navedeni dodatni uvjet, treba montirati jedan objekt tipa A_1 i četiri objekta tipa A_3. Dakle, postavljanje dodatnoga uvjeta izazvalo je smanjenje optimalne vrijednosti varijable x_2, a nije utjecalo na optimalnu vrijednost varijable x_3.

Primjer 3. Brodom treba prevesti dvije vrste robe: *A* i *B*. Dobit po jednom komadu robe *A* iznosi 40 n.j., a dobit po jednom komadu robe *B* 60 n.j. Jedan komad robe *B* zauzima 5 kub. jed. brodskoga prostora, dok roba *A* zauzima brodski prostor u ovisnosti o količini utovarene robe prema sljedećoj tablici:

količina robe A [kom.]	0	1	2	3	4	5	6
potreban obujam brodskoga prostora [kub. jed.]	0	5	8	10	12	15	18

Obujam dijela broda namijenjenoga za prijevoz robe iznosi 20 kub. jed. Komadi obiju vrsta robe su nedjeljivi. Treba napraviti optimalan plan utovara robe *A* i *B* tako da se ostvari maksimalna dobit.

a) Odredimo matematički model promatranoga problema.
b) Nađimo optimalan plan utovara broda i iznos maksimalne dobiti.

Matematički model: Neka je x_A količina utovarene robe vrste *A*, a x_B količina utovarene robe vrste *B*. Tada ukupna dobit ostvarena prijevozom robe vrste *A* iznosi $40 \cdot x_A$ n.j., a ukupna dobit ostvarena prijevozom robe vrste *B* iznosi $60 \cdot x_B$ n.j. Prema tome, ukupna dobit ostvarena prijevozom obiju vrsta roba iznosi

$$f(x_A, x_B) = 40 \cdot x_A + 60 \cdot x_B.$$

Označimo s $O_A(x)$ funkciju koja količini robe *A* pridružuje potreban obujam brodskoga prostora. Funkcija $O_A(x)$ je definirana tablično (točnije, gornjom tablicom) i najveća vrijednost njezina argumenta iznosi 6. To znači da se brodom može prevesti najviše 6 komada robe *A*, pa zbog prirodne cjelobrojnosti vrijednosti x_A vrijedi uvjet:

[19] Istaknimo zasebno: *Što – ako* analiza *ne* daje optimalna rješenja problema koji sadrže uvjet tipa "*barem jedan…*". Opcija *Preset Decision* zahtijeva unos unaprijed zadane *točne* vrijednosti varijabli, a ne unose najmanje ili najveće vrijednosti bilo koje od varijabli.

$$x_A \in [6]_0.$$

S druge strane, na količinu robe B nemamo nikakav početni uvjet osim prirodnoga uvjeta

$$x_B \in \mathbf{N}_0.$$

Preostaje još uvjetom iskazati podatak da ukupan obujam utovarene robe ne može biti strogo veći od obujma dijela broda namijenjenoga za prijevoz robe. Ukupan obujam x_A komada robe A iznosi $O_A(x_A)$, a ukupan obujam x_B komada robe B iznosi $5 \cdot x_B$, pa mora vrijediti nejednakost:

$$O_A(x_A) + 5 \cdot x_B \leq 20.$$

Dakle, traženi matematički model promatranoga problema je:

$$\text{maksimizirati } f(x_A, x_B) = 40 \cdot x_A + 60 \cdot x_B$$

pod uvjetima

$$O_A(x_A) + 5 \cdot x_B \leq 20$$

$$x_A \in [6]_0,$$

$$x_B \in \mathbf{N}_0.$$

<u>*Rješenje*</u>: Odmah napomenimo da, zbog tabličnoga definiranja funkcije ovisnosti obujma brodskoga prostora o količini robe A, ovaj problem **nije** moguće riješiti primjenom programa *Winqsb* jer taj program zahtijeva definiranje navedene funkcije analitičkim izrazom (formulom). Stavimo $S = [20]_0$, pa za svaki $s \in S$ računajmo vrijednosti funkcija:

$$f_1(s) = \max_{0 \leq O_A(x_A) \leq s} (40 \cdot x_A)$$

$$f_2(s) = \max_{0 \leq 5 \cdot x_B \leq s} \left[60 \cdot x_B + f_1(s - 5 \cdot x_B)\right] = \max_{0 \leq x_B \leq \frac{s}{5}} \left[60 \cdot x_B + f_1(s - 5 \cdot x_B)\right]$$

Funkcija $f_1(s)$ računa optimalnu dobit ako se svih s kub. jed. prostora namijenjenoga za prijevoz robe iskoristi isključivo za prijevoz robe vrste A, dok funkcija $f_2(s)$ računa optimalnu dobit ako se taj prostor iskoristi za prijevoz obiju vrsta roba. Tako redom dobivamo:

$f_1(0) = \max\limits_{0 \leq O_A(x_A) \leq 0} (40 \cdot x_A) = 40 \cdot 0 = 0;$ $\quad f_1(1) = \max\limits_{0 \leq O_A(x_A) \leq 1} (40 \cdot x_A) = 40 \cdot 0 = 0;$

$f_1(2) = \max\limits_{0 \leq O_A(x_A) \leq 2} (40 \cdot x_A) = 40 \cdot 0 = 0;$ $\quad f_1(3) = \max\limits_{0 \leq O_A(x_A) \leq 3} (40 \cdot x_A) = 40 \cdot 0 = 0;$

$f_1(4) = \max\limits_{0 \leq O_A(x_A) \leq 4} (40 \cdot x_A) = 40 \cdot 0 = 0;$ $\quad f_1(5) = \max\limits_{0 \leq O_A(x_A) \leq 5} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1\} = 40;$

$f_1(6) = \max\limits_{0 \leq O_A(x_A) \leq 6} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1\} = 40;$

$f_1(7) = \max\limits_{0 \leq O_A(x_A) \leq 7} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1\} = 40;$

$f_1(8) = \max\limits_{0 \leq O_A(x_A) \leq 8} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2\} = 80;$

$f_1(9) = \max\limits_{0 \le O_A(x_A) \le 9} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2\} = 80;$

$f_1(10) = \max\limits_{0 \le O_A(x_A) \le 10} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3\} = 120;$

$f_1(11) = \max\limits_{0 \le O_A(x_A) \le 11} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3\} = 120;$

$f_1(12) = \max\limits_{0 \le O_A(x_A) \le 12} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4\} = 160;$

$f_1(13) = \max\limits_{0 \le O_A(x_A) \le 13} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4\} = 160;$

$f_1(14) = \max\limits_{0 \le O_A(x_A) \le 14} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4\} = 160;$

$f_1(15) = \max\limits_{0 \le O_A(x_A) \le 15} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5\} = 200;$

$f_1(16) = \max\limits_{0 \le O_A(x_A) \le 16} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5\} = 200;$

$f_1(17) = \max\limits_{0 \le O_A(x_A) \le 17} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5\} = 200;$

$f_1(18) = \max\limits_{0 \le O_A(x_A) \le 18} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5, 40 \cdot 6\} = 240;$

$f_1(19) = \max\limits_{0 \le O_A(x_A) \le 18} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5, 40 \cdot 6\} = 240;$

$f_1(20) = \max\limits_{0 \le O_A(x_A) \le 18} (40 \cdot x_A) = \max\{40 \cdot 0, 40 \cdot 1, 40 \cdot 2, 40 \cdot 3, 40 \cdot 4, 40 \cdot 5, 40 \cdot 6\} = 240.$

Tako dobivamo sljedeću tablicu:

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f_1(s)$	0	0	0	0	0	40	40	40	80	80	120	120	160	160	160	200	200	200
x_A^*	0	0	0	0	0	1	1	1	2	2	3	3	4	4	4	5	5	5

s	18	19	20
$f_1(s)$	240	240	240
x_A^*	6	6	6

Preostaje izračunati vrijednosti funkcije $f_2(s)$:

$f_2(0) = \max\limits_{0 \le x_B \le 0} \left[60 \cdot x_B + f_1(0 - 5 \cdot x_B)\right] = 60 \cdot 0 + f_1(0) = 0 + 0 = 0;$

$f_2(1) = \max\limits_{0 \le x_B \le \frac{1}{5}} \left[60 \cdot x_B + f_1(1 - 5 \cdot x_B)\right] = 60 \cdot 0 + f_1(1) = 0 + 0 = 0;$

$f_2(2) = \max\limits_{0 \le x_B \le \frac{2}{5}} \left[60 \cdot x_B + f_1(2 - 5 \cdot x_B)\right] = 60 \cdot 0 + f_1(2) = 0 + 0 = 0;$

$f_2(3) = \max\limits_{0 \le x_B \le \frac{3}{5}} \left[60 \cdot x_B + f_1(3 - 5 \cdot x_B)\right] = 60 \cdot 0 + f_1(3) = 0 + 0 = 0;$

$f_2(4) = \max\limits_{0 \le x_B \le \frac{4}{5}} \left[60 \cdot x_B + f_1(4 - 5 \cdot x_B)\right] = 60 \cdot 0 + f_1(4) = 0 + 0 = 0;$

$f_2(5) = \max\limits_{0 \le x_B \le 1} \left[60 \cdot x_B + f_1(5 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(5), 60 \cdot 1 + f_1(0)\} = \max\{0 + 40, 60 +$
$+ 0\} = 60;$

$f_2(6) = \max\limits_{0 \le x_B \le \frac{6}{5}} \left[60 \cdot x_B + f_1(6 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(6),\ 60 \cdot 1 + f_1(1)\} = \max\{0 + 40,\ 60 + 0\} = 60;$

$f_2(7) = \max\limits_{0 \le x_B \le \frac{7}{5}} \left[60 \cdot x_B + f_1(7 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(7),\ 60 \cdot 1 + f_1(2)\} = \max\{0 + 40,\ 60 + 0\} = 60;$

$f_2(8) = \max\limits_{0 \le x_B \le \frac{8}{5}} \left[60 \cdot x_B + f_1(8 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(8),\ 60 \cdot 1 + f_1(3)\} = \max\{0 + 80,\ 60 + 0\} = 80;$

$f_2(9) = \max\limits_{0 \le x_B \le \frac{9}{5}} \left[60 \cdot x_B + f_1(9 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(9),\ 60 \cdot 1 + f_1(4)\} = \max\{0 + 80,\ 60 + 0\} = 80;$

$f_2(10) = \max\limits_{0 \le x_B \le 2} \left[60 \cdot x_B + f_1(10 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(10),\ 60 \cdot 1 + f_1(5),\ 60 \cdot 2 + f_2(0)\} = \max\{0 + 120,\ 60 + 40,\ 120 + 0\} = 120;$

$f_2(11) = \max\limits_{0 \le x_B \le \frac{11}{5}} \left[60 \cdot x_B + f_1(11 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(11),\ 60 \cdot 1 + f_1(6),\ 60 \cdot 2 + f_2(1)\} = \max\{0 + 120,\ 60 + 40,\ 120 + 0\} = 120;$

$f_2(12) = \max\limits_{0 \le x_B \le \frac{12}{5}} \left[60 \cdot x_B + f_1(12 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(12),\ 60 \cdot 1 + f_1(7),\ 60 \cdot 2 + f_2(2)\} = \max\{0 + 160,\ 60 + 40,\ 120 + 0\} = 160;$

$f_2(13) = \max\limits_{0 \le x_B \le \frac{12}{5}} \left[60 \cdot x_B + f_1(13 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(13),\ 60 \cdot 1 + f_1(8),\ 60 \cdot 2 + f_2(3)\} = \max\{0 + 160,\ 60 + 80,\ 120 + 0\} = 160;$

$f_2(14) = \max\limits_{0 \le x_B \le \frac{14}{5}} \left[60 \cdot x_B + f_1(14 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(14),\ 60 \cdot 1 + f_1(9),\ 60 \cdot 2 + f_2(4)\} = \max\{0 + 160,\ 60 + 80,\ 120 + 0\} = 160;$

$f_2(15) = \max\limits_{0 \le x_B \le 3} \left[60 \cdot x_B + f_1(15 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(15),\ 60 \cdot 1 + f_1(10),\ 60 \cdot 2 + f_2(5),\ 60 \cdot 3 + f_2(0)\} = \max\{0 + 200,\ 60 + 120,\ 120 + 40,\ 180 + 0\} = 200;$

$f_2(16) = \max\limits_{0 \le x_B \le \frac{16}{5}} \left[60 \cdot x_B + f_1(16 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(16),\ 60 \cdot 1 + f_1(11),\ 60 \cdot 2 + f_2(6),\ 60 \cdot 3 + f_2(1)\} = \max\{0 + 200,\ 60 + 120,\ 120 + 40,\ 180 + 0\} = 200;$

$f_2(17) = \max\limits_{0 \le x_B \le \frac{17}{5}} \left[60 \cdot x_B + f_1(17 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(17),\ 60 \cdot 1 + f_1(12),\ 60 \cdot 2 + f_2(7),\ 60 \cdot 3 + f_2(2)\} = \max\{0 + 200,\ 60 + 160,\ 120 + 40,\ 180 + 0\} = 220;$

$f_2(18) = \max\limits_{0 \le x_B \le \frac{18}{5}} \left[60 \cdot x_B + f_1(18 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(18),\ 60 \cdot 1 + f_1(13),\ 60 \cdot 2 + f_2(8),\ 60 \cdot 3 + f_2(3)\} = \max\{0 + 240,\ 60 + 160,\ 120 + 80,\ 180 + 0\} = 240;$

$f_2(19) = \max\limits_{0 \le x_B \le \frac{19}{5}} \left[60 \cdot x_B + f_1(19 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(19),\ 60 \cdot 1 + f_1(14),\ 60 \cdot 2 + f_2(9),\ 60 \cdot 3 + f_2(4)\} = \max\{0 + 240,\ 60 + 160,\ 120 + 80,\ 180 + 0\} = 240;$

$f_2(20) = \max\limits_{0 \le x_B \le 4} \left[60 \cdot x_B + f_1(20 - 5 \cdot x_B)\right] = \max\{60 \cdot 0 + f_1(20),\ 60 \cdot 1 + f_1(15),\ 60 \cdot 2 + f_2(10),$

$60 \cdot 3 + f_2(5),\ 60 \cdot 4 + f_2(0)\} = \max\{0 + 240,\ 60 + 200,\ 120 + 120,\ 180 + 40,\ 240 + 0\} = 260.$

Tako dobivamo sljedeću tablicu:

s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f_1(s)$	0	0	0	0	0	40	40	40	80	80	120	120	160	160	160	200	200	200
x_A^*	0	0	0	0	0	1	1	1	2	2	3	3	4	4	4	5	5	5
$f_2(s)$	0	0	0	0	0	60	60	60	80	80	120	120	160	160	160	200	200	220
x_B^*	0	0	0	0	0	1	1	1	0	0	0, 2	0, 2	0	0	0	0	0	1

s	18	19	20
$f_1(s)$	240	240	240
x_A^*	6	6	6
$f_2(s)$	240	240	260
x_B^*	0	0	1

Prema tome, optimalna dobit je $f_2(20) = 260$ i postiže se za $x_B^* = 1$. Odgovarajuća optimalna vrijednost funkcije f_1 je $f_1(20 - 5 \cdot 1) = f_1(15) = 200$ i postiže se za $x_A^* = 5$. Dakle, optimalan plan prijevoza robe je: prevesti 5 komada robe vrste A i 1 komad robe vrste B.

Primjer 4. Zrakoplovom nosivosti 83 tone treba prevesti tri vrste robe čije su jedinične mase i jedinične dobiti od prijevoza navedeni u sljedećoj tablici.

vrsta robe	*A*	*B*	*C*
jedinična masa [t]	24	22	16
jedinična dobit [n.j.]	96	85	50

Komadi svih triju vrsta robe su nedjeljivi. Treba napraviti plan utovara svih triju vrsta robe tako da se ostvari maksimalna dobit.

a) Odredimo matematički model promatranoga problema.
b) Nađimo optimalan plan utovara i odgovarajuću maksimalnu dobit.

Matematički model: Neka su x_A količina utovarene robe vrste A, x_B količina utovarene robe vrste B i x_C količina utovarene robe vrste C. Tada ukupna dobit od prijevoza robe vrste A iznosi $96 \cdot x_A$, ukupna dobit od prijevoza robe vrste B iznosi $85 \cdot x_B$, a ukupna dobit od prijevoza robe vrste C iznosi $50 \cdot x_C$. Stoga je ukupna dobit od prijevoza svih triju vrsta roba

$$f(x_A, x_B, x_C) = 96 \cdot x_A + 85 \cdot x_B + 50 \cdot x_C.$$

Nadalje, ukupna masa x_A komada robe vrste A iznosi $24 \cdot x_A$ tona, ukupna masa x_B komada robe vrste B iznosi $22 \cdot x_B$ tona, a ukupna masa x_C komada robe vrste C iznosi $16 \cdot x_C$ tona. Ukupna masa svih triju vrsta robe ne smije biti strogo veća od nosivosti zrakoplova, što znači da mora vrijediti uvjet:

$$24 \cdot x_A + 22 \cdot x_B + 16 \cdot x_C \leq 83.$$

Prema prirodi problema, sve tri vrijednosti x_A, x_B i x_C su nenegativni cijeli brojevi. Dakle, traženi matematički model promatranoga problema je:

$$\text{maksimizirati } f(x_A, x_B, x_C) = 96 \cdot x_A + 85 \cdot x_B + 50 \cdot x_C$$

pod uvjetima

$$24 \cdot x_A + 22 \cdot x_B + 16 \cdot x_C \leq 83$$

$$x_A,\ x_B,\ x_C \in \mathbf{N}_0.$$

<u>*Rješenje pomoću programa Winqsb*</u>: "Klasično" bismo definirali $S = [83]_0$ i za svaki $s \in S$ određivali vrijednosti sljedećih triju realnih funkcija:

$$f_1(s) = \max_{0 \leq 24 \cdot x_A \leq s} (96 \cdot x_A) = \max_{0 \leq x_A \leq \frac{s}{24}} (96 \cdot x_A),$$

$$f_2(s) = \max_{0 \leq 22 \cdot x_B \leq s} \left[85 \cdot x_B + f_1(s - 22 \cdot x_B)\right] = \max_{0 \leq x_B \leq \frac{s}{22}} \left[85 \cdot x_B + f_1(s - 22 \cdot x_B)\right],$$

$$f_3(s) = \max_{0 \leq 16 \cdot x_C \leq s} \left[50 \cdot x_C + f_2(s - 16 \cdot x_C)\right] = \max_{0 \leq x_C \leq \frac{s}{16}} \left[50 \cdot x_C + f_2(s - 16 \cdot x_C)\right].$$

Prva funkcija računa optimalnu dobit ako se prevozi isključivo roba vrste *A*, i to s maksimalnom nosivošću *s* [tona]. Druga funkcija računa optimalnu dobit ako se prevoze isključivo robe vrste *A* i vrste *B*, opet s maksimalnom nosivošću *s* [tona]. Treća funkcija računa optimalnu dobit ako se prevoze sve tri vrste robe s maksimalnom nosivošću *s* [tona].

Izračun svih vrijednosti ovih funkcija je prilično mukotrpan i spor iako se određenim "trikovima" on može djelomično ubrzati. Zbog toga ćemo ovaj problem riješiti koristeći potprogram *DP* računalnoga programa *Winqsb*. Navodimo samo parametre prema kojima se rješenje ovoga primjera razlikuje od prethodno navedenih rješenja:

- *Problem Title*: Primjer 1-4-4
- *Number of Items*: 3
- *Item Identification*: A, B, C
- *Units Available*: opet postupimo lukavo, pa u skupu **N** riješimo nejednadžbe $24 \cdot x_A \leq 83$, $22 \cdot x_B \leq 83$ i $16 \cdot x_C \leq 83$. Dobivamo: $x_A \leq 3$, $x_B \leq 3$ i $x_C \leq 5$, pa u ovaj stupac upisujemo 3, 3, 5.
- *Unit Capacity Required*: 24, 22, 16
- *Return Function* (*X*: *Item ID*): 96*X, 85*X, 50*X
- *Capacity* = 83

Nakon pokretanja procedure *Solve and Analyze* u stupcu *Decision Quantity* "očitavamo" vrijednosti 0, 3 i 1. Dakle, optimalan plan utovara je: *utovariti* 3 *komada robe B i* 1 komad *robe C*. Ukupna *optimalna dobit* (*Total Return Value*) iznosi 305 n.j.

I ovdje možemo provesti *što–ako* analizu, pa promotriti kako na dobiveno optimalno rješenje utječe npr. dodatni uvjet da se mora prevesti *točno* jedan komad robe *A*. U tom je slučaju

optimalni plan utovara: *utovariti po* 1 *komad robe svake od vrsta A* i *B*, *te točno* 2 *komada robe vrste C*. Dakle, postavljanje dodatnoga uvjeta uzrokuje promjenu obiju optimalnih vrijednosti: optimalna vrijednost x_B se smanjila, dok se optimalna vrijednost x_C povećala.

Primjer 5. Zrakoplovom nosivosti 50 tona na određenoj se liniji mogu prevoziti putnici, poštanske pošiljke i ostali teret. Prosječna masa i dobit ostvarena po jedinici transporta navedeni su u donjoj tablici.

	1 *putnik*	1 *poštanski paket*	1 *paket ostaloga tereta*
prosječna masa [t]	0.085	0.2	0.9
dobit [000 n.j.]	2	2.5	4.5

U zrakoplov se može smjestiti najviše 120 putnika. Treba odrediti optimalnu strukturu transporta tako da se ostvari maksimalna dobit. Formirajmo odgovarajući matematički model, pa odredimo traženu optimalnu strukturu.

Matematički model: Neka su x_1 broj putnika, x_2 broj poštanskih paketa i x_3 broj paketa drugoga tereta koji će se prevoziti zrakoplovom. Ukupna dobit ostvarena prijevozom x_1 putnika iznosi $2 \cdot x_1$ [000 n.j.], ukupna dobit ostvarena prijevozom x_2 poštanskih paketa iznosi $2.5 \cdot x_2$ [000 n.j.], a ukupna dobit ostvarena prijevozom x_3 paketa ostaloga tereta iznosi $4.5 \cdot x_3$ [000 n.j.]. Stoga ukupna dobit nastala prijevozom svih triju kategorija tereta iznosi:

$$f(x_1, x_2, x_3) = 2 \cdot x_1 + 2.5 \cdot x_2 + 4.5 \cdot x_3.$$

Nadalje, ukupna masa svih x_1 putnika iznosi $0.085 \cdot x_1$ tona, ukupna masa x_2 poštanskih paketa iznosi $0.2 \cdot x_2$ tona, a ukupna masa x_3 paketa ostaloga tereta iznosi $0.9 \cdot x_3$ tona. Budući da ukupna masa svih triju vrsta tereta ne smije biti strogo veća od nosivosti zrakoplova, mora vrijediti nejednakost:

$$0.085 \cdot x_1 + 0.2 \cdot x_2 + 0.9 \cdot x_3 \leq 50.$$

Iz činjenice da se u zrakoplov može smjestiti najviše 120 putnika slijedi nejednakost
:

$$x_1 \leq 120.$$

Sve tri vrijednosti moraju biti nenegativni cijeli brojevi, tj. vrijedi skupovna relacija:

$$x_i \in \mathbf{N}_0, \text{ za svaki } i \in [3].$$

Tako matematički model promatranoga problema glasi:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 2 \cdot x_1 + 2.5 \cdot x_2 + 4.5 \cdot x_3$$

pod uvjetima

$$0.085 \cdot x_1 + 0.2 \cdot x_2 + 0.9 \cdot x_3 \leq 50$$

$$x_1 \in [120]_0$$

$$x_i \in \mathbf{N}_0, \text{ za svaki } i = 1, 2, 3.$$

Rješenje pomoću programa Winqsb: Navodimo samo parametre prema kojima se rješenje ovoga primjera razlikuje od prethodno navedenih rješenja:

- *Problem Title*: Primjer 1-4-5
- *Number of Items*: 3
- *Item Identification*: putnici, poštanski paket, ostali teret
- *Units Available*: opet postupimo lukavo, pa u skupu **N** riješimo nejednadžbe $0.2 \cdot x_2 \leq 50$ i $0.9 \cdot x_3 \leq 50$ i dobivamo: $x_2 \leq 250$, $x_3 \leq 55$. Stoga u ovaj stupac upisujemo: 120, 250, 55.
- *Unit Capacity Required*: 0.085, 0.2, 0.9
- *Return Function* (*X*: *Item ID*): 2*X, 2.5*X, 4.5*X
- *Capacity* = 50

Nakon pokretanja procedure *Solve and Analyze* u stupcu *Decision Quantity* "očitavamo" vrijednosti 120, 199 i 0. Dakle, optimalnu strukturu transporta tvori 120 putnika i 199 poštanskih paketa uz ukupnu optimalnu dobit (*Total Return Value*) 737,50 [000 n.j.], odnosno 737 500 n.j.

Postavimo li dodatni uvjet da moramo prevesti i točno jedan paket ostaloga tereta, *što – ako* analizom dobivamo sljedeću optimalnu strukturu transporta: 120 putnika, 194 poštanska paketa i 1 paket ostaloga tereta. Dakle, postavljanje dodatnoga uvjeta nije utjecalo na optimalan broj putnika u optimalnoj strukturi transporta, ali je smanjilo optimalan broj poštanskih paketa u optimalnoj strukturi transporta.

Zadaci za vježbu:

1. Brodski teret tvore četiri različita artikla. Jedinična masa i jedinična dobit za svaki pojedini artikl navedeni su u sljedećoj tablici.

artikl	*jedinična masa* [tona]	*jedinična dobit* [n.j.]
A	2	50
B	4	120
C	5	170
D	3	80

Ukupna masa brodskoga tereta može biti najviše 9 tona. Treba naći optimalan plan utovara kojim se postiže maksimalna dobit.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalan plan utovara i odgovarajuću maksimalnu dobit.
c) Ukoliko je optimalna količina nekoga artikla jednaka nuli, istražite kako njezino povećanje za 1 utječe na optimalne količine ostalih artikala.

2. Brodom se prevoze dvije vrste robe: *A* i *B*. Pritom se po komadu prevezene robe *A* ostvaruje dobit od 40 n.j., a po komadu prevezene robe *B* dobit od 60 n.j. Nadalje, 1 komad robe *A* ima obujam 3 kub. jed., a 1 komad robe *B* obujam 5 kub. jed. Obujam brodskoga prostora predviđenoga za prijevoz obiju vrsta robe iznosi 30 kub. jed. Treba naći optimalan

plan utovara broda navedenim vrstama robe tako da dobit ostvarena od prijevoza obiju vrsta roba bude maksimalna.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalan plan utovara i odgovarajuću maksimalnu dobit.
c) Riješite **b)** podzadatak uz uvjet da se brodom može prevesti najviše 6 komada robe *A*.

3. Zrakoplovom nosivosti 85 tona treba prevesti tri vrste robe čije su jedinične mase i jedinične dobiti navedene u sljedećoj tablici:

vrsta robe	*A*	*B*	*C*
jedinična masa [tona]	14	12	13
jedinična dobit [000 €]	64	56	60

Komade svih triju vrsta robe nije moguće razdijeliti na manje dijelove. Treba napraviti optimalan plan utovara robe tako da ukupna dobit bude maksimalna.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalan plan utovara i odgovarajuću maksimalnu dobit.
c) Ukoliko je optimalna količina neke vrste robe jednaka nuli, istražite kako njezino povećanje za 1 utječe na optimalne količine ostalih vrsta roba.

4. Tvrtka izvodi montažu dvaju tipova objekata: *A* i *B*. Montažom jednoga objekta tipa *A* ostvaruje se dobit od 10 n.j., a montažom jednoga objekta tipa *B* dobit od 12 n.j. Pri istodobnoj montaži obiju tipova objekata može se koristiti najviše 20 komada posebne vrste opreme, pri čemu za montažu jednoga objekta tipa *A* treba 4 komada opreme, a za montažu jednoga objekta tipa *B* 3 komada opreme. Tvrtka želi istodobnim montiranjem objekata *A* i *B* ostvariti maksimalnu dobit.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalan plan montaže i odgovarajuću optimalnu dobit.
c) Ispitajte kako na optimalan plan montaže utječe poseban ugovor kojim se tvrtka obvezuje montirati barem 2, a najviše 5 objekata tipa *A*.

5. Tvrtka "*Gulikožić* d.o.o." raspolaže s kamionom nosivosti 45 jed. Tim kamionom treba prevesti dvije vrste tereta: *A* i *B*. Masa jednoga komada tereta *A* iznosi 5 jed., a dobit ostvarena njegovim transportom iznosi 6 n.j. Masa jednoga komada tereta *B* iznosi 3 jed., a dobit ostvarena njegovim transportom iznosi 4 n.j. Tvrtka želi prevesti što više tereta obiju vrsta tako da ukupna ostvarena dobit bude maksimalna.

a) Formirajte matematički model promatranoga problema.
b) Nađite optimalan plan prijevoza i odgovarajuću optimalnu ostvarenu dobit.
c) Ukoliko je optimalna količina neke vrste tereta jednaka nuli, istražite kako njezino povećanje za 1 utječe na optimalnu količinu druge vrste tereta.
d) Riješite podzadatke **a)** i **b)** uz dodatni uvjet da se kamionom može prevesti najviše 5 komada tereta *B*.

1.5. Problem optimalne zamjene opreme

Problem optimalne zamjene opreme jedan je od osnovnih problema u cjelokupnoj industriji. Sastoji se u određivanju optimalnoga trenutka zamjene stare opreme novom pod uvjetima da se opseg proizvodnje poveća, a troškovi proizvodnje umanje tako da se time potpuno nadoknade troškovi kupnje nove opreme, gubici uzrokovani zastojem u proizvodnji zbog uvođenja nove opreme, te troškovi obuke djelatnika za rad na novoj opremi.

Dakle, zadatak je odrediti *optimalnu politiku* modernizacije i zamjene opreme uvažavajući pritom različite uvjete vezane za tekuće održavanje, karakteristike proizvodnje i predviđeni tehnološki razvoj. Ovakav proces očito zahtijeva analizu u više faza, pa se time dinamičko programiranje nameće kao prikladan način rješavanja ovakvih problema.

Neke tipove problema zamjene opreme i načine njihova rješavanja ilustrirat ćemo na sljedećim primjerima.

Primjer 1. Utvrđena poslovna politika predviđa korištenje automobila u sljedećih $n = 5$ godina, pa je na početku prve godine razdoblja planiranja kupljen novi automobil po nabavnoj cijeni od $p = 12$ n.j.[20] Radi jednostavnosti, pretpostavljamo da je nabavna cijena automobila stalna tijekom cijeloga razdoblja planiranja. Godišnji neto-troškovi održavanja automobila (c_i) i likvidacijska vrijednost[21] automobila (s_i) iskazani su kao funkcija starosti automobila:

starost automobila (i) [god.]	0	1	2	3	4	5
godišnji neto-trošak održavanja (c_i) [n.j.]	2	4	5	9	12	12
likvidacijska vrijednost (s_i) [n.j.]	–	7	6	2	1	0

Niz $(c_i)_{i \in \mathbf{N}}$ je nužno strogo rastući, tj. povećanjem starosti automobila povećavaju se i godišnji troškovi njegova održavanja. Iz toga razloga nakon određenoga vremena može biti ekonomski isplativije rashodovati dotada korišteni automobil i kupiti novi. S druge je strane niz $(s_i)_{i \in \mathbf{N}}$ nužno strogo padajući, tj. povećanjem starosti automobila smanjuje se njegova likvidacijska vrijednost.

<u>*Politika zamjene*</u> u ovome je slučaju donošenje odluke treba li zadržati postojeći automobil ili ga rashodovati i kupiti novi automobil. Navedeno se odlučivanje vrši početkom svake pojedine godine. Najjednostavniji primjeri politika zamjene su:

P_1) zamjena automobila, tj. rashodovanje dotada korištenoga automobila i kupnja novoga početkom svake pojedine godine (tj. početkom druge, treće, četvrte i pete godine);

P_2) zadržavanje postojećega automobila sve do kraja pete godine.

U našem primjeru početkom prve godine nemamo što odlučivati: kupljen je novi automobil. Stoga se odluke donose početkom svake od sljedeće četiri godine. Budući da početkom svake

[20] Radi jednostavnosti, pretpostavljamo da su u navedenu cijenu uračunati svi pripadajući porezi, prirezi itd.
[21] Likvidacijska vrijednost nekoga sredstva je vrijednost rashodovanoga stalnoga sredstva koja se dobije kad se od tržišne cijene toga sredstva odbiju troškovi rashodovanja.

od tih godina možemo donijeti točno dvije različite odluke, prema pravilu umnoška[22] zaključujemo da postoji ukupno $2^{n-1} = 2^4 = 16$ međusobno različitih politika zamjene. Stoga je od interesa odrediti najbolju među njima, tj. odrediti *optimalnu politiku zamjene*.

<u>*Optimalna politika zamjene*</u> je politika zamjene koja osigurava najmanje ukupne neto-troškove uporabe automobila tijekom svih pet godina. U promatranom primjeru to znači ispitati svih 16 međusobno različitih politika zamjene i među njima odrediti najbolju. No, takva je metoda za iole veće vrijednosti n općenito vrlo spora i neefikasna, pa se stoga nameće problem iznalaženja dovoljno efikasnih metoda i algoritama za određivanje optimalne politike. U tu se svrhu primjenjuju metode i tehnike dinamičkoga programiranja.

Primjer 1. može se riješiti "klasično", tj. definiranjem odgovarajućih rekurzivnih relacija, ali mi ćemo ga ovdje riješiti svođenjem na problem najkraćega puta u grafu opisan u točki 1.1.

Matematički model koji ćemo koristiti je *potpuni usmjereni graf* [23]. Vrhovi toga grafa predstavljat će *početke godina*, tj. vrh i označavat će nam početak godine i. Budući da se korištenje automobila planira u sljedećih $n = 5$ godina, graf će imati ukupno $n + 1 = 5 + 1 = 6$ vrhova: 1, 2, 3, 4, 5 i 6.

Nadalje, za svaki uređeni par (i, j) takav da je $i < j$ vrhove i i j spojit ćemo usmjerenim bridom (lukom) čiji je početak u vrhu i, a kraj u vrhu j. Ukupan broj međusobno različitih usmjerenih bridova u tako dobivenom modelu jednak je $\binom{n}{2} = \binom{6}{2} = 15$ jer su svaka dva međusobno različita vrha spojena točno jednim lukom.

Preostaje nam definirati *težinu* svakoga od dobivenih lukova. Budući da želimo minimizirati ukupne neto-troškove, definirat ćemo težinu w_{ij} kao:

w_{ij} = (nabavna cijena automobila na početku godine i) + (troškovi održavanja u godinama i, $i + 1, \ldots, j - 1$) – (likvidacijska vrijednost automobila na početku godine j).

Napišimo analitički izraz za izračunavanje težina w_{ij}. Nabavnu cijenu na početku godine i označit ćemo s p_i. Na početku godine i automobil je nov (star 0 godina) pa su troškovi održavanja u godini i jednaki troškovima održavanja novoga automobila, tj. c_0. Na početku godine $i + 1$ automobil je star 1 godinu, pa su troškovi održavanja u godini $i + 1$ jednaki troškovima održavanja automobila staroga 1 godinu, tj. c_1. Na početku godine $j - 1$ automobil je star ukupno $(j - 1) - i$ godina, pa su troškovi održavanja u godini $(j - 1)$ jednaki troškovima održavanja automobila staroga $(j - 1) - i$ godina, tj. c_{j-1-i}. Na početku godine j automobil je star ukupno $j - i$ godina, pa je njegova likvidacijska vrijednost jednaka likvidacijskoj vrijednosti automobila staroga $j - i$ godina, tj. s_{j-i}. Prema tome je:

$$w_{ij} = p_i + c_0 + c_1 + \ldots + c_{j-1-i} - s_{j-i} = p_i + \sum_{k=0}^{j-i-1} c_k - s_{j-i}.$$ [24]

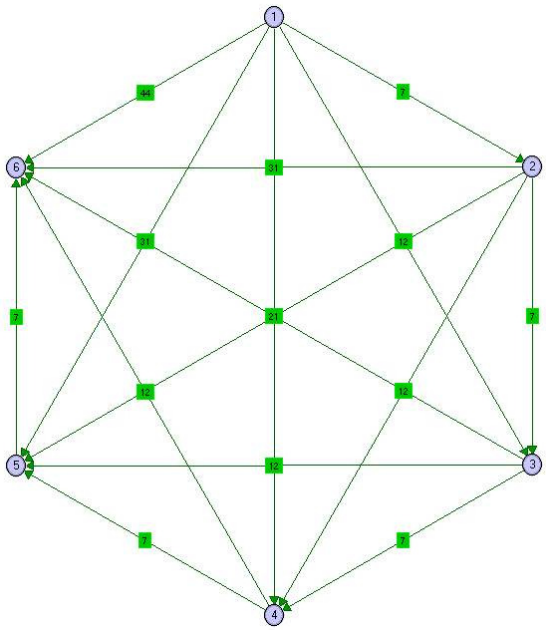
[22] *Pravilo zbroja* i *pravilo umnoška* temeljna su pravila kombinatorike (podsjetite se gradiva iz kolegija *Statistika* 2).
[23] Podsjetite se gradiva iz kolegija *Ekonomska matematika* 2.
[24] Ovdje prešutno pretpostavljamo da se kraj k – te godine podudara s početkom $(k + 1)$ – ve godine.

U našem je primjeru nabavna cijena automobila na početku svake godine ista, tj. vrijedi jednakost $p_1 = p_2 = p_3 = p_4 = p_5 = p = 12$. Tako redom dobivamo:

w_{12} = (nabavna cijena automobila na početku 1. godine) + (troškovi održavanja novoga automobila u 1. godini) – (likvidacijska vrijednost automobila staroga 1 godinu na početku 2. godine) = $p + c_0 - s_1 = 12 + 2 - 7 = 7$;
w_{13} = (nabavna cijena automobila na početku 1. godine) + [(troškovi održavanja novoga automobila u 1. godini) + (troškovi održavanja automobila staroga 1 godinu u 2. godini)] – – (likvidacijska vrijednost automobila staroga 2 godine na početku 3. godine) = $p + (c_0 + c_1) - s_2 = 12 + 2 + 4 - 6 = 12$;
w_{14} = (nabavna cijena automobila na početku 1. godine) + [(troškovi održavanja novoga automobila u 1. godini) + (troškovi održavanja automobila staroga 1 godinu u 2. godini) + (troškovi održavanja automobila staroga 2 godine u 3. godini)] – (likvidacijska vrijednost automobila staroga 3 godine na početku 4. godine) = $p + (c_0 + c_1 + c_2) - s_3 = 12 + 2 + 4 + 5 - 2 = 21$;
w_{15} = (nabavna cijena automobila na početku 1. godine) + [(troškovi održavanja novoga automobila u 1. godini) + (troškovi održavanja automobila staroga 1 godinu u 2. godini) + (troškovi održavanja automobila staroga 2 godine u 3. godini) + (troškovi održavanja automobila staroga 3 godine u 4. godini)] – (likvidacijska vrijednost automobila staroga 4 godine na početku 5. godine) = $p + (c_0 + c_1 + c_2 + c_3) - s_4 = 12 + 2 + 4 + 5 + 9 - 1 = 31$;
w_{16} = (nabavna cijena automobila na početku 1. godine) + [(troškovi održavanja novoga automobila u 1. godini) + (troškovi održavanja automobila staroga 1 godinu u 2. godini) + (troškovi održavanja automobila staroga 2 godine u 3. godini) + (troškovi održavanja automobila staroga 3 godine u 4. godini) + (troškovi održavanja automobila staroga 4 godine u 5. godini)] – (likvidacijska vrijednost automobila staroga 5 godina na početku 6. godine) = = $p + (c_0 + c_1 + c_2 + c_3 + c_4) - s_4 = 12 + 2 + 4 + 5 + 9 + 12 - 0 = 44$;
w_{23} = (nabavna cijena automobila na početku 2. godine) + (troškovi održavanja novoga automobila u 2. godini) – (likvidacijska vrijednost automobila staroga 1 godinu na početku 3. godine) = $p + c_0 - s_1 = 12 + 2 - 7 = 7$;
w_{24} = (nabavna cijena automobila na početku 2. godine) + [(troškovi održavanja novoga automobila u 2. godini) + (troškovi održavanja automobila staroga 1 godinu u 3. godini)] – – (likvidacijska vrijednost automobila staroga 2 godine na početku 4. godine) = $p + (c_0 + c_1) - s_2 = 12 + 2 + 4 - 6 = 12$;
w_{25} = (nabavna cijena automobila na početku 1. godine) + [(troškovi održavanja novoga automobila u 1. godini) + (troškovi održavanja automobila staroga 1 godinu u 2. godini) + (troškovi održavanja automobila staroga 2 godine u 3. godini)] – (likvidacijska vrijednost automobila staroga 3 godine na početku 4. godine) = $p + (c_0 + c_1 + c_2) - s_3 = 12 + 2 + 4 + 5 - 2 = 21$;
w_{26} = (nabavna cijena automobila na početku 2. godine) + [(troškovi održavanja novoga automobila u 2. godini) + (troškovi održavanja automobila staroga 1 godinu u 3. godini) + (troškovi održavanja automobila staroga 2 godine u 4. godini) + (troškovi održavanja automobila staroga 3 godine u 5. godini)] – (likvidacijska vrijednost automobila staroga 4 godine na početku 6. godine) = $p + (c_0 + c_1 + c_2 + c_3) - s_4 = 12 + 2 + 4 + 5 + 9 - 1 = 31$;
w_{34} = (nabavna cijena automobila na početku 4. godine) + (troškovi održavanja novoga automobila u 3. godini) – (likvidacijska vrijednost automobila staroga 1 godinu na početku 4. godine) = $p + c_0 - s_1 = 12 + 2 - 7 = 7$;
w_{35} = (nabavna cijena automobila na početku 3. godine) + [(troškovi održavanja novoga automobila u 3. godini) + (troškovi održavanja automobila staroga 1 godinu u 4. godini)] – – (likvidacijska vrijednost automobila staroga 2 godine na početku 5. godine) = $p + (c_0 + c_1) - s_2 = 12 + 2 + 4 - 6 = 12$;

w_{36} = (nabavna cijena automobila na početku 3. godine) + [(troškovi održavanja novoga automobila u 3. godini) + (troškovi održavanja automobila staroga 1 godinu u 4. godini) + (troškovi održavanja automobila staroga 2 godine u 5. godini)] – (likvidacijska vrijednost automobila staroga 3 godine na početku 6. godine) = $p + (c_0 + c_1 + c_2) - s_3 = 12 + 2 + 4 + 5 - 2 = 21$;
w_{45} = (nabavna cijena automobila na početku 4. godine) + (troškovi održavanja novoga automobila u 4. godini) – (likvidacijska vrijednost automobila staroga 1 godinu na početku 5. godine) = $p + c_0 - s_1 = 12 + 2 - 7 = 7$;
w_{46} = (nabavna cijena automobila na početku 4. godine) + [(troškovi održavanja novoga automobila u 4. godini) + (troškovi održavanja automobila staroga 1 godinu u 5. godini)] – (likvidacijska vrijednost automobila staroga 2 godine na početku 6. godine) = $p + (c_0 + c_1) - s_2 = 12 + 2 + 4 - 6 = 12$;
w_{56} = (nabavna cijena automobila na početku 5. godine) + (troškovi održavanja novoga automobila u 5. godini) – (likvidacijska vrijednost automobila staroga 1 godinu na početku 6. godine) = $p + c_0 - s_1 = 12 + 2 - 7 = 7$.

Za rješavanje primjera u nastavku koristimo program *Graph Magics*. Pokrenimo program i odmah odaberimo tip budućega grafa kao *usmjeren* lijevim klikom miša na ikonicu strelice neposredno desno od ikonice **C**. Sada generirajmo potpuni usmjereni graf sa 6 vrhova i pridružimo njegovim lukovima odgovarajuće težine. Dobivamo:



Slika 15. Model problema iz Primjera 1.

Sljedeći korak u rješavanju zadatka je određivanje najkraćega puta između vrha koji označava početak 1. godine (to je vrh 1) i vrha koji označava kraj 5. godine, odnosno početak 6. godine (to je vrh 6). Desnim klikom miša na vrh 1 i odabirom opcije *Set/Unset as Start Vertex* (*Source*) označimo taj vrh kao početni, a desnim klikom miša na vrh 6 i odabirom opcije *Set/Unset as End Vertex* (*Sink*) označimo taj vrh kao krajnji. Postupkom opisanim u točki 1.1. nalazimo da je najkraći put između vrhova 1 i 6:

1 – 2 – 4 – 6

i da je njegova težina jednaka 31. Preostaje nam jedino interpretirati dobiveni rezultat. U tu svrhu promotrimo unutrašnje vrhove dobivenoga najkraćega puta. To su 2 i 4, pa *automobil trebamo zamijeniti novim automobilom početkom* 2. *i početkom* 4. *godine*. Prema tome, optimalna politika zamjene glasi:

početak 2. *godine*: rashodovanje dotad korištenoga automobila i kupnja novoga;
početak 3. *godine*: zadržavanje dotad korištenoga automobila;
početak 4. *godine*: rashodovanje dotad korištenoga automobila i kupnja novoga;
početak 5. *godine*: zadržavanje dotad korištenoga automobila;
kraj 5. *godine/početak* 6. *godine*: rashodovanje dotad korištenoga automobila.

Rješavanje "klasičnim" putem daje još jedno rješenje:

1 – 3 – 5 – 6

U ovom slučaju *početkom* 3. *i početkom* 5. *godine treba kupiti novi automobil*, tj. optimalna politika zamjene glasi:

početak 2. *godine*: zadržavanje dotad korištenoga automobila;
početak 3. *godine*: rashodovanje dotad korištenoga automobila i kupnja novoga;
početak 4. *godine*: zadržavanje dotad korištenoga automobila;
početak 5. *godine*: rashodovanje dotad korištenoga automobila i kupnja novoga;
kraj 5. *godine/početak* 6. *godine*: rashodovanje dotad korištenoga automobila.

Obje navedene optimalne politike postižu najmanji iznos ukupnih neto–troškova 31 n.j.

Primjer 2. Usporedno sa studiranjem logistike na BA Nordhessen – studiji u Rijeci Ksenofont je putem Student–servisa dobio honorarni posao dostavljača lokalnih besplatnih novina, oglasnoga materijala itd. Za taj mu je posao potreban bicikl kojega trenutno ne posjeduje. Trenutna cijena novoga bicikla je p = 500 n.j. Novi bicikl se može koristiti najviše 3 godine, a potom ga treba prodati. Godišnji neto–troškovi održavanja bicikla – u ovisnosti o njegovoj starosti – navedeni su u sljedećoj tablici:

starost bicikla (i) [god.]	0	1	2
godišnji neto – trošak (c_i) [n.j.]	30	40	60

Likvidacijske vrijednosti bicikla – u ovisnosti o njegovoj starosti – navedeni su u sljedećoj tablici:

starost bicikla (i) [god.]	1	2	3
likvidacijska vrijednost (s_i) [n.j.]	400	300	250

Ako Ksenofont namjerava završiti svoj studij za točno 5 godina i dotad honorarno raditi kao dostavljač, odredimo optimalnu politiku zamjene bicikla tijekom razdoblja od 5 godina. (Pretpostavljamo da je cijena novoga bicikla stalna u cjelokupnom razdoblju planiranja.)

Prije negoli pristupimo rješavanju zadatka, napravit ćemo jednostavnu analizu kojom ćemo pokušati utvrditi koje "trajanje vlasništva" je najekonomičnije, tj. je li općenito bolje zadržati bicikl 1, 2 ili 3 godine? Kupi li Ksenofont bicikl, koristi ga jednu godinu i potom proda, prosječan (zapravo, ukupan) godišnji neto–trošak iznosi $p + c_0 - s_1$ = 500 + 30 – 400 = 130

n.j. Kupi li Ksenofont bicikl, koristi ga ukupno dvije godine i potom proda, prosječan godišnji neto–trošak iznosit će $\frac{p+(c_0+c_1)-s_2}{2}=\frac{500+(30+40)-300}{2}$ = 135 n.j., a ukoliko kupi bicikl, koristi ga sve tri godine i potom proda, prosječan godišnji neto–trošak iznosit će $\frac{p+(c_0+c_1+c_2)-s_3}{3}=\frac{500+(30+40+60)-250}{3}$ $\approx$ 126.67 n.j. Ova analiza pokazuje da je najisplativije zadržati bicikl sve tri godine i potom ga prodati, a ako to nije moguće, onda je najisplativije zadržati bicikl točno jednu godinu. Budući da je razdoblje planiranja n = 5 godina, možemo očekivati da će optimalna politika zamjene biti neka kombinacija jednogodišnjega i trogodišnjega trajanja vlasništva.

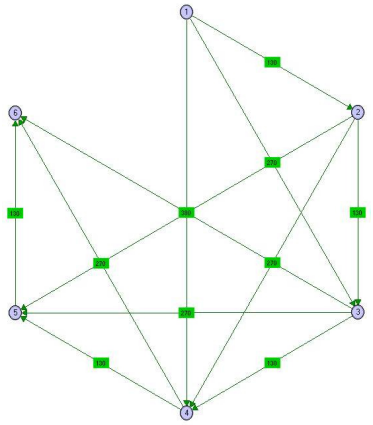
I u ovom ćemo slučaju problem modelirati usmjerenim grafom, ali taj graf neće biti potpun. Naime, bicikl se može koristiti najviše tri godine, a to je razdoblje strogo manje od razdoblja planiranja. Stoga neki vrhovi usmjerenoga grafa neće biti spojeni niti jednom spojnicom.

I u ovom ćemo slučaju pretpostaviti da nam vrhovi grafa označavaju početke godina. Stoga imamo ukupno n + 1 = 6 vrhova, tj. vrh i označavat će nam početak godine i (pritom ponovno pretpostavljamo da se kraj pete godine podudara s početkom šeste godine). Pogledajmo sada koji vrhovi mogu biti spojeni lukom.

Budući da se bicikl može koristiti najviše 3 godine, vrh i ćemo spojiti s vrhom j ako i samo ako vrijedi nejednakost $0 < j - i \leq 3$. Stoga vrh 1 lukovima spajamo s vrhovima 2, 3 i 4, vrh 2 lukovima spajamo s vrhovima 3, 4 i 5, vrh 3 s vrhovima 4, 5 i 6, vrh 4 s vrhovima 5 i 6, a vrh 5 samo s vrhom 6. Izračunajmo težine pojedinih lukova:

$w_{12} = w_{23} = w_{34} = w_{45} = w_{56}$ = (nabavna cijena bicikla) + (troškovi održavanja novoga bicikla) – (likvidacijski troškovi bicikla staroga 1 godinu) = $p + c_0 - s_1$ = 500 + 30 – 400 = 130 n.j.;
$w_{13} = w_{24} = w_{35} = w_{46}$ = (nabavna cijena bicikla) + (ukupni troškovi održavanja novoga bicikla tijekom dvije godine) – (likvidacijski troškovi bicikla staroga 2 godine) = $p + (c_0 + c_1) - s_2$ = 500 + (30 + 40) – 300 = 270 n.j.;
$w_{14} = w_{25} = w_{36}$ = (nabavna cijena bicikla) + (troškovi održavanja novoga bicikla tijekom tri godine) – (likvidacijski troškovi bicikla staroga 3 godine) = $p + (c_0 + c_1 + c_2) - s_3$ = 500 + (30 + 40 + 60) – 250 = 380 n.j.

Dakle, dobiveni usmjereni graf izgleda ovako:



Slika 16. Model problema iz Primjera 2.

Analogno kao i u prethodnom primjeru, označimo vrh 1 kao početni vrh, a vrh 6 kao krajnji vrh i nađemo najkraći put između vrhova 1 i 6:

$$1 - 2 - 3 - 6$$

čija je težina 640. Prema tome, *optimalna politika* zamjene bicikla je sljedeća:

početak 2. godine: prodaja dotad korištenoga bicikla i kupnja novoga bicikla;
početak 3. godine: prodaja dotad korištenoga bicikla i kupnja novoga bicikla;
početak 4. godine: zadržavanje dotad korištenoga bicikla;
početak 5. godine: zadržavanje dotad korištenoga bicikla;
kraj 5. godine/početak 6. godine: prodaja dotad korištenoga bicikla.

Kako vidimo, dobivena optimalna politika je uistinu kombinacija trajanja vlasništva od jedne, odnosno tri godine, tj. navedenoj optimalnoj politici odgovara rastav 5 = 1 + 1 + 3. Zanimljivo je da je ukupan broj različitih optimalnih politika jednak ukupnom broju različitih načina na koji se broj 5 može napisati kao zbroj barem jedne "jedinice" i barem jedne "trojke", a taj je jednak 3. Ti načini su: 1 + 1 + 3, 1 + 3 + 1, 3 + 1 + 1 i svaki od njih generira jednu optimalnu politiku (prvi način generira gore navedenu optimalnu politiku, drugi način generira optimalnu politiku kupnje novoga bicikla u 2. i 5. godini, a treći optimalnu politiku kupnje novoga bicikla u 4. i 5. godini). Sve tri optimalne politike daju iste minimalne ukupne neto–troškove u iznosu od 640 n.j.

Primjer 3. Riješimo prethodni primjer uz sljedeće početne podatke:

- nabavna cijena bicikla na početku i – te godine dana je formulom $p_i = 480 + 20 \cdot i$, za $i = 1, 2, 3, 4, 5$;
- godišnji neto–trošak održavanja bicikla staroga i godina dan je formulom $c_i = 30 + 10 \cdot i^2$, za $i = 0, 1, 2$;
- likvidacijska vrijednost bicikla staroga i godina dana je formulom $s_i = 400 - 20 \cdot i^2$, za $i = 1, 2, 3$.

Iz navedenih podataka slijede donje tablice:

početak godine i	1	2	3	4	5
nabavna cijena bicikla (p_i) [n.j.]	500	520	540	560	580

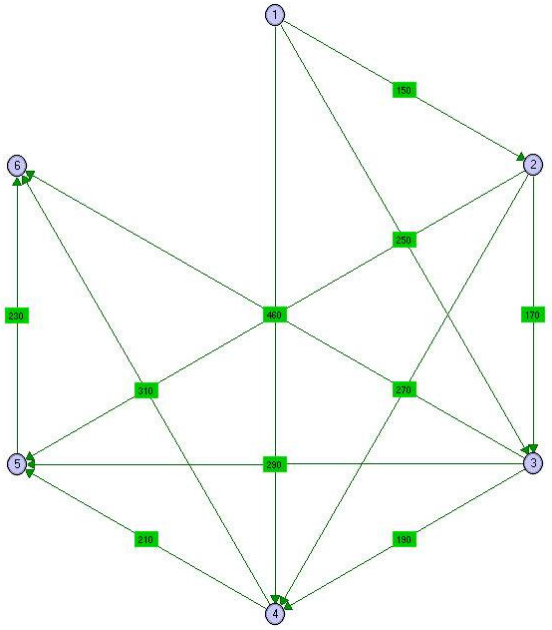
starost bicikla (i) [god.]	0	1	2	3
godišnji trošak (c_i) [n.j.]	30	40	70	–
likvidacijska vrijednost (s_i) [n.j.]	–	380	320	220

Postupak računanja težina odgovarajućih lukova može se pojednostavniti na sljedeći način:

za $j = i + 1$: w_{ij} = (nabavna cijena bicikla na početku godine i) + (troškovi održavanja novoga bicikla) – (likvidacijska vrijednost bicikla staroga 1 godinu) = $p_i + c_0 - s_1 = (480 + 20 \cdot i) + 30 - 380 = 130 + 20 \cdot i$, pa je $w_{12} = 130 + 20 = 150$, $w_{23} = 130 + 20 \cdot 2 = 170$, $w_{34} = 130 + 20 \cdot 3 = 190$, $w_{45} = 130 + 20 \cdot 4 = 210$, $w_{56} = 130 + 20 \cdot 5 = 230$,

<u>za $j = i + 2$</u>: w_{ij} = (nabavna cijena bicikla na početku godine i) + (ukupni troškovi održavanja novoga bicikla tijekom dvije godine) – (likvidacijska vrijednost bicikla staroga 2 godine) = $p_i + (c_0 + c_1) - s_2 = (480 + 20 \cdot i) + (30 + 40) - 320 = 230 + 20 \cdot i$, pa je $w_{13} = 230 + 20 = 250$, $w_{24} = 230 + 20 \cdot 2 = 270$, $w_{35} = 230 + 20 \cdot 3 = 290$, $w_{46} = 230 + 20 \cdot 4 = 310$;
<u>za $j = i + 3$</u>: w_{ij} = (nabavna cijena bicikla na početku godine i) + (ukupni troškovi održavanja novoga bicikla tijekom tri godine) – (likvidacijska vrijednost bicikla staroga 3 godine) = $p_i + (c_0 + c_1 + c_2) - s_3 = (480 + 20 \cdot i) + (30 + 40 + 70) - 220 = 400 + 20 \cdot i$, pa je $w_{14} = 400 + 20 = 420$, $w_{25} = 400 + 20 \cdot 2 = 440$, $w_{36} = 400 + 20 \cdot 3 = 460$.

Pripadni matematički model je sljedeći usmjereni graf:



Slika 17. Model problema iz Primjera 3.

Najkraći put između vrha 1 i vrha 6 u tom grafu je:

1 – 3 – 6

i ima težinu 710. Prema tome, *optimalna politika* zamjene bicikla je sljedeća:

<u>*početak* 2. *godine*</u>: zadržavanje dotad korištenoga bicikla;
<u>*početak* 3. *godine*</u>: prodaja dotad korištenoga bicikla i kupnja novoga bicikla;
<u>*početak* 4. *godine*</u>: zadržavanje dotad korištenoga bicikla;
<u>*početak* 5. *godine*</u>: zadržavanje dotad korištenoga bicikla;
<u>*kraj* 5. *godine*/*početak* 6. *godine*</u>: prodaja dotad korištenoga bicikla.

Minimalni ukupni neto–troškovi u ovom slučaju iznose 710 n.j.

U prethodna tri primjera razmatrali smo slučajeve u kojima je na početku prve godine već bila donijeta odluka o kupnji nove opreme (automobila, bicikla), pa su se sve ostale odluke donosile na početku svake od sljedećih n godina. (Tu uključujemo i završnu odluku o

rashodovanju opreme.) Međutim, u praksi se nerijetko javljaju slučajevi u kojima na početku prve godine već raspolažemo s opremom starom i godina čiji je vijek trajanja ukupno n godina, pa treba razmotriti optimalnu politiku zamjene u sljedećih $n - i$ godina. I u takvim se slučajevima problem zamjene opreme može svesti na problem najkraćega puta u usmjerenom grafu, no, u takvom se grafu pojavljuju negativne težine lukova i višestruki lukovi između dvaju vrhova, pa navedeni model ne možemo implementirati u programu *Graph Magics*. Stoga ćemo takav problem riješiti "klasično" ilustrirajući tehniku dinamičkoga programiranja koja se ovdje primjenjuje.

Primjer 4. Pretpostavimo da na početku prve godine raspolažemo s vozilom starim dvije godine. Vozilo je potrebno koristiti u sljedeće $n = 3$ godine, pri čemu na početku svake godine (uračunavajući i prvu!) donosimo odluku: zadržati postojeće vozilo još godinu dana ili rashodovati postojeće vozilo i kupiti novo. Kao i ranije, godišnji trošak održavanja vozila zadan je tablično kao funkcija starosti automobila:

starost automobila (i) [god.]	0	1	2	3	4
godišnji neto–trošak (c_i) [n.j.]	10	20	40	60	70

Nadalje, cijena novoga vozila u sve tri godine je stalna i iznosi $p = 60$ n.j. Likvidacijska neto–vrijednost automobila staroga i godina zadana je funkcijom:

$$s_i = s(i) = 25 - 5 \cdot i, \text{ za } i = 1, 2, 3, 4, 5.$$

Treba odrediti optimalnu politiku zamjene automobila u cjelokupnom razdoblju planiranja. Primijetimo da je ukupan broj različitih politika zamjene jednak je $2^n = 2^3 = 8$ jer početkom svake godine donosimo jednu od dvije moguće odluke.

Da bismo mogli riješiti navedeni problem, najprije definiramo *funkciju optimalne vrijednosti* $V = V(k, i)$ s:

$V(k, i)$ = najmanji ukupni neto–troškovi od početka godine k do kraja razdoblja planiranja (tj. početka godine $n + 1$) ako na početku godine k raspolažemo s vozilom starosti i.

Funkcija V je funkcija dviju *cjelobrojnih* (točnije, prirodnih) varijabli: k i i. Varijabla k naziva se *varijabla etape* i ona "mjeri" razdoblja (etape) u kojima se donosi točno jedna odluka. U ovom slučaju varijabla etape poprimat će vrijednosti iz skupa {1, 2, 3, 4} jer ćemo početkom 1., 2., 3. i 4. godine donositi odluku o zadržavanju ili rashodovanju vozila. Odmah uočimo da će odluka donijeta početkom $(n + 1)$ – ve, tj. 4. godine biti: *rashodovati vozilo* jer će završiti planirano razdoblje u kojemu želimo koristiti vozilo. Varijabla i naziva se *varijabla stanja* i ona "mjeri" starost vozila na početku godine k. Ta će varijabla poprimati vrijednosti iz skupa {1, 2, 3, 4, 5} jer najveća moguća starost vozila iznosi $i_{max} = 2 + 3 = 5$ godina[25].

Iz postavke navedenoga primjera slijedi da trebamo izračunati vrijednost $V(1, 2)$ jer na početku $k = 1$. godine raspolažemo s vozilom starosti $i = 2$ godine. Navedenu ćemo vrijednost izračunati tako što ćemo najprije definirati dvoindeksnu rekurzivnu relaciju koju

[25] Godišnji trošak održavanja automobila staroga točno 5 godina jednak je nuli. Naime, ukoliko je početkom neke godine starost automobila točno 5 godina, obavezno odmah rashodujemo automobil kako bismo izbjegli daljnje troškove njegova održavanja.

zadovoljavaju vrijednosti $V(k, i)$, a potom početne uvjete koji će nam omogućiti računanje konkretnih vrijednosti te funkcije.
Spomenutu dvoindeksnu rekurzivnu relaciju dobivamo na sljedeći način: Zamislimo da na početku godine k koristimo vozilo staro i godina. Na raspolaganju su nam alternative A_1 = = "*zadržati vozilo još godinu dana*" i A_2 = "*rashodovati vozilo i zamijeniti ga novim vozilom*". Razmotrimo zasebno što se događa donošenjem svake pojedine odluke.

Opredijelimo li se za alternativu A_1, tj. odlučimo li zadržati vozilo i koristiti ga u cijeloj godini k (do početka godine $k + 1$), ukupni neto–trošak u godini k koji proizlazi iz takve odluke tvori isključivo neto–trošak održavanja automobila staroga i godina. Taj je trošak, prema oznakama iz primjera, jednak c_i. Sljedeća etapa u kojoj se donosi odluka je početak $(k + 1)$ – ve godine, a u tom trenutku imamo na raspolaganju vozilo starosti $i + 1$ godina. Najmanji ukupni neto–troškovi od početka $(k + 1)$ – ve godine do kraja razdoblja planiranja, prema definiciji funkcije V, iznose $V(k + 1, i + 1)$. Prema tome, ukupni neto–troškovi *nastali kao posljedica izbora alternative* A_1 *na početku godine* k iznose $c_i + V(k + 1, i + 1)$.

Opredijelimo li se za alternativu A_2, tj. odlučimo li rashodovati dotad korišteno vozilo i kupiti novo, u godini k imat ćemo trošak nabave novoga vozila, godišnji trošak njegova održavanja, te prihod dobiven rashodovanjem i prodajom dotadašnjega vozila, tj. prihod jednak likvidacijskoj vrijednosti dotadašnjega vozila. Trošak nabave novoga vozila jednak je nabavnoj cijeni vozila, tj. p. Godišnji neto–trošak održavanja novoga vozila iznosi c_0, a likvidacijska vrijednost dotadašnjega vozila (staroga i godina) iznosi s_i. Stoga ukupni neto–troškovi u godini k iznose $p + c_0 - s_i$. Sljedeća etapa u kojoj se donosi odluka je početak $(k + 1)$ – ve godine, a u tom trenutku na raspolaganju imamo vozilo starosti 1 godinu. Najmanji ukupni neto–troškovi od početka $(k + 1)$ – ve godine do kraja razdoblja planiranja, prema definiciji funkcije V, iznose $V(k + 1, 1)$. Prema tome, ukupni neto–troškovi *nastali kao posljedica izbora alternative* A_2 *na početku godine* k iznose $p + c_0 - s_i + V(k + 1, 1)$.

Budući da ukupne neto–troškove treba minimizirati u svakoj pojedinoj etapi, slijedi da mora vrijediti sljedeća jednakost:

$$V(k, i) = \min \{c_i + V(k + 1, i + 1), p + c_0 - s_i + V(k + 1, 1)\}.$$

Navedena jednakost je tražena dvoindeksna rekurzivna relacija pomoću koje ćemo računati vrijednosti funkcije V. Preostaje nam zadati *početne uvjete*, tj. neke vrijednosti funkcije V pomoću kojih se može izračunati *svaka* od preostalih vrijednosti te funkcije.

Da bismo zadali početne uvjete, razmatramo odluku koju donosimo na kraju razdoblja planiranja, odnosno početkom četvrte godine. Pripadna vrijednost varijable etape je $k = 4$, pa zapravo zadajemo vrijednosti $V(4, i)$. Na kraju razdoblja planiranja moguća je točno jedna odluka: "*rashodovati dotad korišteno vozilo*". To vozilo ne može biti novo, tj. ne može biti $i_{min} = 0$, nego je najmanja moguća starost vozila jednaka $i = 1$. Najveća moguća starost vozila jest, kako smo ranije utvrdili, $i = 5$, pa računamo vrijednosti $V(4, i)$ za $i = 1, 2, 3, 4, 5$. Rezultat odluke "*rashodovati dotad korišteno vozilo*" je *prihod* jednak likvidacijskoj vrijednosti vozila na početku četvrte godine. Budući da minimiziramo ukupne troškove, prihod ćemo formalno evidentirati kao negativne troškove:

$V(4, 1)$ = –(likvidacijska vrijednost vozila staroga 1 godinu) = $- s_1 = -(25 - 5 \cdot 1) = -20$;

$V(4, 2)$ = –(likvidacijska vrijednost vozila staroga 2 godine) = $- s_2 = -(25 - 5 \cdot 2) = -15$;

$V(4, 3) = -$(likvidacijska vrijednost vozila staroga 3 godine) $= -s_3 = -(25 - 5 \cdot 3) = -10$;

$V(4, 4) = -$(likvidacijska vrijednost vozila staroga 4 godine) $= -s_4 = -(25 - 5 \cdot 4) = -5$;

$V(4, 5) = -$(likvidacijska vrijednost vozila staroga 5 godina) $= -s_5 = -(25 - 5 \cdot 5) = 0$.

Zadavanjem navedenih vrijednosti možemo izračunati svaku od preostalih vrijednosti funkcije V na temelju prethodno navedene rekurzivne relacije. Najprije razmatramo početak 3. godine, odnosno etapu $k = 3$. Najveća moguća starost vozila u toj etapi je $i = 4$, pa računamo vrijednosti $V(3, i)$, za $i = 1, 2, 3, 4$. Imamo redom:

$V(3, 1) = \min\{c_1 + V(3 + 1, 1 + 1), p + c_0 - s_1 + V(3 + 1, 1)\} = \min\{20 + V(4, 2), 60 + 10 - 20 + V(4, 1)\} = \min\{20 + (-15), 50 + (-20)\} = 5$;
$V(3, 2) = \min\{c_2 + V(3 + 1, 2 + 1), p + c_0 - s_2 + V(3 + 1, 1)\} = \min\{40 + V(4, 3), 60 + 10 - 15 + V(4, 1)\} = \min\{40 + (-10), 55 + (-20)\} = 30$;
$V(3, 3) = \min\{c_3 + V(3 + 1, 3 + 1), p + c_0 - s_3 + V(3 + 1, 1)\} = \min\{60 + V(4, 4), 60 + 10 - 10 + V(4, 1)\} = \min\{60 + (-5), 60 + (-20)\} = 40$;
$V(3, 4) = \min\{c_4 + V(3 + 1, 4 + 1), p + c_0 - s_4 + V(3 + 1, 1)\} = \min\{70 + V(4, 5), 60 + 10 - 5 + V(4, 1)\} = \min\{70 + 0, 65 + (-20)\} = 45$.

U sljedećem koraku razmatramo početak druge godine, odnosno etapu $k = 2$. Najveća moguća starost vozila u toj etapi je $i = 3$, pa računamo vrijednosti $V(2, i)$ za $i = 1, 2, 3$. Imamo redom:

$V(2, 1) = \min\{c_1 + V(2 + 1, 1 + 1), p + c_0 - s_1 + V(2 + 1, 1)\} = \min\{20 + V(3, 2), 60 + 10 - 20 + V(3, 1)\} = \min\{20 + 30, 50 + 5\} = 50$;
$V(2, 2) = \min\{c_2 + V(2 + 1, 2 + 1), p + c_0 - s_2 + V(2 + 1, 1)\} = \min\{40 + V(3, 3), 60 + 10 - 15 + V(3, 1)\} = \min\{40 + 40, 55 + 5\} = 60$;
$V(2, 3) = \min\{c_3 + V(2 + 1, 3 + 1), p + c_0 - s_3 + V(2 + 1, 1)\} = \min\{60 + V(3, 4), 60 + 10 - 10 + V(3, 1)\} = \min\{60 + 45, 60 + 5\} = 65$.

Napokon, prelazimo na početak prve godine, odnosno etapu $k = 1$. Najveća moguća starost vozila u toj etapi je $i = 2$, pa radi potpunosti računamo vrijednosti $V(1, i)$, za $i = 1, 2$:

$V(1, 1) = \min\{c_1 + V(1 + 1, 1 + 1), p + c_0 - s_1 + V(1 + 1, 1)\} = \min\{20 + V(2, 2), 60 + 10 - 20 + V(2, 1)\} = \min\{20 + 60, 50 + 50\} = 80$;
$V(1, 2) = \min\{c_2 + V(1 + 1, 2 + 1), p + c_0 - s_2 + V(1 + 1, 1)\} = \min\{40 + V(2, 3), 60 + 10 - 15 + V(2, 1)\} = \min\{40 + 65, 55 + 50\} = 105$.

Ranije smo istaknuli da zapravo želimo izračunati vrijednost $V(1, 2)$. Ta je vrijednost jednaka 105, pa zaključujemo da su *ukupni minimalni troškovi korištenja vozila u razdoblju planiranja* 105 n.j. Primijetimo da se vrijednost $V(1, 2) = 105$ postiže i izborom alternative A_1 i izborom alternative A_2, što znači da imamo ukupno *dvije različite optimalne politike zamjene*. Te politike možemo "očitati" iz gore provedenih izračuna na sljedeći način:

<u>1. *optimalna politika zamjene*:</u>

<u>*početak* 1. *godine*</u>: zadržati vozilo staro 2 godine;
<u>*početak* 2. *godine*</u>: vrijednost $V(1, 2) = 105$ u ovom je slučaju dobivena pomoću vrijednosti $V(2, 3) = 65$. Ta je vrijednost dobivena izborom alternative A_2 na početku 2. godine, što znači da tada *treba rashodovati dotad korišteno vozilo i kupiti novo*;

<u>*početak* 3. *godine*</u>: vrijednost $V(2, 3) = 65$ dobivena je pomoću vrijednosti $V(3, 1) = 5$. Ta je vrijednost dobivena izborom alternative A_1 na početku 3. godine, što znači da tada *treba zadržati dotad korišteno vozilo*;
<u>*početak* 4. *godine*</u>: rashodovati dotad korišteno vozilo.

<u>2. *optimalna politika zamjene*</u>:

<u>*početak* 1. *godine*</u>: rashodovati vozilo staro 2 godine i kupiti novo vozilo;
<u>*početak* 2. *godine*</u>: vrijednost $V(1, 2) = 105$ u ovom je slučaju dobivena pomoću vrijednosti $V(2, 1) = 50$. Ta vrijednost dobivena je izborom alternative A_1 na početku 2. godine, što znači da tada *treba zadržati dotad korišteno vozilo*;
<u>*početak* 3. *godine*</u>: vrijednost $V(2, 1) = 50$ dobivena je pomoću vrijednosti $V(3, 2) = 30$. Ta je vrijednost dobivena izborom alternative A_1 na početku 3. godine, što znači da tada *treba zadržati dotad korišteno vozilo*;
<u>*početak* 4. *godine*</u>: rashodovati dotad korišteno vozilo.

U prethodnim smo primjerima nastojali minimizirati ukupne troškove nastale korištenjem opreme. No, u praksi se vrlo često javljaju i problemi maksimizacije ukupne dobiti nastale korištenjem neke opreme. Postupak rješavanja takvih problema vrlo je sličan onome izloženom u Primjeru 4., pa ćemo ga ilustrirati na sljedećim primjerima.

<u>Primjer 5.</u> Za opremu nekoga proizvodnoga pogona koja osigurava nesmetan i neprekidan proces proizvodnje zadani su nabavna cijena $p = 120$ n.j. i funkcija neto–dobiti nastale korištenjem opreme (d). Nakon određenoga vremena korištenja opremu *ne* možemo prodati, pa početkom svake godine treba donijeti odluku o zamjeni ili zadržavanju dotadašnje opreme. Godišnja neto–dobit od korištenja opreme zadana je kao funkcija starosti opreme (i):

$$d_i = d(i) = \begin{cases} 120 - 20 \cdot i, \text{ za } i \in [6] \\ 0, \text{ za } i \geq 7 \end{cases}$$

Odredimo optimalnu politiku zamjene opreme u razdoblju od $n = 8$ godina ako na početku 1. godine raspolažemo s opremom starom 3 godine.

Kao i u Primjeru 4., uvodimo dvije varijable: *varijabla etape* (k) koja će "mjeriti" razdoblja u kojima se donosi točno jedna odluka i *varijabla stanja* (i) koja će "mjeriti" starost opreme. Definiramo realnu funkciju V dvije cjelobrojne varijable s:

$V = V(k, i)$ = najveća ukupna neto–dobit ostvarena od početka godine k do kraja razdoblja planiranja ako na početku godine k na raspolaganju imamo opremu staru i godina.

Budući da na početku 1. godine raspolažemo s opremom starosti 3 godine, cilj nam je odrediti vrijednost $V(1, 3)$. Da bismo to mogli učiniti, najprije moramo zapisati odgovarajuću rekurzivnu relaciju i zadati početne uvjete.

Za određivanje rekurzivne relacije provodimo sljedeće zaključivanje: Zamislimo da na početku godine k koristimo opremu staru i godina. Na raspolaganju su nam alternative A_1 = = "*zadržati opremu još godinu dana*" i A_2 = "*zamijeniti opremu*". Razmotrimo zasebno što se događa donošenjem svake pojedine odluke.

Opredijelimo li se za alternativu A_1, tj. odlučimo li zadržati opremu i koristiti ga u cijeloj godini k (do početka godine $k + 1$), ukupna neto–dobit u godini k koja proizlazi iz takve odluke jednaka je neto–dobiti nastaloj korištenje opreme stare i godina. Ta je dobit, prema oznakama iz primjera, jednaka d_i. Sljedeća etapa u kojoj se donosi odluka je početak $(k + 1)$ – ve godine, a u tom trenutku imamo na raspolaganju opremu starosti $i + 1$ godina. Najveća ukupna neto–dobit od početka $(k + 1)$ – ve godine do kraja razdoblja planiranja, prema definiciji funkcije V, iznosi $V(k + 1, i + 1)$. Prema tome, ukupna neto–dobit *nastala kao posljedica izbora alternative* A_1 *na početku godine* k iznose $d_i + V(k + 1, i + 1)$.

Opredijelimo li se za alternativu A_2, tj. odlučimo li zamijeniti opremu, u godini k imat ćemo trošak nabave nove opreme i neto–dobit nastalu korištenjem nove opreme. Trošak nabave nove opreme jednak je nabavnoj cijeni opreme, tj. p, a godišnja neto–dobit nastala korištenjem nove opreme iznosi d_0. Stoga ukupna neto–dobit u godini k iznosi $d_0 - p$. Sljedeća etapa u kojoj se donosi odluka je početak $(k + 1)$ – ve godine, a u tom trenutku na raspolaganju imamo opremu starosti 1 godinu. Najveća ukupna neto–dobit od početka $(k + 1)$ – ve godine do kraja razdoblja planiranja, prema definiciji funkcije V, iznosi $V(k + 1, 1)$. Prema tome, ukupna neto–dobit *nastala kao posljedica izbora alternative* A_2 *na početku godine* k iznosi $d_0 - p + V(k + 1, 1)$.

Budući da ukupnu neto–dobit treba minimizirati u svakoj pojedinoj etapi, slijedi da mora vrijediti sljedeća jednakost:

$$V(k, i) = \max \{d_i + V(k + 1, i + 1), d_0 - p + V(k + 1, 1)\} = \max \{d_i + V(k + 1, i + 1), 120 - 120 + V(k + 1, 1)\} = \max \{d_i + V(k + 1, i + 1), V(k + 1, 1)\}$$

Navedena jednakost je tražena dvoindeksna rekurzivna relacija pomoću koje ćemo računati vrijednosti funkcije V. Preostaje nam zadati *početne uvjete*, tj. neke vrijednosti funkcije V pomoću kojih se može izračunati *svaka* od preostalih vrijednosti te funkcije.

Da bismo zadali početne uvjete, razmatramo odluku koju donosimo početkom posljednje godine razdoblja planiranja, odnosno početkom osme godine. Pripadna vrijednost varijable etape je $k = 8$, pa zapravo zadajemo vrijednosti $V(8, i)$. Najmanja moguća vrijednost varijable stanja je $i_{min} = 0$ i postiže se ako početkom 8. godine odlučimo kupiti novu opremu. Najveća moguća vrijednost varijable stanja jednaka je $i_{max} = 3 + 7 = 10$ i postiže se ako opremu staru 3 godine zadržimo tijekom svih 8 godina. Dakle, zadajemo vrijednosti $V(8, i)$ za $i \in [10]$. Te vrijednosti računamo pomoću izraza:

$$V(8, i) = d_i$$

jer u posljednjoj godini koristimo opremu staru 0 (tj. novu opremu), 1, 2, 3, 4, 5, 6, 7, 8, 9, ili 10 godina. Tako dobivamo sljedeću tablicu:

i	0	1	2	3	4	5	6	7	8	9	10
$V(8, i)$	120	100	80	60	40	20	0	0	0	0	0

U sljedećem koraku računamo vrijednosti $V(7, i)$. U etapi $k = 7$, tj. početkom 7. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 6 = 9$, pa računamo vrijednosti $V(7, i)$ za $i \in [9]$. Koristeći navedenu dvoindeksnu rekurzivnu relaciju dobivamo redom:

$V(7, 0) = \max\{d_0 + V(7 + 1, 0 + 1), V(7 + 1, 1)\} = \max\{120 + V(8, 1), V(8, 1)\} = \max\{120 + 100, 100\} = 220$;
$V(7, 1) = \max\{d_1 + V(7 + 1, 1 + 1), V(7 + 1, 1)\} = \max\{100 + V(8, 2), V(8, 1)\} = \max\{100 + 80, 100\} = 180$;
$V(7, 2) = \max\{d_2 + V(7 + 1, 2 + 1), V(7 + 1, 1)\} = \max\{80 + V(8, 3), V(8, 1)\} = \max\{80 + 60, 100\} = 140$;
$V(7, 3) = \max\{d_3 + V(7 + 1, 3 + 1), V(7 + 1, 1)\} = \max\{60 + V(8, 4), V(8, 1)\} = \max\{60 + 40, 100\} = 100$;
$V(7, 4) = V(7, 5) = V(7, 6) = V(7, 7) = V(7, 8) = V(7, 9) = V(8, 1) = 100$ (što znači da ako početkom sedme godine imamo na raspolaganju opremu staru barem 3 godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(6, i)$. U etapi $k = 6$, tj. početkom 6. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 5 = 8$. Stoga računamo vrijednosti $V(6, i)$ za $i \in [8]$. Dobivamo:

$V(6, 0) = \max\{d_0 + V(6 + 1, 0 + 1), V(6 + 1, 1)\} = \max\{120 + V(7, 1), V(7, 1)\} = \max\{120 + 180, 180\} = 300$;
$V(6, 1) = \max\{d_1 + V(6 + 1, 1 + 1), V(6 + 1, 1)\} = \max\{100 + V(7, 2), V(7, 1)\} = \max\{100 + 140, 180\} = 240$;
$V(6, 2) = \max\{d_2 + V(6 + 1, 2 + 1), V(6 + 1, 1)\} = \max\{80 + V(7, 3), V(7, 1)\} = \max\{80 + 100, 180\} = 180$;
$V(6, 3) = V(6, 4) = V(6, 5) = V(6, 6) = V(6, 7) = V(6, 8) = 180$ (što znači da ako početkom šeste godine imamo na raspolaganju opremu staru barem dvije godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(5, i)$. U etapi $k = 5$, tj. početkom 5. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 4 = 7$. Stoga računamo vrijednosti $V(5, i)$ za $i \in [7]$. Dobivamo:

$V(5, 0) = \max\{d_0 + V(5 + 1, 0 + 1), V(5 + 1, 1)\} = \max\{120 + V(6, 1), V(6, 1)\} = \max\{120 + 240, 240\} = 360$;
$V(5, 1) = \max\{d_1 + V(5 + 1, 1 + 1), V(5 + 1, 1)\} = \max\{100 + V(6, 2), V(6, 1)\} = \max\{100 + 180, 240\} = 280$;
$V(5, 2) = \max\{d_2 + V(5 + 1, 2 + 1), V(5 + 1, 1)\} = \max\{80 + V(6, 3), V(6, 1)\} = \max\{80 + 180, 240\} = 260$;
$V(5, 3) = \max\{d_3 + V(5 + 1, 3 + 1), V(5 + 1, 1)\} = \max\{60 + V(6, 4), V(6, 1)\} = \max\{60 + 180, 240\} = 240$;
$V(5, 4) = V(5, 5) = V(5, 6) = V(5, 7) = 240$ (što znači da ako početkom pete godine imamo na raspolaganju opremu staru barem tri godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(4, i)$. U etapi $k = 4$, tj. početkom 4. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 3 = 6$. Stoga računamo vrijednosti $V(4, i)$ za $i \in [6]$. Dobivamo:

$V(4, 0) = \max\{d_0 + V(4 + 1, 0 + 1), V(4 + 1, 1)\} = \max\{120 + V(5, 1), V(5, 1)\} = \max\{120 + 280, 280\} = 400$;
$V(4, 1) = \max\{d_1 + V(4 + 1, 1 + 1), V(4 + 1, 1)\} = \max\{100 + V(5, 2), V(5, 1)\} = \max\{100 + 260, 280\} = 360$;

$V(4, 2) = \max\{d_2 + V(4 + 1, 2 + 1), V(4 + 1, 1)\} = \max\{80 + V(5, 3), V(5, 1)\} = \max\{80 + 240, 280\} = 320$;
$V(4, 3) = \max\{d_3 + V(4 + 1, 3 + 1), V(4 + 1, 1)\} = \max\{60 + V(5, 4), V(5, 1)\} = \max\{60 + 240, 280\} = 300$;
$V(4, 4) = \max\{d_4 + V(4 + 1, 4 + 1), V(4 + 1, 1)\} = \max\{40 + V(5, 5), V(5, 1)\} = \max\{40 + 240, 280\} = 280$;
$V(4, 5) = V(4, 6) = 280$ (što znači da ako početkom četvrte godine imamo na raspolaganju opremu staru barem četiri godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(3, i)$. U etapi $k = 3$, tj. početkom 3. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 2 = 5$. Stoga računamo vrijednosti $V(3, i)$ za $i \in [5]$. Dobivamo:

$V(3, 0) = \max\{d_0 + V(3 + 1, 0 + 1), V(3 + 1, 1)\} = \max\{120 + V(4, 1), V(4, 1)\} = \max\{120 + 360, 360\} = 480$;
$V(3, 1) = \max\{d_1 + V(3 + 1, 1 + 1), V(3 + 1, 1)\} = \max\{100 + V(4, 2), V(4, 1)\} = \max\{100 + 320, 360\} = 420$;
$V(3, 2) = \max\{d_2 + V(3 + 1, 2 + 1), V(3 + 1, 1)\} = \max\{80 + V(4, 3), V(4, 1)\} = \max\{80 + 300, 360\} = 380$;
$V(3, 3) = \max\{d_3 + V(3 + 1, 3 + 1), V(3 + 1, 1)\} = \max\{60 + V(4, 4), V(4, 1)\} = \max\{60 + 280, 360\} = 360$;
$V(3, 4) = V(3, 5) = 360$ (što znači da ako početkom treće godine imamo na raspolaganju opremu staru barem tri godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(2, i)$. U etapi $k = 2$, tj. početkom 2. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3 + 1 = 4$. Stoga računamo vrijednosti $V(2, i)$ za $i \in [5]$. Dobivamo:

$V(2, 0) = \max\{d_0 + V(2 + 1, 0 + 1), V(2 + 1, 1)\} = \max\{120 + V(3, 1), V(3, 1)\} = \max\{120 + 420, 420\} = 540$;
$V(2, 1) = \max\{d_1 + V(2 + 1, 1 + 1), V(2 + 1, 1)\} = \max\{100 + V(3, 2), V(3, 1)\} = \max\{100 + 380, 420\} = 480$;
$V(2, 2) = \max\{d_2 + V(2 + 1, 2 + 1), V(2 + 1, 1)\} = \max\{80 + V(3, 3), V(3, 1)\} = \max\{80 + + 360, 420\} = 440$;
$V(2, 3) = \max\{d_3 + V(2 + 1, 3 + 1), V(2 + 1, 1)\} = \max\{60 + V(3, 4), V(3, 1)\} = \max\{60 + + 360, 420\} = 420$;
$V(2, 4) = \max\{d_4 + V(2 + 1, 4 + 1), V(2 + 1, 1)\} = \max\{40 + V(3, 5), V(3, 1)\} = \max\{40 + + 360, 420\} = 420$ (što znači da ako početkom druge godine imamo na raspolaganju opremu staru barem tri godine, svakako se isplati kupiti novu opremu).

U sljedećem koraku računamo vrijednosti $V(1, i)$. U etapi $k = 1$, tj. početkom 1. godine najveća moguća vrijednost varijable stanja je $i_{max} = 3$. Stoga računamo vrijednosti $V(1, i)$ za $i \in [3]$. Dobivamo:

$V(1, 0) = \max\{d_0 + V(1 + 1, 0 + 1), V(1 + 1, 1)\} = \max\{120 + V(2, 1), V(2, 1)\} = \max\{120 + 480, 480\} = 600$;
$V(1, 1) = \max\{d_1 + V(1 + 1, 1 + 1), V(1 + 1, 1)\} = \max\{100 + V(2, 2), V(2, 1)\} = \max\{100 + 440, 480\} = 540$;

$V(1, 2) = \max\{d_2 + V(1 + 1, 2 + 1), V(1 + 1, 1)\} = \max\{80 + V(2, 3), V(2, 1)\} = \max\{80 + 420, 480\} = 500$;
$V(1, 3) = \max\{d_3 + V(1 + 1, 3 + 1), V(1 + 1, 1)\} = \max\{60 + V(2, 4), V(2, 1)\} = \max\{60 + 420, 480\} = 480$.

Prema tome, tražena maksimalna ukupna neto–dobit u cjelokupnom razdoblju planiranja iznosi 480 n.j. Preostaje odrediti optimalnu politiku zamjene opreme, odnosno "očitati" slijed odluka. Vrijednost $V(1, 3) = 480$ postiže se neovisno o izboru alternative (A_1 ili A_2). Stoga imamo sljedeće optimalne politike:

1. *optimalna politika*:

početak 1. *godine*: zadržati postojeću opremu;
početak 2. *godine*: zamjena opreme;
početak 3., *početak* 4. *i početak* 5. *godine*: zadržati postojeću opremu;
početak 6. *godine*: zamjena opreme;
početak 7. *i početak* 8. *godine*: zadržati postojeću opremu;

2. *optimalna politika*:

početak 1. *godine*: zadržati postojeću opremu;
početak 2. *godine*: zamjena opreme;
početak 3. *i početak* 4. *godine*: zadržati postojeću opremu;
početak 5. *godine*: zamjena opreme;
početak 6., *početak* 7. *i početak* 8. *godine*: zadržati postojeću opremu;

3. *optimalna politika*:

početak 1. *godine*: zamjena opreme;
početak 2., *početak* 3. *i početak* 4. *godine*: zadržati postojeću opremu;
početak 5. *godine*: zamjena opreme;
početak 6., *početak* 7. *i početak* 8. *godine*: zadržati postojeću opremu;

Ukoliko unaprijed donesemo odluku da na početku 1. godine zamjenjujemo dotad korištenu opreme, postupak donošenja optimalne politike zamjene bitno možemo ubrzati određivanjem *kritičnoga puta u usmjerenom grafu*[26] pomoću računalnoga programa *Grin*. Nužan uvjet za rješavanje takvoga modela pomoću računalnoga programa *Grin* jest da težina svakoga luka oblika (i, j) za $i < j$ bude strogo pozitivan realan broj[27]. Pokažimo to na sljedećem primjeru.

Primjer 6. Riješimo prethodni primjer uz uvjet da početkom 1. godine obavezno nabavljamo novu opremu. Uočimo odmah da kao rješenje ovoga primjera moramo dobiti 3. optimalnu politiku iz prethodnoga primjera.

Potencijalni matematički model ovdje će biti *potpun* usmjeren graf čiji će vrhovi označavati početke godina. Budući da je razdoblje planiranja dugo $n = 8$ godina, potpun usmjeren graf

[26] Primjer primjene toga problema na problem rasporeda aktivnosti obrađen je u kolegiju *Ekonomska matematika 2*.
[27] Ukoliko je težina luka (i, j) strogo negativan realan broj, luk (i, j) treba zamijeniti lukom (j, i) čija je težina strogo pozitivan realan broj. Navedena zamjena može dovesti do nastanka ciklusa u grafu (naročito ako je graf potpun), a u takvim slučajevima traženi kritični put ne postoji.

imat će ukupno $n + 1$, tj. 9 vrhova: 1, 2, 3, 4, 5, 6, 7, 8, 9. (Vrh 9 interpretiramo kao kraj 8. godine, odnosno kraj razdoblja planiranja.) Vrhovi i i j bit će spojeni lukom od i prema j ako i samo ako je $i < j$. Stoga ćemo imati ukupno $\binom{9}{2} = 36$ različitih lukova. Preostaje zadati težinu svakoga luka:

w_{ij} = (ukupna neto–dobit nastala korištenjem opreme u godinama i, $i + 1$, …, $j - 1$, $j - 1$) – (nabavna cijena opreme u godini i) = $\sum_{k=0}^{j-i-1} d_k - p_i$.

Provjerimo možemo li za računalno rješavanje promatranoga problema iskoristiti računalni program *Grin*. Uočimo da je $d_0 = p = 120$ n.j. To znači da je ukupna neto–dobit u prvoj godini korištenja opreme uvijek jednaka 0, pa za svaki $i \in [8]$ vrijede jednakosti:

$$w_{i,\, i+1} = 0,$$

$$w_{i,\, i+k} = \sum_{m=1}^{k-1} d_k \text{ , za svaki } k \in \mathbf{N} \setminus \{1\} \text{ za koji postoji luk } (i,\, i+k).$$

Prema definiciji veličina d_i za svaki $i \in \mathbf{N}$ očito vrijedi nejednakost $d_i \geq 0$, pa je nužan uvjet ispunjen. Nadalje, zbog

$d_1 = 120 - 20 \cdot 1 = 100;$

$d_2 = 120 - 20 \cdot 2 = 80;$

$d_3 = 120 - 20 \cdot 3 = 60;$

$d_4 = 120 - 20 \cdot 4 = 40;$

$d_5 = 120 - 20 \cdot 5 = 20;$

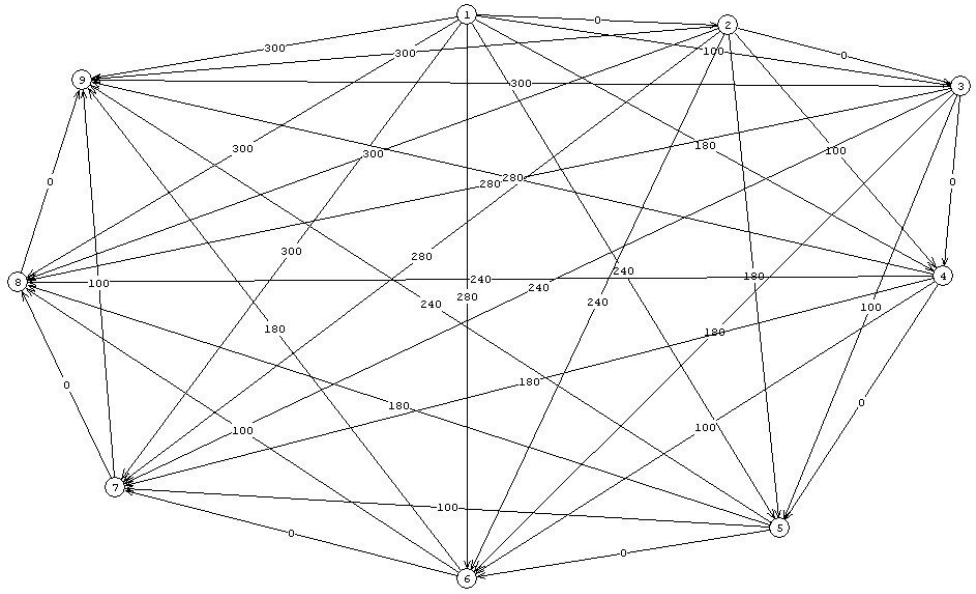
$d_6 = 120 - 20 \cdot 6 = 0;$

$d_7 = d_8 = 0,$

slijedi:

$w_{12} = w_{23} = w_{34} = w_{45} = w_{56} = w_{67} = w_{78} = w_{89} = 0;$
$w_{13} = w_{24} = w_{35} = w_{46} = w_{57} = w_{68} = w_{79} = d_1 = 100;$
$w_{14} = w_{25} = w_{36} = w_{47} = w_{58} = w_{69} = d_1 + d_2 = 100 + 80 = 180;$
$w_{15} = w_{26} = w_{37} = w_{48} = w_{59} = d_1 + d_2 + d_3 = 100 + 80 + 60 = 240;$
$w_{16} = w_{27} = w_{38} = w_{49} = d_1 + d_2 + d_3 + d_4 = 100 + 80 + 60 + 40 = 280;$
$w_{17} = w_{28} = w_{39} = d_1 + d_2 + d_3 + d_4 + d_5 = 100 + 80 + 60 + 40 + 20 = 300;$
$w_{18} = w_{29} = d_1 + d_2 + d_3 + d_4 + d_5 + d_6 = 100 + 80 + 60 + 40 + 20 + 0 = 300;$
$w_{19} = d_1 + d_2 + d_3 + d_4 + d_5 + d_6 + d_7 = 100 + 80 + 60 + 40 + 20 + 0 + 0 = 300.$

Koristeći program *Grin* dobivamo traženi matematički model:



Slika 18. Model problema iz Primjera 6.

Pokretanjem procedure *Critical Path* unutar podizbornika *Network* u sklopu izbornika *Property* dobivamo traženi kritični put:

$$1 - 5 - 9$$

čija je duljina 480. Prema tome, novu opremu treba kupiti početkom 1. godine (to smo već učinili) i početkom 5. godine, a to se upravo podudara s 3. optimalnom politikom dobivenom u rješenju prethodnoga primjera. I tražena najveća ukupna neto–dobit jednaka je kao u prethodnom primjeru i iznosi 480 n.j. (težina kritičnoga puta).

<u>Primjer 7.</u> Poslovna politika neke tvrtke predviđa korištenje određenoga tipa opreme u razdoblju od $n = 5$ godina. U tu je svrhu početkom prve godine nabavljena nova oprema po nabavnoj cijeni od $p_1 = 100$ n.j. Predviđa se da će početkom svake preostale godine razdoblja planiranja nabavna cijena opreme biti za 10% veća nego na početku toj godini neposredno prethodne godine. Neto–dobit nastala korištenjem opreme zadana je kao funkcija starosti opreme i (iskazane u godinama):

$$d_i = 90 - 18 \cdot i, \text{ za } i \in [5].$$

Ako početkom svake preostale godine razdoblja planiranja treba donijeti odluku o zadržavanju ili zamjeni stare opreme, pokažimo da je optimalna politika zamjene opreme zadržati novu opremu do kraja razdoblja planiranja.

Iako se na prvi pogled čini da i ovaj primjer možemo brzo riješiti rabeći računalni program *Grin*, to ipak nije moguće. Naime, niz nabavnih cijena opreme $(p_i)_{i\in\mathbf{N}}$ je strogo rastući, a niz $(d_i)_{i\in\mathbf{N}}$ padajući. Budući da je $p_1 = 100 > 90 = d_0$, zaključujemo da u prvoj godini korištenja nove opreme imamo trošak od $p_1 - d_0 = 10$ n.j., što matematički modeliramo kao negativan prihod –10. Tako u odgovarajućem usmjerenom grafu imamo barem jednu negativnu težinu luka, pa rješavanje pomoću računalnoga programa *Grin* nije moguće. Stoga ćemo problem

riješiti "klasično" formiranjem odgovarajuće rekurzivne relacije i početnih uvjeta, te izračunavanjem konkretnih vrijednosti funkcije optimalne dobiti.

Potpuno analogno kao u ranijim primjerima dobivamo da je funkcija optimalne dobiti

$$V = V(k, i) = \max \{d_i + V(k + 1, i + 1), d_0 - p_k + V(k + 1, 1)\}.$$

Da bismo mogli izračunati konkretne vrijednosti te funkcije, najprije ćemo izračunati sve potrebne vrijednosti p_k za $k = 1, 2, 3, 4, 5$ i d_i za $i = 0, 1, 2, 3, 4$. Dobivamo:

$p_1 = 100$: $\qquad d_0 = 90 - 18 \cdot 0 = 90;$

$p_2 = 100 + \frac{10}{100} \cdot 100 = 110;$ $\qquad d_1 = 90 - 18 \cdot 1 = 72;$

$p_3 = 110 + \frac{10}{100} \cdot 110 = 121;$ $\qquad d_2 = 90 - 18 \cdot 2 = 54;$

$p_4 = 121 + \frac{10}{100} \cdot 121 = 133.1;$ $\qquad d_3 = 90 - 18 \cdot 3 = 36;$

$p_5 = 133.1 + \frac{10}{100} \cdot 133.1 = 146.41;$ $\qquad d_4 = 90 - 18 \cdot 4 = 18.$

Početni su uvjeti vrijednosti funkcije V za $k = 5$ i $i = 0, 1, 2, 3, 4$ jer je najveća moguća starost opreme na početku 5. godine jednaka $i = 4$ godine. Stoga je:

$V(5, i) = \max \{d_i, d_0 - p_5) =$ (zbog $d_0 - p_5 = 90 - 146.41 = -56.41 < 0 < d_i$ za svaki $i = 0, 1, 2, 3, 4$) $= d_i$,

tj.

$V(5, 0) = d_0 = 90;$
$V(5, 1) = d_1 = 72;$
$V(5, 2) = d_2 = 54;$
$V(5, 3) = d_3 = 36;$
$V(5, 4) = d_4 = 18.$

U sljedećem koraku računamo vrijednosti funkcije V za $k = 4$ i $i = 0, 1, 2, 3$ jer je najveća moguća starost opreme na početku 4. godine jednaka $i = 3$. Dobivamo:

$V(4, 0) = \max \{d_0 + V(4 + 1, 0 + 1), d_0 - p_4 + V(4 + 1, 1)\} = \max \{d_0 + V(5, 1), d_0 - p_4 + V(5, 1)\} = \max \{90 + 72, 90 - 146.41 + 72\} = 162;$
$V(4, 1) = \max \{d_1 + V(4 + 1, 1 + 1), d_0 - p_4 + V(4 + 1, 1)\} = \max \{d_1 + V(5, 2), d_0 - p_4 + V(5, 1)\} = \max \{72 + 54, 90 - 146.41 + 72\} = 126;$
$V(4, 2) = \max \{d_2 + V(4 + 1, 2 + 1), d_0 - p_4 + V(4 + 1, 1)\} = \max \{d_2 + V(5, 3), d_0 - p_4 + V(5, 1)\} = \max \{54 + 36, 90 - 146.41 + 72\} = 90;$
$V(4, 3) = \max \{d_3 + V(4 + 1, 3 + 1), d_0 - p_4 + V(4 + 1, 1)\} = \max \{d_4 + V(5, 4), d_0 - p_4 + V(5, 1)\} = \max \{36 + 18, 90 - 146.41 + 72\} = 54.$

U sljedećem koraku računamo vrijednosti funkcije V za $k = 3$ i $i = 0, 1, 2$ jer je najveća moguća starost opreme na početku 3. godine jednaka $i = 2$. Dobivamo:

$V(3, 0) = \max\{d_0 + V(3 + 1, 0 + 1), d_0 - p_3 + V(3 + 1, 1)\} = \max\{d_0 + V(4, 1), d_0 - p_3 + V(4, 1)\} = \max\{90 + 126, 90 - 121 + 126\} = 216;$
$V(3, 1) = \max\{d_1 + V(3 + 1, 1 + 1), d_0 - p_3 + V(3 + 1, 1)\} = \max\{d_1 + V(4, 2), d_0 - p_3 + V(4, 1)\} = \max\{72 + 90, 90 - 121 + 72\} = 162;$
$V(3, 2) = \max\{d_2 + V(3 + 1, 2 + 1), d_0 - p_3 + V(3 + 1, 1)\} = \max\{d_2 + V(4, 3), d_0 - p_3 + V(4, 1)\} = \max\{54 + 54, 90 - 121 + 72\} = 108.$

U sljedećem koraku računamo vrijednosti funkcije V za $k = 2$ i $i = 0, 1$ jer je najveća moguća starost opreme na početku 2. godine jednaka $i = 1$. Dobivamo:

$V(2, 0) = \max\{d_0 + V(2 + 1, 0 + 1), d_0 - p_2 + V(2 + 1, 1)\} = \max\{d_0 + V(3, 1), d_0 - p_2 + V(3, 1)\} = \max\{90 + 162, 90 - 110 + 162\} = 252;$
$V(2, 1) = \max\{d_1 + V(2 + 1, 1 + 1), d_0 - p_2 + V(2 + 1, 1)\} = \max\{d_1 + V(3, 2), d_0 - p_2 + V(3, 1)\} = \max\{72 + 108, 90 - 110 + 162\} = 180.$

Preostaje izračunati

$V(1, 0) = \max\{d_0 + V(1 + 1, 0 + 1), d_0 - p_1 + V(1 + 1, 1)\} = \max\{d_0 + V(2, 1), d_0 - p_1 + V(2, 1)\} = \max\{90 + 180, 90 - 72 + 180\} = 270.$

Iz navedenih je izračuna razvidno da se vrijednost $V(1, 0)$ računa pomoću vrijednosti $V(2, 1)$, $V(3, 2)$, $V(4, 3)$ i $V(5, 4)$. Sve navedene vrijednosti posljedica su odabira alternative "*zadržati dotadašnju opremu*" na početku odgovarajuće godine. Dakle, optimalna strategija za cijelo razdoblje planiranja je: *zadržati kupljenu opremu*, a to smo i htjeli pokazati.

Zadaci za vježbu:

1. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u sljedećih $n = 7$ godina. Zbog toga je početkom prve godine kupljeno vozilo po nabavnoj cijeni od $p_1 = 20$ n.j. Godišnji neto-troškovi održavanja i likvidacijska vrijednost vozila iskazani su kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4	5	6	7
godišnji neto–trošak (c_i) [n.j.]	2	5	8	11	13	17	19	20
likvidacijska vrijednost (s_i) [n.j.]	–	11	9	6	4	3	1	0

Uz pretpostavku da se početkom svake od sljedećih 6 godina donosi odluka o zadržavanju ili zamjeni postojećega vozila, odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja tako da ukupni neto–troškovi u cijelom razdoblju planiranja budu minimalni ako je nabavna cijena novoga vozila:

a) tijekom cijeloga razdoblja planiranja jednaka p_1;
b) u prve četiri godine razdoblja planiranja jednaka p_1, a u preostalim godinama za 10% veća;

2. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u sljedećih $n = 6$ godina. Na početku prve godine raspolažemo s vozilom starim točno dvije godine. Godišnji neto-troškovi održavanja i likvidacijska vrijednost vozila iskazani su kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4	5	6	7	8
godišnji neto - trošak (c_i) [n.j.]	4	6	8	10	13	15	17	19	20
likvidacijska vrijednost (s_i) [n.j.]	–	12	9	7	5	3	2	1	0

Uz pretpostavku da se u razdoblju planiranja početkom svake pojedine godine donosi odluka o zadržavanju ili zamjeni postojećega vozila, odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja tako da ukupni neto–troškovi budu minimalni ako je nabavna cijena novoga vozila:

a) tijekom cijeloga razdoblja planiranja jednaka $p = 20$ n.j. ;
b) u prve tri godine razdoblja planiranja jednaka $p = 20$ n.j., a u preostalim godinama za 10% veća;

3. Poslovna politika neke tvrtke predviđa proširenje proizvodnoga asortimana, pa u tu svrhu treba nabaviti novu opremu za ostvaraj predviđenoga poslovnoga plana. Nabavna cijena nove opreme je $p = 1000$ n.j. i ona se može koristiti najviše 4 godine, a potom je treba rashodovati. Godišnji neto–trošak održavanja opreme i likvidacijska vrijednost opreme iskazani su kao funkcija starosti opreme:

starost opreme (i) [god.]	0	1	2	3	4
godišnji trošak (c_i) [n.j.]	200	300	500	600	800
likvidacijska vrijednost (s_i) [n.j.]	–	400	200	120	100

Cilj je minimizirati ukupne neto–troškove u cjelokupnom razdoblju planiranja, pri čemu je nabavna cijena nove opreme stalna i jednaka p. Početkom svake pojedine godine donosi se odluka o zadržavanju ili zamjeni dotad korištene opreme. Odredite optimalnu politiku zamjene opreme za razdoblje planiranja u trajanju od:

a) $n = 4$ godine;
b) $n = 7$ godina.

4. Riješite prethodni zadatak za razdoblje planiranja od $n = 5$ godina uz sljedeće početne podatke:

- nabavna cijena nove opreme na početku i – te godine dana je formulom $p_i = 990 + 10 \cdot i$, za $i = 1, 2, 3, 4, 5$;
- godišnji neto–trošak održavanja opreme stare i godina dan je formulom $c_i = 200 + 50 \cdot i^2$, za $i = 0, 1, 2, 3$;
- likvidacijska vrijednost opreme stare i godina dana je formulom $s_i = 900 - 50 \cdot i^2$, za $i = 1, 2, 3, 4$.

5. Neka tvrtka na početku prve godine raspolaže s vozilom starim jednu godinu. Vozilo je potrebno koristiti u sljedeće $n = 4$ godine, pri čemu se na početku svake godine (uračunavajući i prvu!) donosi odluka o zadržavanju ili zamjeni dotad korištenoga vozila. Godišnji neto–trošak održavanja vozila zadan je tablično kao funkcija starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4

godišnji neto–trošak (c_i) [n.j.]	15	25	45	65	75

Likvidacijska vrijednost vozila staroga i godina zadana je funkcijom:

$$s_i = s(i) = 45 - 5 \cdot i, \text{ za } i = 1, 2, 3, 4, 5.$$

Cilj je minimizirati ukupne neto–troškove u cjelokupnom razdoblju planiranja. Odredite optimalnu politiku zamjene automobila u tom razdoblju ako je nabavna cijena novoga vozila:

a) stalna tijekom cijeloga razdoblja planiranja i iznosi $p = 70$ n.j.;
b) u prve dvije godine jednaka $p = 70$ n.j., a u preostalim godinama razdoblja planiranja za 10% veća.

6. Za opremu nekoga proizvodnoga pogona koja osigurava nesmetan i neprekidan proces proizvodnje zadani su nabavna cijena $p = 200$ n.j. i funkcija neto–dobiti nastale korištenjem opreme (d). Nakon određenoga vremena korištenja oprema se ne može prodati, pa početkom svake godine treba donijeti odluku o zamjeni ili zadržavanju dotad korištene opreme. Godišnja neto–dobit od korištenja opreme zadana je kao funkcija starosti opreme (i):

$$d_i = d(i) = \begin{cases} 200 - 40 \cdot i, & \text{za } i \in [5] \\ 0, & \text{za } i \geq 6 \end{cases}$$

Cilj je maksimizirati ukupnu neto–dobit u cjelokupnom razdoblju planiranja, pri čemu je nabavna cijena nove opreme stalna tijekom cijeloga razdoblja planiranja. Odredite optimalnu politiku zamjene opreme u razdoblju od $n = 7$ godina ako početkom prve godine:

a) obavezno kupujemo novu opremu;
b) raspolažemo s opremom starom jednu godinu;
c) raspolažemo s opremom starom dvije godine.

7. Riješite prethodni zadatak ako je godišnja neto–dobit zadana izrazom:

$$d_i = d(i) = \begin{cases} 180 - 30 \cdot i, & \text{za } i \in [6] \\ 0, & \text{za } i \geq 7 \end{cases}$$

ako je razdoblje planiranja $n = 8$ godina i ako na početku prve godine raspolažemo s opremom starom jednu godinu.

8. Riješite Zadatak 6. ako je godišnja neto-dobit zadana izrazom:

$$d_i = d(i) = \begin{cases} 250 - 50 \cdot i, & \text{za } i \in [5] \\ 0, & \text{za } i \geq 6 \end{cases}$$

ako je razdoblje planiranja $n = 7$ godina i ako na početku prve godine raspolažemo s opremom starom dvije godine.

1.6. Problem trgovačkoga putnika. Problem kineskoga poštara.

U ovoj ćemo točki razmotriti dva relativno stara, ali vrlo poznata problema čije je rješavanje vezano uz razvoj teorije grafova, a primjena na modeliranje niza praktičnih problema unutar operacijskih istraživanja. Prvi od tih problema je tzv. problem trgovačkoga putnika. Kao početak razmatranja problema trgovačkoga putnika navodi se sredina 19. stoljeća. Točnije, 1856. godine irski matematičar Wiliam Rowan Hamilton postavio je sljedeći problem:

***Problem* 1.** Trgovački putnik treba krenuti iz grada A, obići ukupno n gradova tako da svaki od njih posjeti točno jednom i vratiti se u polazište (grad A). Pritom svaka dva pojedina grada *ne* moraju nužno biti povezana izravnim putem. Naći (netrivijalan) nužan i dovoljan uvjet da trgovački putnik ostvari svoj plan, te, ako je plan ostvariv, odrediti način obilaska svih n gradova.

Navedeni je problem i danas jedan od najvećih neriješenih problema teorije grafova jer dosad nije otkriven navedeni netrivijalan nužan i dovoljan uvjet. Jedan od najjednostavnijih dovoljnih uvjeta je 1852. godine otkrio danski matematičar Gabriel Andrew Dirac i po njemu se navedeni uvjet naziva *Diracov uvjet*:

***Poučak* 1.** (G.A.Dirac, 1852.) Neka je G jednostavan neusmjeren graf s ukupno n vrhova takav da je svaki njegov vrh incidentan s barem $\frac{n}{2}$ ostalih vrhova. Tada G sadrži *Hamiltonov ciklus*, tj. postoji barem jedan vrh v takav da – krećući se iz vrha v po bridovima grafa G – svaki vrh možemo obići točno jednom i vratiti se u polazni vrh v.

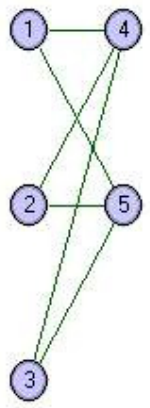
Današnja formulacija problema trgovačkoga putnika je uglavnom povezana s *potpunim težinskim* grafovima i dodatno zahtijeva izravnu povezanost bilo kojih dvaju gradova i zadavanje troškova prijevoza (npr. cijene prijevozne karte za vlak ili zrakoplov) između bilo kojih dvaju gradova. Osnovni netrivijalni rezultat je da *bilo koji* potpuni graf s ukupno $n \geq 2$ vrhova sadrži *Hamiltonov put*, tj. put koji sadrži sve vrhove grafa G – u slučaju potpunoga neusmjerenoga grafa taj je rezultat lagana posljedica Diracova poučka, ali njegovu valjanost u slučaju potpunoga usmjerenoga grafa tek je 1934. godine dokazao mađarski matematičar László Rédei. U takvim slučajevima se Problem 1. modificira u sljedeći problem:

***Problem* 2.** Trgovački putnik treba krenuti iz grada A, obići ukupno n gradova tako da svaki od njih posjeti točno jednom i vratiti se u polazište (grad A). Ako su zadani troškovi putovanja između bilo kojih dvaju različitih gradova, odrediti najjeftiniji način obilaska svih n gradova.

Ovaj problem je dobro definiran jer činjenica da su zadani troškovi putovanja između bilo kojih dvaju različitih gradova povlači da je matematički model Problema 2. potpun neusmjeren graf, a on, prema Diracovu poučku, uvijek sadrži barem jedan Hamiltonov ciklus. Zbog velike složenosti "gruboga" algoritma (uspoređivanje težina svih Hamiltonovih ciklusa) za određivanje Hamiltonova ciklusa najmanje težine danas postoje razni heuristički algoritmi koji daju razna praktično najbolja rješenja.

Računalno modeliranje i rješavanje Problema 1. pokazat ćemo na sljedeća tri primjera koristeći računalni program *Graph Magics*.

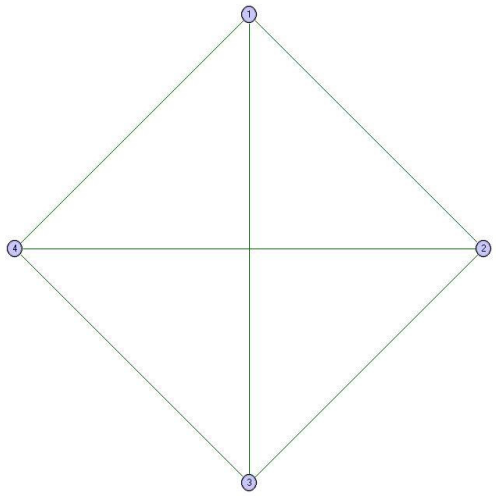
Primjer 1. Potpuni bipartitan graf $K_{3,\,2}$ nema Hamiltonov ciklus. U programu *Graph Magics* taj graf možemo generirati na sljedeći način: istodobno pritisnemo tipke `Ctrl` i G, pa unutar dijaloškoga okvira *Graph Generator* u pravokutnik ispod natpisa *Number of Vertices* upišemo 5 (jer $K_{3,\,2}$ ima ukupno $n = 2 + 3 = 5$ vrhova), u pravokutnik ispod natpisa *Number of Edges* upišemo 6 (jer $K_{3,\,2}$ ima ukupno $2 \cdot 3 = 6$ bridova) ili bilo koji prirodan broj strogo veći od 6, te lijevim klikom miša označimo kružić pored natpisa *Bipartite Graph* i kvadratić pored natpisa *Connected Graph* (u dijelu dijaloškoga okvira pod nazivom *Graph Type*). Potom lijevom tipkom miša kliknemo na *Generate*, potom na *Export* i na *OK*. Kao rezultat dobivamo sljedeći graf (ili neki izomorfan njemu):



Slika 19. Potpun bipartitan graf $K_{3,\,2}$

Iz Diracova poučka izravno slijedi da dobiveni graf ne sadrži Hamiltonov ciklus jer su vrhovi 1, 2 i 3 incidentni s $2 < 2.5 = \frac{5}{2}$ ostalih vrhova (za postojanje Hamiltonova ciklusa svaki od tih bi vrhova trebao bi biti incidentan s barem tri različita vrha grafa).

Primjer 2. Potpun graf K_4 ima Hamiltonov ciklus. U programu *Graph Magics* taj graf generiramo na sljedeći način: istodobno pritisnemo tipke `Ctrl` i G, pa unutar dijaloškoga okvira *Graph Generator* u pravokutnik ispod natpisa *Number of Vertices* upišemo 4 (jer K_4 ima ukupno 4 vrha), u pravokutnik ispod natpisa *Number of Edges* upišemo 6 (jer K_4 ima ukupno $\binom{4}{2} = 6$ bridova) ili bilo koji prirodan broj strogo veći od 6, te lijevim klikom miša označimo kružić pored natpisa *General* i kvadratić pored natpisa *Connected Graph* (u dijelu dijaloškoga okvira pod nazivom *Graph Type*). Potom lijevom tipkom miša kliknemo na *Generate*, potom na *Export* i na *OK*. Kao rezultat dobivamo sljedeći graf (ili neki izomorfan njemu):

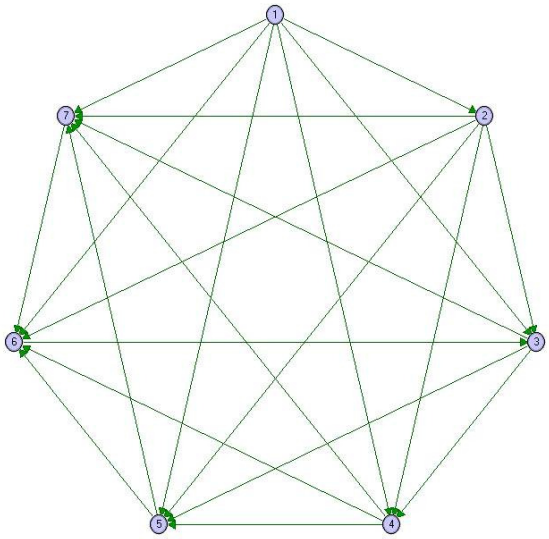


Slika 20. Potpuni neusmjereni graf K_4

Jedan Hamiltonov ciklus dobivamo tako da lijevom tipkom miša kliknemo na izbornik *Graph*, odaberemo opciju *Find* i podopciju *Hamilton Circuit*:

1 – 2 – 3 – 4 – 1.

<u>Primjer 3.</u> Jedan od mogućih potpunih digrafova (ili, kraće, *turnira*) izgleda ovako:



Slika 21. Potpuni digraf sa 7 vrhova

Jedan mogući Hamiltonov put u navedenom grafu je npr. 1 – 2 – 3 – 4 – 5 – 6 – 7 (možemo ga dobiti odabirom opcije *Hamiltonian Path* unutar podizbornika *Find* izbornika *Graph*). No, navedeni graf ne sadrži i Hamiltonov ciklus jer iz vrha 1 imamo samo izlazne lukove, pa u taj vrh ne možemo doći niti iz jednoga drugoga vrha grafa.

Pokažimo sada na primjeru kako se rješava problem trgovačkoga putnika. Napomenimo da računalni program *Graph Magics ne* rješava problem trgovačkoga putnika, već samo određuje Hamiltonov ciklus neovisno o zadanim težinama. Stoga ćemo za rješavanje problema

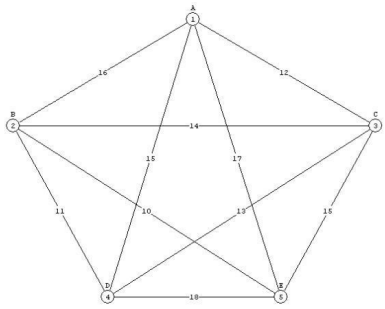
trgovačkoga putnika primijeniti računalni program *Grin* koji u sklopu raznih algoritama na grafovima ima ugrađen jedan od heurističkih algoritama za rješavanje problema trgovačkoga putnika (*Property* ⇒ *Network* ⇒ *Salesman Problem*).

Primjer 4. Trgovački putnik treba krenuti iz grada *A*, obići svaki od gradova *B*, *C*, *D* i *E* točno jednom, te se vratiti u polazište. Troškovi putovanja između pojedinih gradova (iskazani u €) zadani su sljedećom tablicom:

grad	*A*	*B*	*C*	*D*	*E*
A	0	16	12	15	17
B	16	0	14	11	10
C	12	14	0	13	15
D	15	11	13	0	18
E	17	10	15	18	0

Odredimo najjeftiniju rutu obilaska sva četiri grada i izračunajmo pripadne minimalne ukupne troškove putovanja.

Matematički model promatranoga problema je sljedeći potpuni neusmjereni težinski graf:



Slika 22. Model problema iz Primjera 4.

Lijevom tipkom miša kliknimo na izbornik *Property*, odaberimo opciju *NetWork* i podopciju *Salesman Problem*. Odaberimo vrh 1 kao polazište (*Source Point*), pa dobivamo:

```
Cycle :
      1, 5, 2, 4, 3, 1.
Weight of the Cycle =  63
```

Prema tome, tražena najjeftinija ruta trgovačkoga putnika je:

$$A - E - B - D - C - A$$

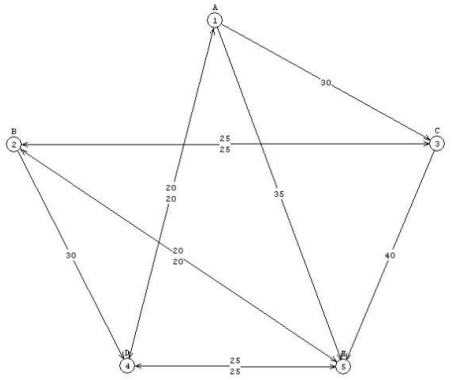
i njezina ukupna težina je w_{min} = 63, što znači da najmanji ukupni troškovi putovanja iznose 63 €.

Primjer 5. Trgovački putnik kreće s lokacije *A*, obilazi lokacije *B*, *C*, *D* i *E*, te se vraća na lokaciju *A*. Pri obilasku može koristiti jednosmjerne ulice *AC*, *AE*, *BD* i *CE*, te dvosmjerne ulice *AD*, *BC*, *BE* i *DE*. Trajanje putovanja između pojedinih lokacija (iskazano u minutama) zadano je u sljedećoj tablici.

lokacija	*A*	*B*	*C*	*D*	*E*
A	0	0	30	20	35
B	0	0	25	30	20
C	0	25	0	0	40
D	20	0	0	0	25
E	0	20	0	25	0

Odredimo rutu obilaska svih četiriju lokacija za koju je potrebno najmanje vremena i izračunajmo pripadno trajanje putovanja.

Matematički model promatranoga problema je sljedeći graf:



Slika 23. Model problema iz Primjera 5.

Lijevom tipkom miša kliknimo na izbornik *Property*, odaberimo opciju *NetWork* i podopciju *Salesman Problem*. Odaberimo vrh 1 kao polazište (*Source Point*), pa dobivamo:

```
Cycle :
      1, 3, 2, 5, 4, 1.
Weight of the Cycle =  120
```

Prema tome, tražena ruta obilaska svih četiriju lokacija za koju je potrebno najmanje vremena je:

$$A - C - B - E - D - A$$

a njezina težina je w_{min} = 120, pa je traženo najmanje ukupno vrijeme obilaska svih četiriju lokacija 120 minuta (= 2 sata).

U prethodnim smo primjerima promatrali problem obilaska svih vrhova zadanoga grafa tako da se svaki vrh obiđe točno jednom. Svojevrstan pandan tome problemu je problem obilaska svih *bridova* (ili *lukova*) zadanoga (ne)usmjerenoga grafa tako da se svaki brid (luk) grafa obiđe točno jednom. Početak razmatranja toga problema vezan je uz same početke nastanka teorije grafova, odnosno uz poznati *problem königsberških mostova* koji je još 1736. godine uspješno riješio slavni švicarski matematičar Leonhard Euler[28]. Općenita formulacija problema je sljedeća:

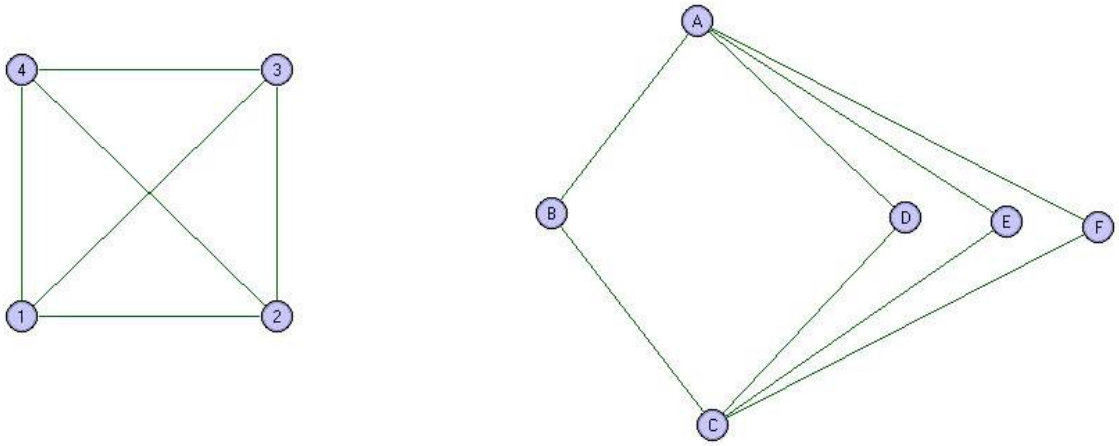
***Problem* 3.** Za zadani povezani[29] graf G odrediti postoji li staza, odnosno *tura* (zatvorena staza) koja prolazi svim bridovima (lukovima) grafa.

Staza, odnosno tura iz Problema 3. nazivaju se *Eulerova staza* (*Eulerova tura*), a graf koji dozvoljava Eulerovu turu naziva se *Eulerov graf*. Stoga Problem 3. zapravo traži utvrđivanje nužnih i dovoljnih uvjeta da bilo koji zadani povezani graf bude Eulerov. Najpoznatiji takvi uvjeti su:

***Poučak* 2.** Povezan graf G ima Eulerovu stazu ako i samo ako ima najviše dva vrha neparnoga stupnja. Ekvivalentno, povezan graf G ima Eulerovu stazu ako i samo ako su najviše dva njegova vrha susjedni s neparnim brojem preostalih vrhova grafa[30].

***Poučak* 3.** Povezani graf G je Eulerov ako i samo ako je stupanj svakoga njegovoga vrha paran prirodan broj. Ekvivalentno, povezani graf G je Eulerov ako i samo ako je svaki njegov vrh susjedan s parnim brojem preostalih vrhova grafa.

Iako ih se može shvatiti kao međusobne pandane, Eulerovi i Hamiltonovi grafovi zapravo nemaju izravne veze. Npr. lako se pokaže da je potpuni neusmjereni graf K_4 (lijevi od dvaju grafova na Slici 24.) Hamiltonov, ali ne i Eulerov, te da je desni od dvaju grafova na donjoj slici Eulerov, ali ne i Hamiltonov.



Slika 24. Primjer Hamiltonova, ali ne i Eulerova grafa i primjer Eulerova, ali ne i Hamiltonova grafa.

[28] Problem königsberških mostova i njegovo rješavanje razmatrani su u sklopu kolegija *Ekonomska matematika* 2.
[29] Podsjetimo da je graf G povezan ako postoji staza između bilo kojih dvaju različitih vrhova toga grafa.
[30] Iz navedenoga poučka izravno slijedi da graf mostova u Königsbergu nema Eulerovu stazu, tj. da željeni obilazak mostova nije moguć.

Opći problem utvrđivanja je li neki graf Eulerov ili nije konkretizirao je kineski matematičar Mei–Ko Kwan 1962. godine postavivši tzv. *problem kineskoga poštara*[31]:

***Problem* 4.** U nekom gradu poštar pokupi poštu u glavnom poštanskom uredu, odlazi razdijeliti je u dodijeljeni okrug i potom se vraća u glavni poštanski ured. Pri dijeljenju pošte poštar obavezno mora barem jednom proći svakom ulicom svojega okruga. Ako su zadane duljine svih ulica u okrugu, odrediti rutu kojom mora hodati poštar tako da ukupan prijeđeni put bude što kraći.

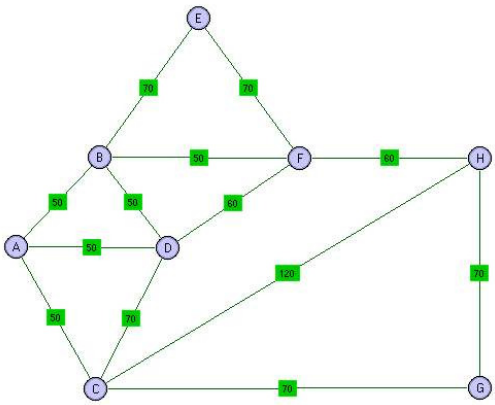
Navedeni se problem ekvivalentno može formulirati kao *problem autobusne rute*, i to na sljedeći način:

***Problem* 5.** Kako bi smanjio troškove za gorivo potrebno za rad autobusa, autobusni je prijevoznik modelirao autobusna stajališta kao vrhove, a ulice kao bridove (lukove) rute svojih autobusa. Ako je poznata duljina svake ulice, odrediti autobusnu rutu koja barem jednom prolazi svakom ulicom tako da troškovi za gorivo budu što manji.

Istaknimo odmah najjednostavniji slučaj: Ukoliko je graf G Eulerov graf, onda je svaka Eulerova tura na grafu G optimalna jer se njome svaki brid prijeđe točno jednom. U takvom je slučaju dovoljno naći bilo koju Eulerovu turu i odrediti njezinu težinu. Najpoznatiji algoritam za konstruiranje takve ture je *Fleuryjev algoritam* kojega je otkrio jedan od manje poznatih francuskih matematičara Lou Fleury 1883. godine. Ukoliko, pak, graf G nije Eulerov graf, za rješenje problema kineskoga poštara koristi se *Edmonds–Johnsonov algoritam* kojega su 1973. godine zajednički objavili kanadski matematičar Jack Edmonds i američki matematičar Ellis L. Johnson. Jedna od varijanti toga algoritma za bilo koji tip grafa implementirana je u pogramu *Graph Magics*, pa ćemo je koristiti u sljedećim primjerima.

Primjer 6. Neki kvart se sastoji od ukupno 13 dvosmjernih ulica. Ulice su određene svojim početnim/krajnjim točkama A, B, C, D, E, F, G i H takvima da je $|AB| = |AC| = |AD| = |BD| = |BF| = 50$ m, $|DF| = |FH| = 60$ m, $|BE| = |CD| = |CG| = |EF| = |GH| = 70$ m i $|CH| = 120$ m. Poštar kreće iz točke A, obilazi svih 13 ulica barem jednom i vraća se natrag u točku A. Odredimo rutu takvu da ukupna prijeđena udaljenost bude najmanja moguća.

Matematički model promatranoga problema je sljedeći neusmjereni graf:



Slika 25. Model problema iz Primjera 6.

[31] Mei–Ko Kwan je navedeni problem nazvao *problem poštara* (analogno problemu trgovačkoga putnika), a njemu u čast problem je dobio današnji naziv zahvaljujući američkom matematičaru Alanu Goldmanu.

Navedeni ćemo problem riješiti pokretanjem procedure *Chinese Postman Problem*. Ta se procedura nalazi na podizborniku *Find* izbornika *Graph*. Njezinim pokretanjem dobivamo sljedeće rješenje:

```
Circuit's Total Cost - 1000
Circuit's Total Length (Edges passed) - 16

Path:

A >> C >> H >> F >> H >> G >> C >> D >> F >> B >> F >> E >> B >> D >> A >> B >> A
```

Stoga je tražena ruta:

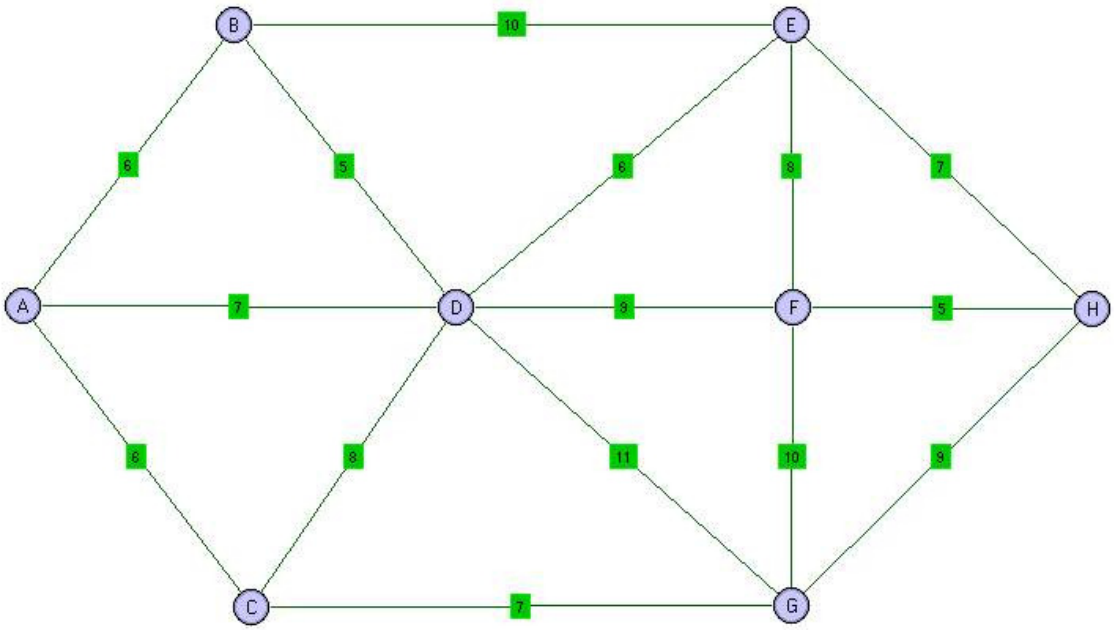
AC – CH – HF – FH – HG – GC – CD – DF – FB – BF – FE – EB – BD – DA – AB – BA

i njezina je duljina jednaka 1000. Kako vidimo, poštar će morati dvaput proći ulicama *BF*, *FH* i *AB*, te će hodajući navedenom rutom prijeći ukupno 1000 metara, odnosno 1 km. Primijetimo da rješenje zadatka nije jednoznačno jer je npr. i ruta

AD – DC – CG – GH – HC – CA – AB – BD – DF – FB – BE – EF – FH – HF – FB – BA

duljine 1000 m također rješenje zadatka.

<u>Primjer 7.</u> Uprava za ceste nekoga područja odgovorna je za održavanje donje mreže cesta:



Slika 26. Mreža cesta iz Primjera 26.

(Težina pojedinoga brida jednaka je duljini [u km] odgovarajuće ceste.) Sjedište uprave je na lokaciji *A*.

a) Čelnik uprave za ceste želi provjeriti stanje na svakoj pojedinoj cesti tako da krene iz sjedišta uprave, obiđe svaku pojedinu cestu i vrati se u sjedište uprave. Odredimo rutu njegova obilaska tako da u tom obilasku prijeđe što manju ukupnu udaljenost.

b) Utvrdimo je li moguć obilazak svake pojedine ceste *točno jednom*, pri čemu polazište i odredište mogu biti dvije različite lokacije. Obrazložimo svoj odgovor.

c) Riješimo prethodni zadatak uz dodatan uvjet da se izgradi dionica *BC* duga 8 kilometara.

U prvom podzadatku rješavamo klasičan problem kineskoga poštara. Modeliranjem primjera u programu *Graph Magics* i pokretanjem procedure *Chinese Postman Problem* dobivamo sljedeći rezultat:

```
Circuit's Total Cost - 136
Circuit's Total Length (Edges passed) - 18

Path:

A >> D >> G >> H >> G >> C >> G >> F >> H >> E >> F >> D >> E >> B >> D >>
C >> A >> B >> A
```

Dakle, tražena ruta je:

AD – DG – GH – HG – GC – CG – GF – FH – HE – EF – FD – DE – EB – BD – DC – CA – AB – BA

i njezina je težina jednaka 136, što znači da će čelnik uprave za ceste u navedenom optimalnom obilasku prevaliti ukupno 136 km.

U drugom podzadatku provjeravamo nužan i dovoljan uvjet za postojanje Eulerove staze. Lako se vidi da je zadani graf povezan, pa ćemo primijeniti Poučke 2. i 3. Uočimo da je vrh *A* susjedan s točno tri vrha grafa *G* (to su vrhovi *B*, *C* i *D*), tj. njegov je stupanj jednak 3, pa iz Poučka 3. slijedi da ne postoji Eulerova tura. Dakle, polazište i odredište ne mogu biti na istoj lokaciji, pa je najbolji rezultat koji eventualno možemo dobiti Eulerova staza. U tu svrhu provjeravamo uvjete iz Poučka 2., tj. određujemo stupnjeve vrhova grafa *G*. Uočimo da su vrhovi *A*, *B* i *C* susjedni s točno tri vrha grafa *G*, tj. njihovi su stupnjevi jednaki 3, pa graf *G* sadrži najmanje 3 vrha neparnoga stupnja. Prema Poučku 2. graf *G* ne sadrži Eulerovu stazu, pa traženi obilazak nije moguć, tj. postoji barem jedna cesta koju će čelnik uprave za ceste morati obići barem dva puta. Dobiveni odgovor možemo provjeriti i koristeći računalni program *Graph Magics* i proceduru *Eulerian Path/Circuit*.

Ukoliko se, pak, izgradi dionica *BC* duga 8 km, lako se vidi da će u tom slučaju dobiveni graf *G'* sadržavati točno dva vrha neparnoga stupnja, i to su vrhovi *A* i *H* čiji je stupanj 3. Stoga će graf *G'* sadržavati Eulerovu stazu, ali ne i Eulerovu turu. Traženu Eulerovu stazu u tom slučaju dobivamo primjenom procedure *Eulerian Path/Circuit* programa *Graph Magics*:

```
Start Vertex - A

Path:
A >> B >> C >> A >> D >> B >> E >> D >> C >> G >> D >> F >> E >> H >> F >> G >> H
```

pa je polazište lokacija *A*, odredište lokacija *H*, a ruta:

AB – BC – CA – AD – DB – BE – ED – DC – CG – GD – DF – FE – EH – HF – FG – GH.

Zadaci za vježbu:

1. Za koje je vrijednosti prirodnoga broja $n \in \mathbf{N}$ potpuni neusmjeren graf K_n istodobno Eulerov i Hamiltonov? (*Naputak*: Koristite Poučak 3.)

2. Trgovački putnik treba krenuti iz grada *A*, obići svaki od gradova *B*, *C*, *D, E* i *F* točno jednom, te se vratiti u polazište. Troškovi putovanja između pojedinih gradova [u USD] zadani su sljedećom tablicom:

grad	*A*	*B*	*C*	*D*	*E*	*F*
A	0	17	15	12	16	19
B	17	0	16	14	13	12
C	15	16	0	15	17	16
D	12	14	15	0	16	14
E	16	13	17	16	0	18
F	19	12	16	14	16	0

Modelirajte promatrani problem odgovarajućim grafom, pa odredite najjeftiniju rutu obilaska svih pet gradova i izračunajte pripadne minimalne ukupne troškove putovanja.

3. Trgovački putnik treba krenuti iz lokacije *A*, obići svaku od lokacija *B*, *C*, *D* i *E* točno jednom, te se vratiti u polazište. Ulice kojima se može kretati putnik su isključivo jednosmjerne, pa su međusobne udaljenosti pojedinih lokacija [u km] zadane sljedećom tablicom:

lokacija	*A*	*B*	*C*	*D*	*E*
A	0	6	11	3	4
B	7	0	14	8	10
C	12	5	0	10	2
D	6	15	7	0	5
E	4	9	8	13	0

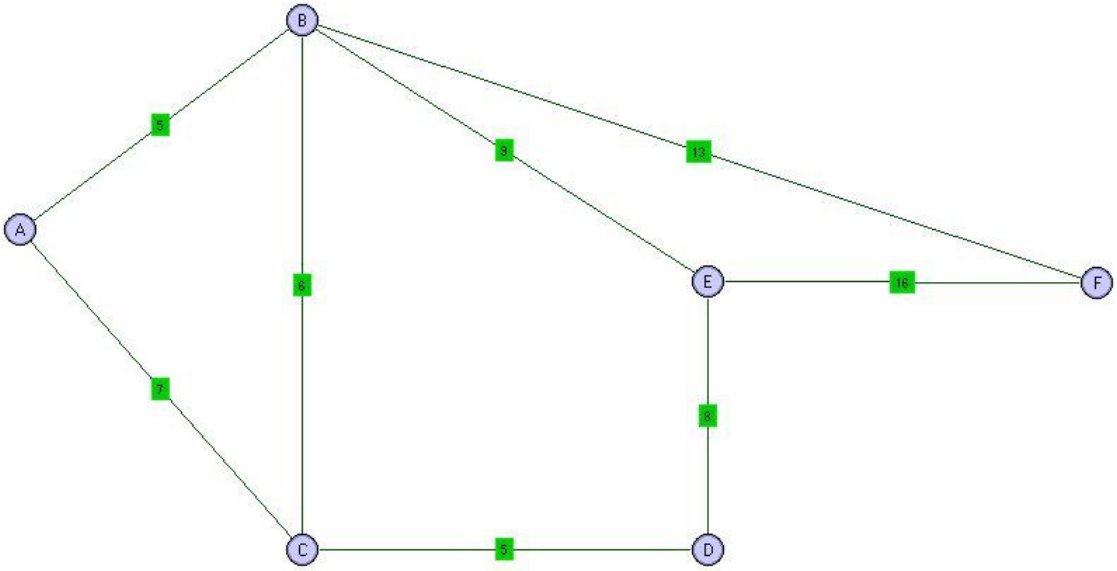
Modelirajte promatrani problem odgovarajućim grafom, pa odredite najkraću rutu obilaska svih četiriju lokacija i njezinu duljinu.

4. Trgovački putnik kreće s lokacije *A*, obilazi lokacije *B*, *C*, *D* i *E*, te se vraća na lokaciju *A*. Pri obilasku može koristiti jednosmjerne ulice *CA*, *CB, DB* i *ED*, te dvosmjerne ulice *AD*, *BE* i *CE*. Trajanje putovanja između pojedinih lokacija [u minutama] zadano je u sljedećoj tablici.

lokacija	*A*	*B*	*C*	*D*	*E*
A	0	0	0	20	0
B	0	0	0	0	25
C	25	20	0	0	30
D	15	20	0	0	0
E	0	25	15	20	0

Modelirajte promatrani problem odgovarajućim grafom, pa odredite rutu obilaska svih četiriju lokacija za koju je potrebno najmanje vremena i pripadno trajanje putovanja.

5. Uprava za ceste nekoga područja odgovorna je za posipanje cesta solju u zimskim uvjetima. Mreža dvosmjernih cesta zajedno s duljinama svake pojedine ceste [u km] prikazana je na donjoj slici.



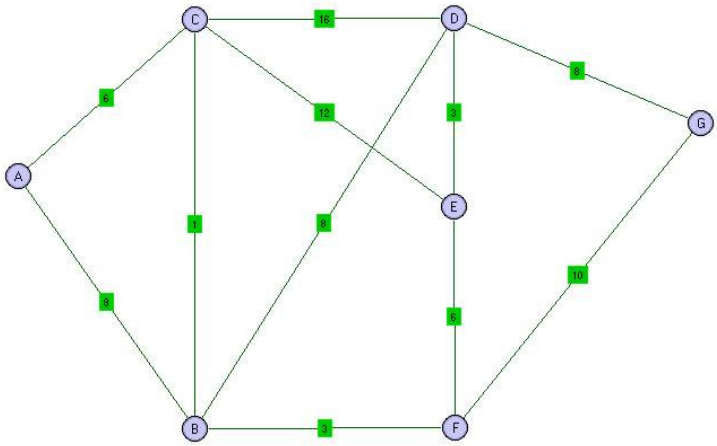
Slika 27. Mreža dvosmjernih cesta iz Zadatka 5.

Vozilo za posipanje cesta solju kreće s lokacije *A*, barem jednom obilazi svaku pojedinu cestu i vraća se na lokaciju *A*.

a) Odredite rutu kojom se treba kretati vozilo tako da ukupna duljina puta prijeđenoga na toj ruti bude što manja.

b) Je li moguće odabrati rutu tako da vozilo *točno jednom* obiđe svaku pojedinu cestu? Obraložite svoj odgovor.

6. Riješite prethodni zadatak ako je mreža dvosmjernih cesta zajedno s pripadnim udaljenostima prikazana na donjoj slici.



Slika 28. Mreža dvosmjernih cesta iz Zadatka 6.

2. *Riješeni pismeni ispiti iz predmeta Operacijska istraživanja*

2.1. **Primjer 1.**

1. Na raspolaganju nam je točno 5 komada svake od namirnica N_1 i N_2, točno 6 komada svake od namirnica N_3 i N_4, te ruksak najvećega dozvoljenoga kapaciteta 20 kg. Podaci o masi i neto–cijeni jednoga komada svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*neto–cijena* [kn]	*masa* [kg]
N_1	35.00	2.2
N_2	37.00	2.5
N_3	30.00	1.5
N_4	33.00	2

Niti jedan komad bilo koje namirnice nije dozvoljeno razdijeliti na manje dijelove. U ruksak treba staviti namirnice što veće ukupne vrijednosti tako da njihova ukupna masa ne bude veća od kapaciteta ruksaka.

a) Formirajte odgovarajući matematički model, pa odredite optimalan izbor namirnica i izračunajte pripadnu optimalnu ukupnu vrijednost.

b) Ako je optimalna količina neke vrste namirnica jednaka nuli, utvrdite kako se povećanje optimalne količine te vrste namirnica za 1 odražava na optimalne vrijednosti ostalih vrsta namirnica.

Rješenje: Za svaki $i \in [4]$ označimo s x_i broj komada namirnice N_i koje ćemo staviti u ruksak. Tada je matematički model promatranoga problema

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 35.00 \cdot x_1 + 37.00 \cdot x_2 + 30.00 \cdot x_3 + 33.00 \cdot x_4$$

pod uvjetima

$$2.2 \cdot x_1 + 2.5 \cdot x_2 + 1.5 \cdot x_3 + 2 \cdot x_4 \leq 20$$

$$x_1, x_2 \in [5]_0,\ x_3, x_4 \in [6]_0$$

U rješavanju ovoga problema koristit ćemo potprogram *DP* (*Dynamic Programming*) programa *Winqsb*. Prigodom pokretanja toga programa na izborniku *File* treba odabrati opciju *New Problem*, a potom na dobivenom izborniku kliknuti na kružić pored natpisa *Knapsack Problem* i u pravokutnik pored natpisa *Number of Items* upisati 4. (Naslov problema može biti proizvoljan, npr. *Zadatak* 1.) Dobiva se tablica koju po stupcima treba popuniti na sljedeći način:

Item Identification: N1, N2, N3, N4
Units Available: 5, 5, 6, 6
Unit Capacity Required: 2.2, 2.5, 1.5, 2
Return Function (*X*: *Item ID*): 35.00*X, 37.00*X, 30.00*X, 34.00*X
Knapsack Capacity: 20

a) Lijevim klikom miša na ikonicu *Solve the Problem* dobiva se tablica u čijem se stupcu *Decision Quantity* (*X*) nalaze početne optimalne vrijednosti svih četiriju nezavisnih varijabli:

$$x_1^* = 0,\ x_2^* = 1,\ x_3^* = x_4^* = 5.$$

Dakle, optimalan izbor namirnica je: *jedan komad namirnice* N_2, *te po pet komada svake od namirnica* N_3 *i* N_4;

b) Iz rezultata **a)** podzadatka slijedi da je optimalna količina namirnice N_1 jednaka 0. Stoga tu količinu povećamo za 1. Lijevim klikom miša kliknemo na ikonicu *What If*. U retku *N*1 stupca *Preset Decision* upišimo 1 i kliknimo na *OK*. Tako dobivamo traženo optimalno rješenje:

$$x_1^* = 1,\ x_2^* = 1,\ x_3^* = 4,\ x_4^* = 5.$$

Dakle, povećanje optimalne količine namirnice *N*1 ne utječe na promjenu optimalnih vrijednosti namirnica N_2 i N_4, a smanjuje optimalnu količinu namirnice N_3.

2. Zrakoplovom nosivosti 83 tone na nekoj se liniji mogu prevoziti putnici, poštanske pošiljke i ostali teret. Prosječna masa i neto–dobit ostvarena po jedinici transporta navedeni su u donjoj tablici.

	1 *putnik*	1 *poštanski paket*	1 *paket ostaloga tereta*
prosječna masa [t]	0.08	0.75	0.85
neto – dobit [000 n.j.]	2.9	2.8	3

Zrakoplovom je odjednom moguće prevesti najviše 50 putnika. Treba odrediti strukturu transporta tako da ukupna neto–dobit od jednoga transporta bude što veća.

a) Formirajte odgovarajući matematički model, pa odredite optimalnu strukturu transporta i pripadnu optimalnu ukupnu neto–dobit.

b) Za koliko će se postotaka promijeniti optimalna neto-dobit ukoliko se postavi dodatni uvjet da se zrakoplovom može prevesti najviše 100 poštanskih paketa?

c) Za koliko će se postotaka promijeniti optimalna neto-dobit ukoliko se postavi dodatni uvjet da zrakoplovom treba prevesti točno 20 paketa ostaloga tereta?

Rješenje: **a)** Neka je x_1 broj putnika, x_2 broj poštanskih paketa, a x_3 broj ostalih paketa koji će se prevesti jednim transportom. Tada je matematički model promatranoga problema

$$\text{maksimizirati } f(x_1, x_2, x_3) = 2.9 \cdot x_1 + 2.8 \cdot x_2 + 3 \cdot x_3$$

pod uvjetima

$$0.08 \cdot x_1 + 0.75 \cdot x_2 + 0.85 \cdot x_3 \le 83$$

$$x_1 \in [50]_0,$$

$$x_2, x_3, x_4 \in \mathbf{N}_0.$$

Da bismo mogli iskoristiti potprogram *DP*, moramo izračunati i najveće moguće vrijednosti varijabli x_2 i x_3 (najveća moguća vrijednost varijable x_1 je zadana i iznosi 50):

$$(x_2)_{max} = \left\lfloor \frac{83}{0.75} \right\rfloor = 110,\ (x_3)_{max} = \left\lfloor \frac{83}{0.85} \right\rfloor = 97.$$

Kao u rješenju prethodnoga zadatka, pokrenimo potprogram *DP* (*Dynamic Programming*) programa *Winqsb*, na izborniku *File* odaberimo opciju *New Problem*, potom na dobivenom izborniku kliknimo na kružić pored natpisa *Knapsack Problem* i u pravokutnik pored natpisa *Number of Items* upišimo: 3. (Naslov problema može biti proizvoljan, npr. *Zadatak* 2.) Dobiva se tablica koju po stupcima treba popuniti na sljedeći način:

Item Identification: putnici, pošta, ostali teret
Units Available: 50, 110, 97
Unit Capacity Required: 0.08, 0.75, 0.85
Return Function (*X*: *Item ID*): 2.9*X, 2.8*X, 3*X
Knapsack Capacity: 83

Lijevim klikom miša na ikonicu *Solve the Problem* dobiva se tablica u čijem se stupcu *Decision Quantity* (*X*) nalaze optimalne vrijednosti svih triju nezavisnih varijabli:

$$x_1^* = 50,\ x_2^* = 103,\ x_3^* = 2.$$

Prema tome, *optimalna struktura transporta* je: 50 *putnika*, 103 *poštanska paketa i* 2 *paketa ostaloga tereta*. Pripadna optimalna ukupna neto-dobit navedena je u retku *Total Return Value* i iznosi 439 400 n.j.

b) Zatvorimo prozor dobiven u rješenju **a)** podzadatka, pa u originalnoj postavci u presjeku retka *pošta* i stupca *Units Available* upišimo 100. Potom ponovno kliknimo na ikonicu *Solve the Problem*, pa dobivamo novo optimalno rješenje:

$$x_1^* = 50,\ x_2^* = 95,\ x_3^* = 9.$$

Dakle, *optimalna struktura transporta* je: 50 *putnika*, 95 *poštanskih paketa* i 9 *paketa ostaloga tereta*. Pripadna optimalna ukupna neto-dobit navedena je u retku *Total Return Value* i iznosi 438 000 n.j. Stoga je traženi postotak jednak

$$p = \frac{439.4 - 438}{439.4} \cdot 100 = 0{,}32\ (\%),$$

tj. dodavanjem navedenoga uvjeta početna optimalna ukupna neto-dobit smanjit će se za približno 0.32%.

c) Zatvorimo prozor dobiven u rješenju **b)** podzadatka, pa u originalnoj postavci u presjeku retka *pošta* i stupca *Units Available* ponovno upišimo 110. Potom ponovno kliknimo na ikonicu *Solve the Problem* tako da dobijemo optimalno rješenje iz **a)** podzadatka. Sada kliknimo na ikonicu *What If*, pa u presjeku retka *ostali teret* i stupca *Preset Decision* lijeve tablice upišimo: 20. Tako dobivamo novo optimalno rješenje uz zadani dodatni uvjet:

$$x_1^* = 50,\ x_2^* = 82,\ x_3^* = 20.$$

Dakle, *optimalna struktura transporta* u ovom je slučaju: 50 *putnika*, 82 *poštanska paketa i* 20 *paketa ostaloga tereta*. Pripadna optimalna ukupna neto-dobit navedena je u pravokutniku pored natpisa *Total Return* i iznosi 434 600 n.j. Stoga je traženi postotak jednak:

$$p_1 = \frac{439.4 - 434.6}{439.4} \cdot 100 = 1{,}09\ (\%),$$

tj. dodavanjem navedenoga uvjeta početna optimalna ukupna neto-dobit smanjit će se za približno 1.09%.

3. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u razdoblju planiranja od $n = 4$ godine. Nabavna cijena novoga vozila tijekom cijeloga razdoblja planiranja je stalna i iznosi $p = 20$ n.j. Godišnji neto-trošak održavanja i likvidacijska vrijednost vozila iskazani su tablično kao funkcije starosti automobila:

starost vozila (i) [god.]	0	1	2	3	4	5	6
godišnji neto–trošak (c_i) [n.j.]	5	8	12	14	17	19	20
likvidacijska vrijednost (s_i) [n.j.]	-	11	7	5	4	2	0

Početkom svake godine razdoblja planiranja donosi se odluka o zadržavanju ili zamjeni dotad korištenoga vozila tako da ukupni troškovi u cijelom razdoblju planiranja budu što manji. Formirajte odgovarajući matematički model, pa odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja ako se na početku prve godine razdoblja planiranja:

a) donosi odluka o obaveznoj nabavi novoga vozila;

b) raspolaže s vozilom starim jednu godinu (nije nužno nabaviti novo vozilo).

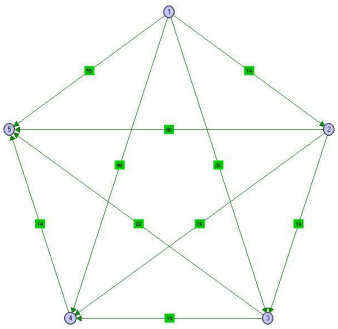
Rješenje: **a)** Budući da smo na početku prve godine već donijeli odluku o nabavi novoga vozila, za rješenje podzadatka iskoristit ćemo računalni program *Graph Magics*. (Podzadatak je moguće riješiti i koristeći računalni program *Grin*.) Matematički model u ovom je slučaju potpuni usmjereni težinski graf s ukupno 5 vrhova: 1, 2, 3, 4, 5, pri čemu vrh i označava početak godine i. Za svaki uređeni par $(i, j) \in [5]^2$ takav da je $i < j$ vrhove i i j spojit ćemo lukom (i, j) čija je težina w_{ij} definirana izrazom:

w_{ij} = (nabavna cijena vozila na početku godine i) + (troškovi održavanja vozila u godinama i, $i + 1, \ldots, j - 1$) – (likvidacijska vrijednost vozila na početku godine j) $= p + \sum_{k=0}^{j-i-1} c_k - s_{j-i}$.

Tako redom dobivamo:

$w_{12} = w_{23} = w_{34} = w_{45} = p + c_0 - s_1 = 20 + 5 - 11 = 14;$
$w_{13} = w_{24} = w_{35} = p + c_0 + c_1 - s_2 = 20 + 5 + 8 - 7 = 26;$
$w_{14} = w_{25} = p + c_0 + c_1 + c_2 - s_3 = 20 + 5 + 8 + 12 - 5 = 40;$
$w_{15} = p + c_0 + c_1 + c_2 + c_3 - s_4 = 20 + 5 + 8 + 12 + 14 - 4 = 55.$

Koristeći program *Graph Magics* dobivamo graf prikazan na slici 29.:



Slika 29.

u kojemu tražimo najkraći put između vrha 1 i vrha 5. Desnim klikom miša na vrh 1 i odabirom opcije *Set/Unset as Start Vertex* (*Source*) označimo vrh 1 kao početni, a desnim klikom miša na vrh 5 i odabirom opcije *Set/Unset as End Vertex* (*Sink*) označimo vrh 5 kao krajnji vrh puta. Desnim klikom miša na bilo koju bjelinu unutar prozora, odabirom opcije *Find* i podopcije *Shortest Path* (*from Star Vertex to End*) dobivamo traženi najkraći put:

1 – 3 – 5

čija je težina 52. Iz ovoga rezultata slijedi da početkom 3. godine treba kupiti novo vozilo, a u svim ostalim godinama zadržati dotad korišteno vozilo. Stoga je tražena optimalna politika:

<u>*početak* 2. *godine*</u>: zadržati dotad korišteno vozilo;
<u>*početak* 3. *godine*</u>: zamijeniti (kupiti novo) vozilo;
<u>*početak* 4. *godine*</u>: zadržati dotad korišteno vozilo.

(Na početku 1. godine, prema pretpostavci podzadatka, kupljeno je novo vozilo, dok je početak 5. godine ujedno i kraj razdoblja planiranja, pa se dotad korišteno vozilo obavezno rashoduje.) Pripadni optimalni ukupni neto–troškovi jednaki su težini najkraćega puta i iznose 52 n.j.

b) Definiramo cjelobrojnu funkciju dvije cjelobrojne varijable $V : \mathbf{N}^2 \to \mathbf{N}$ s:

$V = V(k, i)$ = najmanji ukupni neto-troškovi od početka godine k do kraja razdoblja planiranja ako na početku godine k raspolažemo s vozilom starim i godina

Budući da, prema pretpostavci podzadatka, na početku prve godine raspolažemo s vozilom starim jednu godinu, želimo izračunati vrijednost $V(1, 1)$. Da bismo to mogli učiniti, trebamo definirati rekurzivnu relaciju za funkciju V. Razlikovat ćemo sljedeće alternative:
I. Na početku godine k raspolažemo s vozilom starosti i godina i donosimo odluku o zamjeni (tj. kupnji novoga) vozila

Tada u godini k najprije rashodujemo dotadašnje vozilo, pa time ostvarujemo prihod od s_i n.j. Potom kupujemo novo vozilo ostvarujući trošak od p n.j. i održavamo ga ostvarujući trošak od c_0 n.j. Na taj način dolazimo na početak sljedeće, $(k + 1)$ – ve godine kada raspolažemo s vozilom starim točno jednu godinu, pa su najmanji ukupni neto-troškovi od početka $(k + 1)$ – ve godine do kraja razdoblja $V(k + 1, 1)$. Ukupni neto-troškovi nastali izborom ove alternative iznose: $p + c_0 - s_i + V(k + 1, 1)$.

II. Na početku godine k raspolažemo s vozilom starosti i godina i donosimo odluku o zadržavanju dotad korištenoga vozila

Tada u godini k imamo jedino troškove održavanja vozila staroga i godina, a oni iznose c_i n.j. Na taj način dolazimo na početak sljedeće, $(k + 1)$ – ve godine kada raspolažemo s vozilom starim točno $i + 1$ godina. Najmanji ukupni neto-troškovi od početka $(k + 1)$ – ve godine do kraja razdoblja u ovom slučaju iznose $V(k + 1, i + 1)$. Ukupni neto-troškovi nastali izborom ove alternative iznose: $c_i + V(k + 1, i + 1)$.

Između ponuđenih alternativa odabrat ćemo onu koja ostvaruje najmanje ukupne neto-troškove, pa odatle slijedi:

$$V(k, i) = \min\{p + c_0 - s_i + V(k + 1, 1), c_i + V(k + 1, 1)\},$$

odnosno, nakon uvrštavanja $p = 20$ i $c_0 = 5$,

$$V(k, i) = \min\{25 - s_i + V(k + 1, 1), c_i + V(k + 1, 1)\}.$$

Preostaje nam zadati početne uvjete. Njih dobivamo uzimajući $k = 5$, tj. promatrajući što će se dogoditi po završetku razdoblja planiranja (početak pete godine). Tada obvezno trebamo rashodovati dotad korišteno vozilo. Ukoliko na početku 5. godine raspolažemo s vozilom starim i godina, onda mora vrijediti jednakost:

$$V(5, i) = -s_i$$

jer prihod ekonomski evidentiramo kao negativan trošak. Vrijednosti broja i mogu biti: 1 (ako smo početkom 4. godine kupili novo vozilo), 2 (ako smo početkom 3. godine kupili novo vozilo), 3 (ako smo početkom 2. godine kupili novo vozilo), 4 (ako smo početkom 1. godine kupili novo vozilo) i 5 (ako smo tijekom cijeloga razdoblja planiranja zadržali vozilo staro jednu godinu s kojim smo raspolagali na početku 1. godine). Prema tome je:

$$\begin{aligned}
V(5, 1) &= -s_1 = -11\\
V(5, 2) &= -s_2 = -7\\
V(5, 3) &= -s_3 = -5\\
V(5, 4) &= -s_4 = -4\\
V(5, 5) &= -s_5 = -2
\end{aligned}$$

U sljedećem koraku računamo vrijednosti funkcije $V(4, i)$. Vrijednosti broja i mogu biti: 1 (ako smo početkom 3. godine kupili novo vozilo), 2 (ako smo početkom 2. godine kupili novo vozilo), 3 (ako smo početkom 1. godine kupili novo vozilo) i 4 (ako smo tijekom prve tri godine zadržali vozilo staro jedinu godinu s kojim smo raspolagali na početku 1. godine). Prema tome je:

$$\begin{aligned}
V(4, 1) &= \min\{25 - s_1 + V(5, 1), c_1 + V(5, 2)\} = \min\{25 - 11 - 11, 8 - 7\} = \min\{3, 1\} = 1;\\
V(4, 2) &= \min\{25 - s_2 + V(5, 1), c_2 + V(5, 3)\} = \min\{25 - 7 - 11, 12 - 5\} = \min\{7, 7\} = 7;\\
V(4, 3) &= \min\{25 - s_3 + V(5, 1), c_3 + V(5, 4)\} = \min\{25 - 5 - 11, 14 - 4\} = \min\{9, 10\} = 9;\\
V(4, 4) &= \min\{25 - s_4 + V(5, 1), c_4 + V(5, 5)\} = \min\{25 - 4 - 11, 17 - 2\} = \min\{10, 15\} = 10.
\end{aligned}$$

U sljedećem koraku računamo vrijednosti funkcije $V(3, i)$. Vrijednosti broja i mogu biti: 1 (ako smo početkom 2. godine kupili novo vozilo), 2 (ako smo početkom 1. godine kupili novo

vozilo) i 3 (ako smo tijekom prve dvije godine zadržali vozilo staro jedinu godinu s kojim smo raspolagali na početku 1. godine). Prema tome je:

$V(3, 1) = \min\{25 - s_1 + V(4, 1), c_1 + V(4, 2)\} = \min\{25 - 11 + 1, 8 + 3\} = \min\{15, 11\} = 11;$
$V(3, 2) = \min\{25 - s_2 + V(4, 1), c_2 + V(4, 3)\} = \min\{25 - 7 + 1, 12 + 9\} = \min\{19, 21\} = 19;$
$V(3, 3) = \min\{25 - s_3 + V(4, 1), c_3 + V(4, 4)\} = \min\{25 - 5 + 1, 14 + 10\} = \min\{21, 24\} = 21.$

U sljedećem koraku računamo vrijednosti funkcije $V(2, i)$. Vrijednosti broja i mogu biti: 1 (ako smo početkom 1. godine kupili novo vozilo) i 2 (ako smo tijekom prve godine zadržali vozilo staro jedinu godinu s kojim smo raspolagali na početku te godine). Prema tome je:

$V(2, 1) = \min\{25 - s_1 + V(3, 1), c_1 + V(3, 2)\} = \min\{25 - 11 + 11, 8 + 19\} = \min\{25, 27\} = 25;$
$V(2, 2) = \min\{25 - s_2 + V(3, 1), c_2 + V(3, 3)\} = \min\{25 - 7 + 11, 12 + 21\} = \min\{29, 33\} = 29.$

Tako je konačno:

$V(1, 1) = \min\{25 - s_1 + V(2, 1), c_1 + V(2, 2)\} = \min\{25 - 11 + 25, 8 + 29\} = \min\{39, 37\} = 37.$

Tako vidimo da smo vrijednost $V(1, 1)$ dobili izborom alternative II i pomoću vrijednosti $V(2, 2)$, vrijednost $V(2, 2)$ izborom alternative I i pomoću vrijednosti $V(3, 1)$, vrijednost $V(3, 1)$ izborom alternative II i pomoću vrijednosti $V(4, 2)$, a vrijednost $V(4, 2)$ izborom bilo koje od dviju alternativa. Stoga imamo dvije različite optimalne politike:

1. *optimalna politika*:

početak 1. *godine*: zadržavanje dotad korištenoga vozila;
početak 2. *godine*: zamjena (kupnja novoga) vozila;
početak 3. *godine*: zadržavanje dotad korištenoga vozila;
početak 4. *godine*: zamjena (kupnja novoga) vozila;
početak 5. *godine*: rashodovanje dotad korištenoga vozila.

2. *optimalna politika*:

početak 1. *godine*: zadržavanje dotad korištenoga vozila;
početak 2. *godine*: zamjena (kupnja novoga) vozila;
početak 3. *godine*: zadržavanje dotad korištenoga vozila;
početak 4. *godine*: zadržavanje dotad korištenoga vozila;
početak 5. *godine*: rashodovanje dotad korištenoga vozila.

Pripadni optimalni ukupni neto-troškovi u obje varijante iznose 37 n.j.

4. Mreža dvosmjernih ulica nekoga područja sastoji se od ukupno 9 lokacija: A, B, C, D, E, F, G, H, I i J. Najkraća vremena prolaska (iskazana u minutama) ulicom između odgovarajućih lokacija su: $|CD| = |HI| = 1$, $|GH| = 3$, $|HJ| = 4$, $|BD| = 5$, $|AB| = |BJ| = 6$, $|IJ| = 7$, $|AC| = 8$, $|DE| = 9$, $|DF| = 10$, $|AD| = 12$, $|FI| = 13$, $|DI| = 14$, $|DG| = 15$ i $|DH| = 16$.

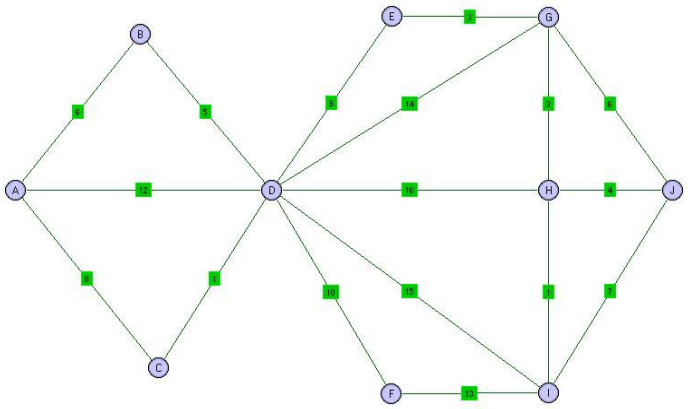
a) Redovita policijska ophodnja kreće iz policijske postaje smještene na lokaciji *A*, barem jednom prolazi svakom ulicom i vraća se na lokaciju *A*. Nađite rutu obilaska za koju je potrebno najmanje vremena i odredite pripadno optimalno vrijeme.

b) Može li policajac na putu iz policijske postaje prema svojoj kući smještenoj na lokaciji *J* točno jednom obići svaku pojedinu ulicu? Obrazložite svoj odgovor i navedite jednu rutu obilaska (ako postoji).

c) Ako hitna pomoć smještena na lokaciji *B* primi hitan poziv s lokacije *I*, odredite rutu kojom se vozilo treba kretati tako da u što kraćem vremenu stigne na cilj i odredite pripadno optimalno vrijeme.

d) Može li turistički autobus koji polazi s lokacije *E* i vozi zadanom mrežom ulica točno jednom obići svaku od preostalih lokacija i vratiti se na polazište? Ako može, navedite vremenski najkraću rutu kojom autobus treba voziti i izračunajte pripadno optimalno vrijeme. Ako ne može, odredite može li autobus svoj obilazak završiti na nekoj drugoj lokaciji i, ako može, vremenski najkraću rutu kojom autobus treba voziti od polazišta od odredišta, te vremenski najkraću rutu za povratak s odredišta na polazište.

Rješenje: Zadanu mrežu dvosmjernih ulica modeliramo npr. sljedećim neusmjerenim težinskim grafom:



a) Rješavamo problem kineskoga poštara s lokacijom *A* kao polazištem. Desnom tipkom miša kliknemo na vrh *A* i odaberemo opciju *Set/Unset as Start Vertex* (*Source*). Potom desnom tipkom miša kliknemo na bilo koju bjelinu unutar prozora, odaberemo opciju *Find* te podopciju *Chinese Postman Problem*. Tako (unutar kartice *View Data*) dobivamo sljedeći ispis:

```
Circuit's Total Cost - 160
Circuit's Total Length (Edges passed) - 22

Path:

A >> C >> D >> I >> J >> G >> J >> H >> I >> F >> D >> H >> G >> E >> G >>
D >> E >> D >> C >> A >> D >> B >> A
```

Prema tome, tražena optimalna ruta je:

$A – C – D – I – J – G – J – H – I – F – D – H – G – E – G – D – E – D – C – A – D – B – A$

Njezina ukupna težina je 160, što znači da pripadno optimalno vrijeme iznosi 160 minuta.

b) Tražimo postoji li Eulerova staza između vrhova *A* i *J*. Stupnjevi vrhova grafa su: $d_A = 3$, $d_B = = 2$, $d_C = 2$, $d_D = 8$, $d_E = 2$, $d_F = 2$, $d_G = 4$, $d_H = 4$, $d_I = 4$, $d_J = 3$. Budući da imamo točno dva vrha neparnoga stupnja (*A* i *J*), prema Poučku 1.6.2. postoji Eulerova staza u zadanom grafu i njezini krajnji vrhovi su upravo vrhovi neparnoga stupnja, tj. *A* i *J*. Dakle, na putu iz policijske postaje ka svojoj kući policajac može točno jednom obići svaku pojedinu ulicu. Traženu rutu dobivamo desnim klikom miša na bilo koju bjelinu unutar grafa, odabirom opcije *Find* i podopcije *Eulerian Path/Circuit*:

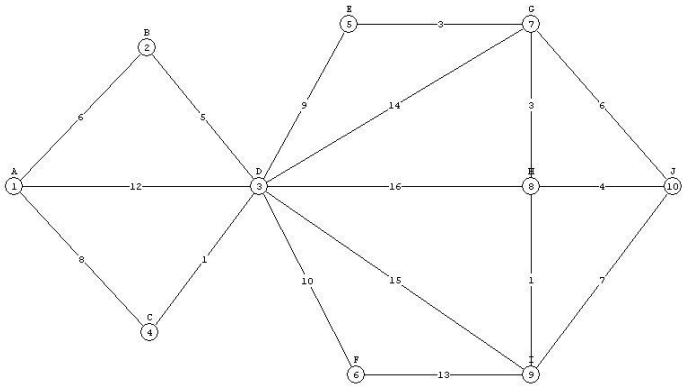
$$A – B – D – A – C – D – E – G – D – F – I – D – H – G – J – H – I – J.$$

c) Tražimo najkraći put između vrhova *B* i *I*. Analogno kao u rješenju zadatka 3. a) dobivamo traženu rutu:

$$B – D – I$$

čija je težina 20. Stoga pripadno optimalno vrijeme iznosi 20 minuta.

d) Zadani ćemo graf najprije modelirati koristeći računalni program *Grin*:



Sada rješavamo problem trgovačkoga putnika s lokacijom *E* kao polazištem. Lijevom tipkom miša kliknimo na izbornik *Property*, pa odaberimo podizbornik *NetWork* i opciju *Salesman Problem*. Dobivamo informacijski okvir *Select Source Point* unutar kojega lijevom tipkom miša kliknemo na *OK*, a potom istom tipkom miša kliknemo na vrh 5 (tj. *E*). Tako dobivamo izvještaj:

Salesman Cycle not found

Prema tome, autobus ne može točno jednom obići svaku pojedinu lokaciju i vratiti se na polazište, već barem jednu lokaciju mora obići barem dva puta. Da bismo utvrdili može li autobus završiti obilazak na nekoj drugoj lokaciji, odredimo postoji li Hamiltonov put u zadanom grafu.

Zatvorimo ranije dobiveni prozor, pa lijevom tipkom miša kliknimo na izbornik *Property*, odaberimo podizbornik *Graph* i opciju *Hamiltonian Path*. Nakon što ponovno označimo vrh 5

(tj. E) kao polazište, dobivamo izvještaj s ukupno 4 različita Hamiltonova puta. Od svih njih najmanju težinu 52 ima prvi, odnosno put $E - G - J - H - I - F - D - C - A - B$. Stoga turistički autobus može završiti obilazak na lokaciji B, a optimalno vrijeme potrebno za taj obilazak je 52 minute. Optimalna ruta povratka s odredišta B na polazište E jednaka je najkraćem putu iz vrha B u vrh E, a pripadno optimalno vrijeme jednako je duljini toga najkraćega puta. Analogno kao u rješenju podzadatka **c)** dobivamo da je tražena ruta:

$$B - D - E$$

koja ima težinu 14, pa optimalno vrijeme za povratak s odredišta B na polazište E iznosi 14 minuta.

2.2. **Primjer 2.**

1. Na raspolaganju nam je točno 4 komada namirnice N_1 , po 5 komada svake od namirnica N_2 i N_3, točno 6 komada namirnice N_4, te ruksak najvećega dozvoljenoga kapaciteta 20 kg. Podaci o masi i neto–cijeni jednoga komada svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*neto–cijena* [kn]	*masa* [kg]
N_1	36.00	2
N_2	34.00	2.5
N_3	33.00	1.5
N_4	35.00	2

Niti jedan komad bilo koje namirnice nije dozvoljeno razdijeliti na manje dijelove. U ruksak treba staviti barem 2 komada svake pojedine vrste namirnica tako da njihova ukupna masa ne bude veća od kapaciteta ruksaka i da njihova ukupna vrijednost bude što veća. Formirajte odgovarajući matematički model, pa odredite optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.

***Rješenje*:** Za svaki $i \in [4]$ označimo s x_i ukupan broj komada namirnica N_i koje treba staviti u ruksak. Tada je matematički model promatranoga problema

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 36.00 \cdot x_1 + 34.00 \cdot x_2 + 33.00 \cdot x_3 + 35.00 \cdot x_4$$

pod uvjetima

$$2 \cdot x_1 + 2.5 \cdot x_2 + 1.5 \cdot x_3 + 2 \cdot x_4 \leq 20$$

$$2 \leq x_1 \leq 4,\ 2 \leq x_2 \leq 5,\ 2 \leq x_3 \leq 5,\ 2 \leq x_4 \leq 6,$$

$$x_1, x_2, x_3, x_4 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 4,\ x_2 = x_3 = x_4 = 2,\ f_{max} = 348 \text{ kn}.$$

Dakle, optimalan izbor namirnica je: 4 komada namirnice N_1, te po 2 komada svake od namirnica N_2, N_3 i N_4. Njihova optimalna ukupna vrijednost je f_{max} = 348,00 kn.

2. Neka tvrtka istodobno treba montirati tri tipa objekata: A_1, A_2 i A_3. Pri montaži svih tipova objekata koristi se posebna oprema čiji je ukupni kapacitet 25 komada. Osnovni podaci o broju potrebnih komada posebne opreme i dobiti za izradu svakoga pojedinoga tipa opreme navedeni su u sljedećoj tablici.

tip objekta	*broj potrebnih komada opreme*	*dobit* [n.j./objekt]
A_1	5	8
A_2	7	12
A_3	6	10

Odlukom uprave tvrtke istodobno se mora montirati barem jedan objekt svakoga tipa. Tvrtka treba napraviti plan montaže tako da ukupna dobit ostvarena montažom svih triju tipova objekata bude maksimalna. Formirajte odgovarajući matematički model, pa odredite optimalan plan montaže i pripadnu optimalnu ukupnu dobit.

Rješenje: Za svaki $i \in [3]$ označimo s x_i broj objekata tipa A_i koje treba montirati. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 8 \cdot x_1 + 12 \cdot x_2 + 10 \cdot x_3$$

pod uvjetima

$$5 \cdot x_1 + 7 \cdot x_2 + 6 \cdot x_3 \leq 25$$

$$x_1, x_2, x_3 \geq 1,$$

$$x_1, x_2, x_3 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = x_3 = 1, x_2 = 2, f_{max} = 42.$$

Prema tome, optimalan plan montaže je: montirati po jedan objekt tipa A_1, odnosno A_3, te dva objekta tipa A_2. Pripadna optimalna ukupna dobit iznosi 42 n.j.

3. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u razdoblju planiranja od $n = 4$ godine. Stoga je na početku razdoblja planiranja kupljeno novo vozilo po cijeni od $p = 30$ n.j. Godišnji neto-trošak održavanja i likvidacijska vrijednost vozila iskazani su tablično kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4	5
godišnji neto–trošak (c_i) [n.j.]	9	14	20	27	30	-
likvidacijska vrijednost (s_i) [n.j.]	-	16	9	6	3	0

Početkom svake od sljedećih godina razdoblja planiranja donosi se odluka o zadržavanju ili zamjeni dotad korištenoga vozila tako da ukupni troškovi u cijelom razdoblju planiranja budu što manji. Pritom se pretpostavlja da je nabavna cijena vozila stalna tijekom cijeloga razdoblja planiranja. Formirajte odgovarajući matematički model, pa odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja i pripadne optimalne ukupne neto-troškove.

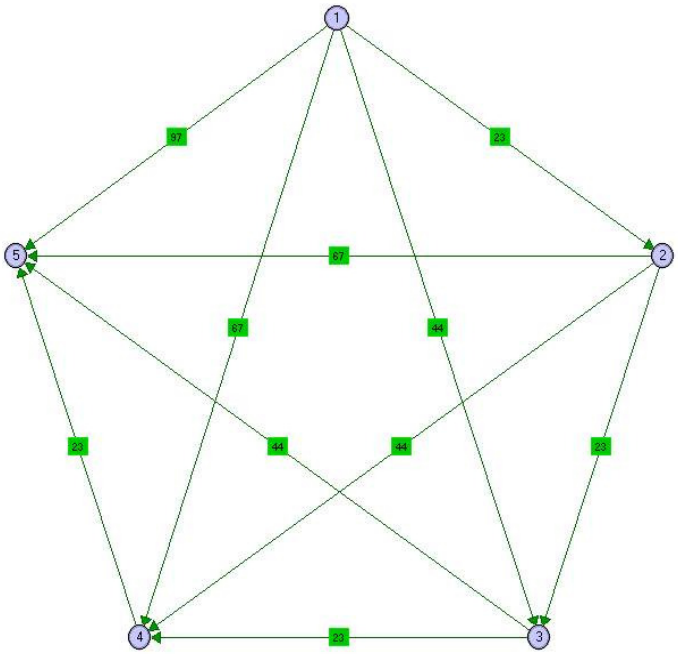
Rješenje: Matematički model je potpuni usmjereni težinski graf K_5 čiji su vrhovi početci godina, tj. vrh i označava početak godine i. Za svaki uređeni par $(i, j) \in [5]^2$ takav da je $i < j$ vrhove i i j spajamo lukom (i, j) čija je težina w_{ij} definirana izrazom:

w_{ij} = (nabavna cijena vozila na početku godine i) + (troškovi održavanja vozila u godinama i, $i + 1, \ldots, j - 1$) – (likvidacijska vrijednost vozila na početku godine j) = $p + \sum_{k=0}^{j-i} c_k - s_{j-i}$.

Zbog toga je:

$w_{12} = w_{23} = w_{34} = w_{45} = p + c_0 - s_1 = 30 + 9 - 16 = 23;$
$w_{13} = w_{24} = w_{35} = p + (c_0 + c_1) - s_2 = 30 + 9 + 14 - 9 = 44;$
$w_{14} = w_{25} = p + (c_0 + c_1 + c_2) - s_3 = 30 + 9 + 14 + 20 - 6 = 67;$
$w_{15} = p + (c_0 + c_1 + c_2 + c_3) - s_4 = 30 + 9 + 14 + 20 + 27 - 3 = 97;$

Korištenjem programa *Graph Magics* dobivamo sljedeći graf:



u kojemu tražimo najkraći put između vrhova 1 i 5. Primjenom procedure *Shortest Path* (*from Start Vertex to End*) dobiva se najkraći put 1 – 3 – 5 čija je ukupna težina 88. Stoga je tražena optimalna politika:

<u>*početak* 2. *i* 4. *godine*</u>: zadržavanje dotadašnjega vozila;
<u>*početak* 3. *godine*</u>: zamjena (kupnja novoga) vozila;

početak 5. godine: rashodovanje vozila.

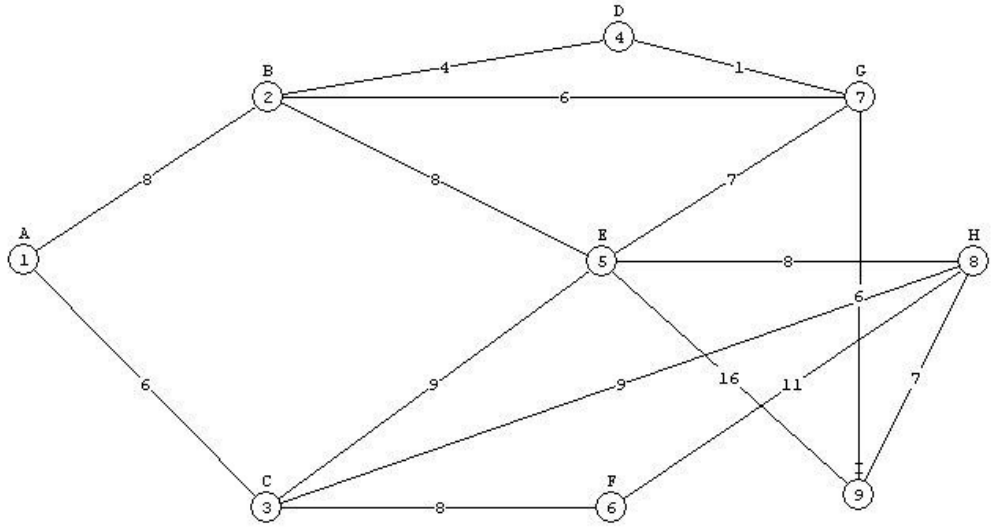
Pripadni optimalni ukupni neto-troškovi iznose 88 n.j.

4. Mreža dvosmjernih ulica nekoga područja sastoji se od 9 lokacija: *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H* i *I*. Najkraća vremena prolaska (iskazana u minutama) ulicom između odgovarajućih lokacija su: $|DG| = 1$, $|BD| = 4$, $|AC| = |BG| = |GI| = 6$, $|EG| = |HI| = 7$, $|AB| = |BE| = |CF| = |EH| = 8$, $|CE| = |CH| = 9$, $|FH| = 11$ i $|EI| = 16$.

a) Turistički autobus polazi s lokacije *A* i, vozeći zadanom mrežom ulica, točno jednom obilazi svaku od preostalih 8 lokacija. Odredite rutu obilaska za koju je potrebno najmanje vremena i izračunajte pripadno optimalno vrijeme.

b) Dokažite da se među svim zadanim lokacijama mogu odabrati dvije međusobno različite lokacije *X* i *Y* za koje postoji ruta od *X* do *Y* koja svaku od zadanih ulica obilazi točno jednom. Za odabrane lokacije odredite traženu rutu. Sve svoje tvrdnje detaljno obrazložite.

Rješenje: Matematički model je neusmjereni težinski graf s ukupno 9 vrhova koji označavaju lokacije, te 15 bridova čije težine su najkraća vremena prolaska ulicom određenom odgovarajućim lokacijama. Jedan od mogućih modela dobiven programom *Grin* je:



a) Tražimo rješenje problema trgovačkoga putnika. Primjenom procedure *Salesman problem* dobivamo da je jedna moguća tražena ruta 1 – 3 – 6 – 8 – 9 – 5 – 7 – 4 – 2 – 1, odnosno *A* – *C* – *F* – *H* – *I* – *E* – *G* – *D* – *B* – *A*. Njezina težina je 68, što znači da je optimalno (najkraće) vrijeme obilaska svih lokacija 68 minuta.

b) Vrhovi neparnoga stupnja u zadanom grafu su *E* i *I*, pa iz Poučaka 2. i 3. u točki 1.6. slijedi da postoji Eulerova staza, ali ne i Eulerov ciklus, a odavde izravno slijedi prva tvrdnja. Primjenom procedure *Eulerian Path* dobivamo jedno moguće rješenje: $X = I$, $Y = E$, odnosno traženu rutu: *I* – *H* – *F* – *C* – *E* – *G* – *I* – *E* – *H* – *C* – *A* – *B* – *D* – *G* – *B* – *E*.

2.3. **Primjer 3.**

1. Na raspolaganju nam je točno 5 komada namirnice N_1 , po 4 komada svake od namirnica N_2 i N_3, točno 6 komada namirnice N_4, te ruksak najvećega dozvoljenoga kapaciteta 25 kg. Podaci o masi i neto–cijeni jednoga komada svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*neto–cijena* [kn]	*masa* [kg]
N_1	36.00	2
N_2	34.00	2.5
N_3	33.00	1.5
N_4	35.00	2

Niti jedan komad bilo koje namirnice nije dozvoljeno razdijeliti na manje dijelove. U ruksak treba staviti barem 2 komada svake pojedine vrste namirnica tako da njihova ukupna masa ne bude veća od kapaciteta ruksaka i da njihova ukupna vrijednost bude što veća. Formirajte odgovarajući matematički model, pa odredite optimalan izbor namirnica i pripadnu optimalnu ukupnu vrijednost.

***Rješenje*:** Za svaki i = 1, 2, 3, 4 označimo s x_i ukupan broj komada namirnica N_i koje treba staviti u ruksak. Tada je matematički model promatranoga problema

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 36.00 \cdot x_1 + 34.00 \cdot x_2 + 33.00 \cdot x_3 + 35.00 \cdot x_4$$

pod uvjetima

$$2 \cdot x_1 + 2.5 \cdot x_2 + 1.5 \cdot x_3 + 2 \cdot x_4 \leq 25$$

$$2 \leq x_1 \leq 5,\ 2 \leq x_2 \leq 4,\ 2 \leq x_3 \leq 4,\ 2 \leq x_4 \leq 6,$$

$$x_1, x_2, x_3, x_4 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 5,\ x_2 = x_4 = 2,\ x_3 = 4,\ \ f_{max} = 450{,}00 \text{ kn}.$$

Dakle, optimalan izbor namirnica je: 5 komada namirnice N_1, po 2 komada svake od namirnica N_2 i N_4, te 4 komada namirnice N_3. Njihova optimalna ukupna vrijednost je f_{max} = 450,00 kn.

2. Zrakoplovom nosivosti 83 tone na nekoj se liniji mogu prevoziti putnici, poštanske pošiljke i ostali teret. Prosječna masa i neto–dobit ostvarena po jedinici transporta navedeni su u donjoj tablici.

	1 *putnik*	1 *poštanski paket*	1 *paket ostaloga tereta*
prosječna masa [t]	0.08	0.75	0.85
neto – dobit [000 n.j.]	1.8	1.7	1.9

Zrakoplovom je odjednom moguće prevesti najviše 50 putnika. Treba odrediti strukturu transporta tako da ukupna neto–dobit od jednoga transporta bude što veća.

a) Formirajte odgovarajući matematički model, pa odredite optimalnu strukturu transporta, pripadnu optimalnu ukupnu neto–dobit i postotak iskorištenosti nosivosti zrakoplova.

b) Za koliko će se postotaka promijeniti optimalna neto-dobit iz **a)** podzadatka ako se uvede dodatni uvjet da se u jednom transportu ne može prevesti više od 100 poštanskih paketa?

Rješenje: **a)** Označimo s x_1 broj putnika, s x_2 broj poštanskih paketa, te s x_3 broj paketa ostaloga tereta koje treba prevesti. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 1.8 \cdot x_1 + 1.7 \cdot x_2 + 1.9 \cdot x_3$$

pod uvjetima

$$0.08 \cdot x_1 + 0.75 \cdot x_2 + 0.85 \cdot x_3 \leq 83$$

$$0 \leq x_1 \leq 50,$$

$$x_1, x_2, x_3 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 50,\ x_2 = 103,\ x_3 = 2,\ f_{max} = 268{,}90.$$

Dakle, optimalna struktura transporta je: 50 putnika, 103 poštanska paketa i 2 paketa ostaloga tereta. Pripadna optimalna ukupna neto-dobit iznosi 268 900 n.j., dok je postotak iskorištenosti nosivosti zrakoplova

$$p = \frac{0.08 \cdot 50 + 0.75 \cdot 103 + 0.85 \cdot 2}{83} \cdot 100 = 99{,}94\ (\%).$$

b) U ovome se slučaju, uz dodatni uvjet $0 \leq x_2 \leq 100$, dobiva $x_1 = 50$, $x_2 = 95$, $x_3 = 9$ i $f_{max} = 268{,}60$, pa će se optimalna ukupna neto–dobit smanjiti za

$$p_1 = \frac{268{,}90 - 268{,}60}{268{,}90} \cdot 100 = 0{,}11\%.$$

3. Za opremu nekoga proizvodnoga pogona koja osigurava neprekidan proces proizvodnje zadani su nabavna cijena $p = 100$ n.j. i funkcija neto–dobiti nastale korištenjem opreme (d). Nakon određenoga vremena korištenja opremu nije moguće prodati, pa početkom svake godine treba donijeti odluku o zamjeni ili zadržavanju dotadašnje opreme. Godišnja neto–dobit od korištenja opreme zadana je kao funkcija starosti opreme (i):

$$d_i = d(i) = \begin{cases} 100 - 25 \cdot i, & \text{za } i \in [4] \\ 0, & \text{za } i \geq 5 \end{cases}$$

Pretpostavljamo da je nabavna cijena opreme stalna tijekom cijeloga razdoblja planiranja. Odredite optimalnu politiku zamjene opreme u razdoblju od $n = 6$ godina i pripadnu optimalnu ukupnu neto–dobit ako je na početku prve godine kupljena nova oprema.

Rješenje: Matematički model je potpun usmjeren težinski graf K_7 takav da vrh i označava početak godine i. Za svaki uređeni par $(i, j) \in [7]^2$ takav da je $i < j$ vrhove i i j spajamo lukom (i, j) čija je težina w_{ij} definirana izrazom:

w_{ij} = (ukupna neto–dobit nastala korištenjem opreme u godinama $i, i + 1, \ldots, j - 1, j - 1$) – (nabavna cijena opreme u godini i) = $\sum_{k=0}^{j-i-1} d_k - p$.

Zbog $d_0 = p = 100$ za svaki $i \in [6]$ vrijede jednakosti:

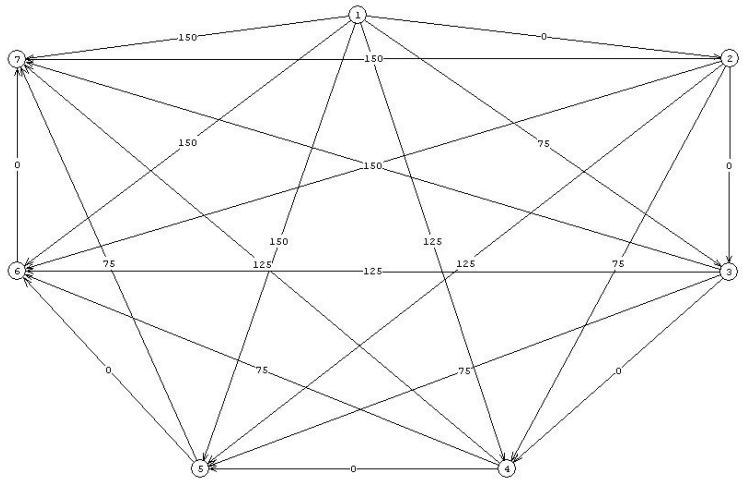
$$w_{i,\, i+1} = 0,$$

$$w_{i,\, i+k} = \sum_{m=1}^{k-1} d_k \text{ , za svaki } k \in \mathbf{N} \setminus \{1\} \text{ za koji postoji luk } (i, i + k),$$

pa zbog $d_1 = 100 - 25 \cdot 1 = 75$, $d_2 = 100 - 25 \cdot 2 = 50$, $d_3 = 100 - 25 \cdot 3 = 25$, $d_4 = 100 - 25 \cdot 4 = 0$ i $d_5 = 0$ slijedi:

$w_{12} = w_{23} = w_{34} = w_{45} = w_{56} = w_{67} = w_{78} = w_{89} = 0$;
$w_{13} = w_{24} = w_{35} = w_{46} = w_{57} = w_{68} = w_{79} = d_1 = 75$;
$w_{14} = w_{25} = w_{36} = w_{47} = w_{58} = w_{69} = d_1 + d_2 = 75 + 50 = 125$;
$w_{15} = w_{26} = w_{37} = w_{48} = w_{59} = d_1 + d_2 + d_3 = 75 + 50 + 25 = 150$;
$w_{16} = w_{27} = w_{38} = w_{49} = d_1 + d_2 + d_3 + d_4 = 75 + 50 + 25 + 0 = 150$;
$w_{17} = w_{28} = w_{39} = d_1 + d_2 + d_3 + d_4 + d_5 = 75 + 50 + 25 + 0 + 0 = 150$;

Koristeći program *Grin* dobivamo traženi matematički model:



Primjenom procedure *Critical Path* dobiva se kritični put 1 – 4 – 7 čija je težina 250. Prema tome, optimalna politika je:

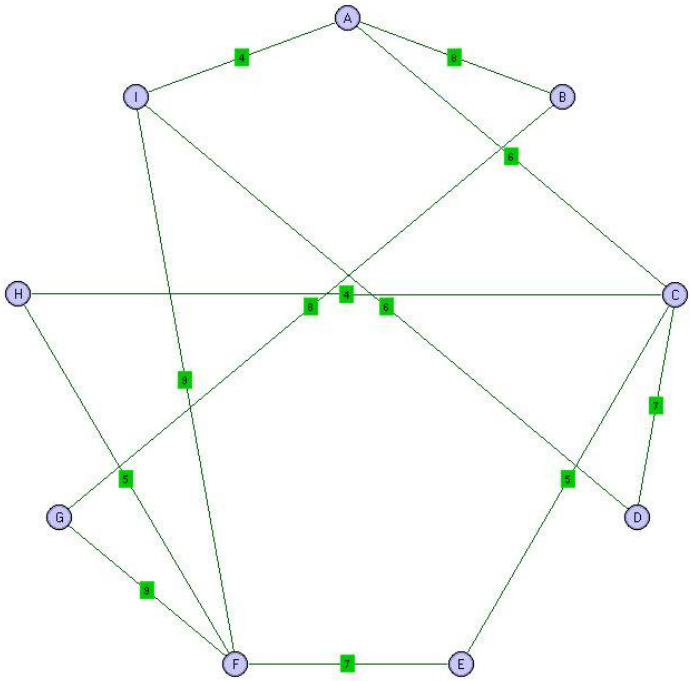
početak 2., 3., 5. *i* 6. *godine*: zadržavanje dotadašnje opreme;
početak 4. *godine*: zamjena (kupnja nove) opreme.

Pripadna optimalna ukupna neto-dobit iznosi 250 n.j.

4. Mreža dvosmjernih ulica nekoga područja sastoji se od 9 lokacija: A, B, C, D, E, F, G, H i I. Najkraća vremena prolaska (iskazana u minutama) ulicom između odgovarajućih lokacija su: $|AI| = |CH| = 4$, $|CE| = |FH| = 5$, $|AC| = |DI| = 6$, $|CD| = |EF| = 7$, $|AB| = |BG| = 8$, $|FG| = |FI| = 9$.

a) Policijska postaja smještena je na lokaciji F. Može li policijska patrola polazeći s te lokacije točno jednom obići svaku od zadanih ulica i vratiti se u postaju? Ako može, navedite pripadnu rutu obilaska. Ako ne može, objasnite svoj odgovor, pa odredite rutu obilaska kojom patrola u što kraćem vremenu barem jednom može obići svaku od zadanih ulica i izračunajte pripadno optimalno vrijeme. Sve svoje tvrdnje detaljno obrazložite.
b) Postoje li lokacije X i Y tako da turistički autobus polazeći s lokacije X i vozeći zadanom mrežom ulica točno jednom obiđe svaku od zadanih lokacija i završi svoj obilazak na lokaciji Y? Ako postoje, odredite ih i navedite odgovarajuću rutu obilaska. Ako ne postoje, objasnite svoj odgovor.

Rješenje: Pripadni matematički model je neusmjereni težinski graf s ukupno 9 vrhova koji označavaju lokacije i 12 bridova čije težine su najkraća vremena prolaska ulicom određenom odgovarajućim lokacijama. Jedan od mogućih modela dobiven programom *Graph Magics* je:



a) Tražimo postoji li Eulerov ciklus u zadanom grafu. Vrh I ima stupanj 3, pa prema Poučku 1.6. traženi ciklus ne postoji. Stoga ćemo rutu koja barem jednom obilazi svaku ulicu odrediti kao rješenje problema kineskoga poštara. Primjenom procedure *Chinese Postman Problem* dobivamo: $F – H – C – D – I – A – C – E – F – G – B – A – I – F$ čija je težina 82. Stoga je optimalno vrijeme željenoga obilaska 82 minute.

b) Tražimo postoji li barem jedan Hamiltonov put u zadanom grafu. Primjenom procedure *Hamiltonian Path/Circuit* dobiva se $X = D$, $Y = H$ (ili obrnuto) odnosno tražena ruta: $D - I - A - B - G - F - E - C - H$ (ili obrnuto).

2.4. **Primjer 4.**

1. Na raspolaganju nam je točno 6 komada namirnice N_1 , po 5 komada svake od namirnica N_2 i N_3, točno 4 komada namirnice N_4, te ruksak najvećega dozvoljenoga kapaciteta 20 kg. Podaci o masi i neto–cijeni jednoga komada svake pojedine namirnice navedeni su u sljedećoj tablici.

namirnica	*neto–cijena* [kn]	*masa* [kg]
N_1	36.00	2
N_2	34.00	2.5
N_3	33.00	1.5
N_4	35.00	2

Niti jedan komad bilo koje namirnice nije dozvoljeno razdijeliti na manje dijelove. U ruksak treba staviti barem 1 komad svake pojedine vrste namirnica tako da njihova ukupna masa ne bude veća od kapaciteta ruksaka i da njihova ukupna vrijednost bude što veća. Formirajte odgovarajući matematički model, pa odredite optimalan izbor namirnica i njihovu optimalnu ukupnu vrijednost.

Rješenje: Za svaki $i \in [4]$ označimo s x_i ukupan broj komada namirnica N_i koje treba staviti u ruksak. Tada je matematički model promatranoga problema

$$\text{maksimizirati } f(x_1, x_2, x_3, x_4) = 36.00 \cdot x_1 + 34.00 \cdot x_2 + 33.00 \cdot x_3 + 35.00 \cdot x_4$$

pod uvjetima

$$2 \cdot x_1 + 2.5 \cdot x_2 + 1.5 \cdot x_3 + 2 \cdot x_4 \leq 20$$

$$1 \leq x_1 \leq 6,\ 1 \leq x_2 \leq 5,\ 1 \leq x_3 \leq 5,\ 1 \leq x_4 \leq 4,$$

$$x_1, x_2, x_3, x_4 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 4,\ x_2 = x_4 = 1,\ x_3 = 5,\ f_{max} = 378{,}00 \text{ kn}.$$

Dakle, optimalan izbor namirnica je: 4 komada namirnice N_1, po 1 komad svake od namirnica N_2 i N_4, te 5 komada namirnice N_3. Njihova optimalna ukupna vrijednost je $f_{max} = 378{,}00$ kn.

2. Zrakoplovom nosivosti 85 tona na nekoj se liniji mogu prevoziti putnici, poštanske pošiljke i ostali teret. Prosječna masa i neto–dobit ostvarena po jedinici transporta navedeni su u donjoj tablici.

	1 *putnik*	1 *poštanski paket*	1 *paket ostaloga tereta*
prosječna masa [t]	0.08	0.80	0.85
neto – dobit [000 n.j.]	2.9	2.8	3

Zrakoplovom je odjednom moguće prevesti najviše 60 putnika. Treba odrediti strukturu transporta tako da ukupna neto–dobit jednoga transporta bude što veća.

a) Formirajte odgovarajući matematički model, pa odredite optimalnu strukturu transporta, pripadnu optimalnu ukupnu neto–dobit i postotak iskorištenosti nosivosti zrakoplova.

b) Za koliko će se postotaka promijeniti optimalna neto-dobit iz **a)** podzadatka ako se uvede dodatni uvjet da se u jednom transportu mora prevesti točno 20 poštanskih paketa?

Rješenje: **a)** Označimo s x_1 broj putnika, s x_2 broj poštanskih paketa, te s x_3 broj paketa ostaloga tereta koje treba prevesti. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 2.9 \cdot x_1 + 2.8 \cdot x_2 + 3 \cdot x_3$$

pod uvjetima

$$0.08 \cdot x_1 + 0.80 \cdot x_2 + 0.85 \cdot x_3 \leq 85$$

$$0 \leq x_1 \leq 60,$$

$$x_1, x_2, x_3 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 60,\ x_2 = 12,\ x_3 = 83\ ,\ f_{max} = 456{,}60.$$

Dakle, optimalna struktura transporta je: 60 putnika, 12 poštanskih paketa i 83 paketa ostaloga tereta. Pripadna optimalna ukupna neto-dobit iznosi 456 600 n.j, a postotak iskorištenosti nosivosti zrakoplova

$$p = \frac{0.08 \cdot 60 + 0.80 \cdot 12 + 0.85 \cdot 83}{85} \cdot 100 = 99{,}94\ (\%).$$

b) U ovome se slučaju, uz dodatan uvjet $x_2 = 20$, dobiva $x_1 = 60$, $x_2 = 20$, $x_3 = 75$ i $f_{max} = 455$, pa će se optimalna ukupna neto–dobit smanjiti za

$$p_1 = \frac{456{,}60 - 455}{456{,}60} \cdot 100 = 0{,}35\%.$$

3. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u razdoblju planiranja od $n = 6$ godina. Stoga je na početku prve godine razdoblja planiranja kupljeno novo vozilo po cijeni od $p = 100$ n.j. i ta je cijena stalna tijekom cijeloga razdoblja planiranja. Godišnji neto-trošak održavanja i likvidacijska vrijednost vozila iskazani su tablično kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3
godišnji neto–trošak (c_i) [n.j.]	20	40	70	–
likvidacijska vrijednost (s_i) [n.j.]	-	50	25	10

Svako pojedino vozilo se može koristiti najviše tri godine, a potom ga obvezno treba rashodovati. Zbog toga se početkom svake od sljedećih godina razdoblja planiranja donosi odluka o zamjeni ili zadržavanju dotadašnjega vozila tako da ukupni neto-troškovi u cjelokupnom razdoblju planiranja budu što manji. Formirajte odgovarajući matematički model, pa odredite optimalnu politiku zamjene vozila u cjelokupnom razdoblju planiranja i pripadne optimalne ukupne neto-troškove.

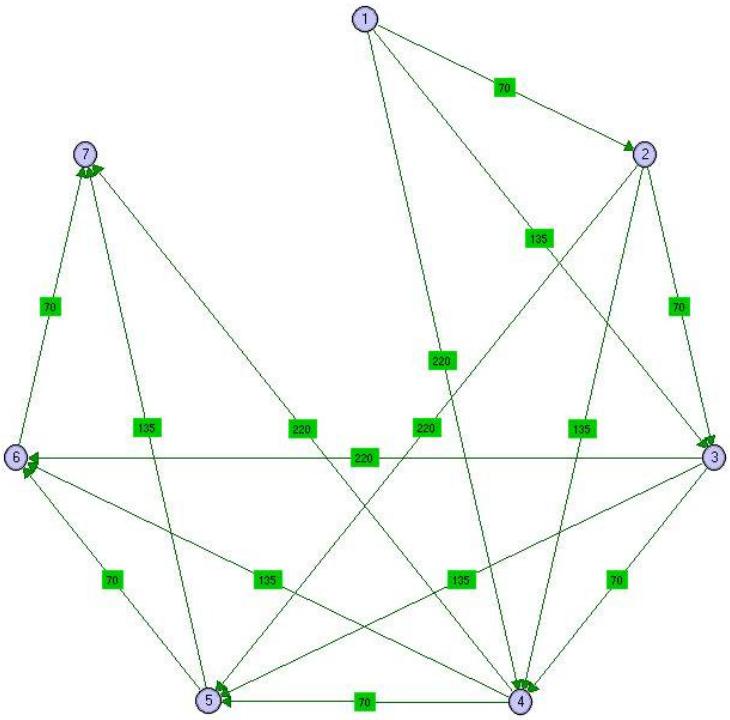
Rješenje: Matematički model je usmjereni težinski graf s ukupno 7 vrhova takvih da vrh i označava početak godine i. Za svaki uređeni par $(i, j) \in [7]^2$ takav da je $j - i \leq 3$ vrhove i i j spajamo lukom (i, j) čija je težina w_{ij} definirana izrazom:

w_{ij} = (nabavna cijena vozila na početku godine i) + (troškovi održavanja vozila u godinama i, $i + 1, \ldots, j - 1$) – (likvidacijska vrijednost vozila na početku godine j) = $p + \sum_{k=0}^{j-i} c_k - s_{j-i}$.

Zbog toga je:

$w_{12} = w_{23} = w_{34} = w_{45} = w_{56} = w_{67} = p + c_0 - s_1 = 100 + 20 - 50 = 70;$
$w_{13} = w_{24} = w_{35} = w_{46} = w_{57} = p + (c_0 + c_1) - s_2 = 100 + 20 + 40 - 25 = 135;$
$w_{14} = w_{25} = w_{36} = w_{47} = p + (c_0 + c_1 + c_2) - s_3 = 100 + 20 + 40 + 70 - 10 = 220.$

Korištenjem programa *Graph Magics* dobivamo sljedeći usmjereni graf



u kojemu tražimo najkraći put između vrhova 1 i 7. Primjenom procedure *Shortest Path* (*from Start Vertex to End*) dobivamo navedeni put: 1 – 3 – 5 – 7 čija je težina 405. Stoga je tražena optimalna politika zamjene:

početak 2., 4. *i* 6. *godine*: zadržavanje dotadašnjega vozila;
početak 3. *i* 5. *godine*: zamjena (kupnja novoga) vozila.

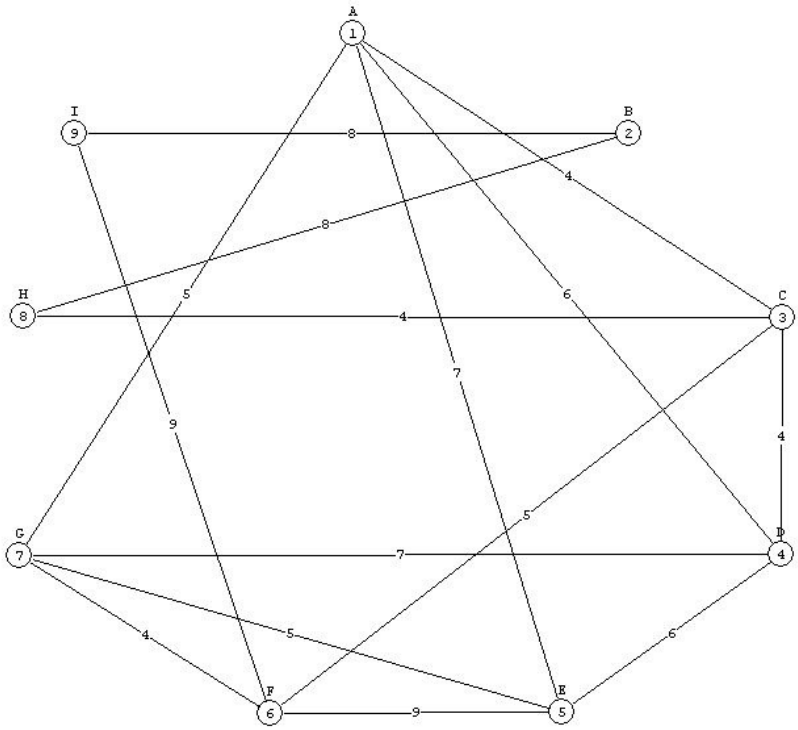
Pripadni optimalni ukupni neto-troškovi iznose 405. n.j.

4. Mreža dvosmjernih ulica nekoga područja sastoji se od 9 lokacija: *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H* i *I*. Najkraća vremena prolaska (iskazana u minutama) ulicom između odgovarajućih lokacija su: $|AC| = |CD| = |CH| = |FG| = 4$, $|AG| = |EG| = |CF| = 5$, $|AD| = |DE| = 6$, $|AE| = |DG| = 7$, $|BH| = |BI| = 8$, $|EF| = |FI| = 9$.

a) Policijska postaja smještena je na lokaciji *A*. Može li policijska patrola polazeći s te lokacije točno jednom obići svaku od zadanih ulica i vratiti se u postaju? Ako može, navedite pripadnu rutu obilaska. Ako ne može, objasnite svoj odgovor, pa odredite rutu obilaska kojom patrola u što kraćem vremenu barem jednom može obići svaku od zadanih ulica i izračunajte pripdno optimalno vrijeme. Sve svoje tvrdnje detaljno obrazložite.

b) Turistički autobus polazi s lokacije *D* i, vozeći zadanom mrežom ulica, barem jednom obilazi svaku od preostalih lokacija i vraća se na polazište. Odredite rutu obilaska za koju je potrebno najmanje vremena i izračunajte pripadno optimalno vrijeme.

Rješenje: Pripadni matematički model je neusmjereni težinski graf s ukupno 9 vrhova koji označavaju lokacije i 15 bridova čije težine su najkraća vremena prolaska ulicom određenom odgovarajućim lokacijama. Jedan od mogućih modela dobiven programom *Grin* je:



a) U dobivenom grafu svi vrhovi imaju paran stupanj, pa primjenom Poučka 1.3. zaključujemo da postoji Eulerov ciklus. Stoga policijska patrola polazeći s lokacije *A* može točno jednom obići svaku ulicu i vratiti se u policijsku postaju. Jedna moguća ruta dobivena procedurom *Eulerian Cycle* je: *A – C – D – E – G – F – I – B – H – C – F – E – A – G – D – A*.

b) Rješavamo problem trgovačkoga putnika. Primjenom procedure *Salesman Problem* dobivamo jednu moguću rutu: *D – E – G – F – I – B – H – C – A – D* čija je težina 54. Stoga pripadno optimalno vrijeme iznosi 54 minute.

2.5. **Primjer 5.**

1. Iznos od 30 n.j. treba rasporediti na tri industrijske grane G_1, G_2 i G_3 tako da ukupna ostvarena neto-dobit bude što veća i da iznos uložen u svaku pojedinu granu bude cjelobrojan. U granu G_1 moguće je uložiti najviše 10 n.j., a za x uloženih novčanih jedinica ostvaruje se neto-dobit $0.4 \cdot x^2$ n.j. U granu G_2 moguće je uložiti najviše 8 n.j., a za y uloženih novčanih jedinica ostvaruje se neto-dobit $10 \cdot y$ n.j. U granu G_3 moguće je uložiti najviše 15 n.j., a za z uloženih novčanih jedinica ostvaruje se neto–dobit $12 \cdot z$ n.j. Formirajte odgovarajući matematički model, pa odredite optimalnu razdiobu ukupnoga iznosa i izračunajte pripadnu optimalnu ukupnu ostvarenu neto-dobit.

***Rješenje*:** Za svaki $i \in [3]$ neka je x_i iznos uložen u granu G_i. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 0.4 \cdot x_1^2 + 10 \cdot x_2 + 12 \cdot x_3$$

pod uvjetima

$$x_1 + x_2 + x_3 = 30$$
$$0 \le x_1 \le 10,\ 0 \le x_2 \le 8,\ 0 \le x_3 \le 15,$$
$$x_1, x_2, x_3 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 7,\ x_2 = 8,\ x_3 = 15,\ f_{max} = 279{,}60.$$

Dakle, optimalan plan ulaganja je: u granu G_1 treba uložiti 7 n.j., u granu G_2 8 n.j., a u granu G_3 15 n.j. Pripadna optimalna ukupna neto-dobit iznosi 279,60 n.j.

2. Tvrtka "*Gulikožić* d.o.o." raspolaže s kamionom nosivosti 45 jed. Tim kamionom treba prevesti dvije vrste tereta: *A* i *B*, pri čemu se može prevesti najviše 5 jed. tereta *B*. Masa jednoga komada tereta *A* iznosi 5 jed., a dobit ostvarena njegovim transportom iznosi 6 n.j. Masa jednoga komada tereta *B* iznosi 3 jed., a dobit ostvarena njegovim transportom iznosi 4 n.j. Tvrtka želi prevesti što više tereta obiju vrsta tako da ukupna ostvarena dobit bude maksimalna. Formirajte matematički model promatranoga problema, pa nađite optimalan plan prijevoza i izračunajte pripadnu optimalnu ostvarenu dobit.

Rješenje: Neka je x ukupan broj komada tereta A, a y ukupan broj komada tereta B koji će biti preveženi. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 6 \cdot x + 4 \cdot y$$

pod uvjetima

$$5 \cdot x + 3 \cdot y \leq 45$$

$$0 \leq y \leq 5,$$

$$x, y \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 6, x_2 = 5, f_{max} = 56.$$

Prema tome, optimalan plan prijevoza je: prevesti 6 komada tereta A i 5 komada tereta B. Pripadna optimalna ukupna dobit iznosi 56 n.j.

3. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u razdoblju planiranja od $n = 4$ godine i obvezan rashod vozila početkom 5. godine. Na početku razdoblja planiranja raspolaže se s vozilom starim godinu dana. Godišnji neto-trošak održavanja i likvidacijska vrijednost vozila iskazani su tablično kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4	5	6
godišnji neto–trošak (c_i) [n.j.]	9	14	20	27	29	30	-
likvidacijska vrijednost (s_i) [n.j.]	-	16	9	6	3	1	0

Početkom svake pojedine godine razdoblja planiranja (uračunavajući prvu) donosi se odluka o zadržavanju ili zamjeni dotad korištenoga vozila tako da ukupni troškovi u cijelom razdoblju planiranja budu što manji. Pritom se pretpostavlja da je nabavna cijena vozila stalna tijekom cijeloga razdoblja planiranja i jednaka $p = 30$ n.j. Odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja i izračunajte pripadne optimalne ukupne neto-troškove.

Rješenje: Definirajmo cjelobrojnu funkciju dvije cjelobrojne varijable $V : \mathbf{N}^2 \to \mathbf{N}$ formulom

$V = V(k, i)$ = optimalni ukupni neto-troškovi od početka godine k do kraja razdoblja planiranja ako početkom godine k raspolažemo s vozilom starim i godina

Budući da na početku prve godine raspolažemo s vozilom starim godinu dana, trebamo izračunati $V(1, 1)$. Odredimo najprije izraz za izračunavanje vrijednosti funkcije $V(k, i)$. Razmatramo dvije alternative:

I. Na početku godine k zadržavamo dotad korišteno vozilo

Tada su ukupni neto–troškovi u godini k jednaki godišnjem neto–trošku održavanja vozila staroga i godina, tj. c_i n.j., pa na početku $(k + 1)$ – ve godine raspolažemo s vozilom starim $i + 1$ godina. Optimalni ukupni troškovi od početka godine $k + 1$ do kraja razdoblja ako početkom godine $k + 1$ raspolažemo s vozilom starim $i + 1$ godina iznose $V(k + 1, i + 1)$. Stoga su ukupni optimalni troškovi nastali izborom ove alternative $c_i + V(k + 1, i + 1)$.

II. Na početku godine k kupujemo novo vozilo

Tada u godini k najprije rashodujemo dotad korišteno vozilo staro i godina i ostvarujemo prihod jednak likvidacijskoj vrijednosti vozila staroga i godina, tj. s_i n.j. Potom kupujemo novo vozilo po nabavnoj cijeni p i na njegovo održavanje u godini k utrošimo iznos jednak godišnjem neto–trošku održavanja novoga vozila, tj. iznos jednak godišnjem neto–trošku održavanja vozila staroga 0 godina, tj. c_0. Na početku $(k + 1)$ – ve godine raspolažemo s vozilom starim jednu godinu, pa optimalni ukupni troškovi od početka godine $k + 1$ do kraja razdoblja ako početkom godine $k + 1$ raspolažemo s vozilom starim jednu godinu iznose $V(k + 1, 1)$. Stoga su ukupni optimalni troškovi nastali izborom ove alternative: $p + c_0 - s_i + V(k + 1, i + 1)$.

Od dvije ponuđene alternative izabrat ćemo onu koja ostvaruje manje ukupne neto-troškove, pa vrijedi jednakost:

$$V(k, i) = \min\{c_i + V(k + 1, i + 1), p + c_0 - s_i + V(k + 1, 1)\},$$

koja za $p = 30$ i $c_0 = 9$ prelazi u

$$V(k, i) = \min\{c_i + V(k + 1, i + 1), 39 - s_i + V(k + 1, 1)\}.$$

Zadajmo sada početne vrijednosti funkcije V. Znamo da početkom 5. godine nastupa obvezan rashod vozila, pa će početne vrijednosti biti vrijednosti $V(5, i)$ za $i \in [5]$ jer najveća moguća starost vozila na početku 5. godine iznosi $1 + 4 = 5$ godina (dobiva se ukoliko tijekom cijeloga razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$$V(5, 1) = -16;$$
$$V(5, 2) = -9;$$
$$V(5, 3) = -6;$$
$$V(5, 4) = -3;$$
$$V(5, 5) = -1.$$

U nastavku računamo vrijednosti funkcije $V(4, i)$ za $i \in [4]$ jer najveća moguća starost vozila na početku 4. godine iznosi $1 + 3 = 4$ godine (dobiva se ukoliko tijekom prve tri godine razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$V(4, 1) = \min\{c_1 + V(4 + 1, 1 + 1), 39 - s_1 + V(4 + 1, 1)\} = \min\{c_1 + V(5, 2), 39 - s_1 + V(5, 1)\} = \min\{14 + (-9), 39 - 16 + (-16)\} = \min\{5, 7\} = 5;$
$V(4, 2) = \min\{c_2 + V(4 + 1, 2 + 1), 39 - s_2 + V(4 + 1, 1)\} = \min\{c_2 + V(5, 3), 39 - s_2 + V(5, 1)\} = \min\{20 + (-6), 39 - 9 + (-16)\} = \min\{14, 14\} = 14;$
$V(4, 3) = \min\{c_3 + V(4 + 1, 3 + 1), 39 - s_3 + V(4 + 1, 1)\} = \min\{c_3 + V(5, 4), 39 - s_3 + V(5, 1)\} = \min\{27 + (-3), 39 - 6 + (-16)\} = \min\{24, 17\} = 17;$
$V(4, 4) = \min\{c_4 + V(4 + 1, 4 + 1), 39 - s_4 + V(4 + 1, 1)\} = \min\{c_4 + V(5, 5), 39 - s_4 + V(5, 1)\} = \min\{29 + (-1), 39 - 3 + (-16)\} = \min\{28, 20\} = 20.$

U nastavku računamo vrijednosti funkcije $V(3, i)$ za $i \in [3]$ jer najveća moguća starost vozila na početku 3. godine iznosi $1 + 2 = 3$ godine (dobiva se ukoliko tijekom prve dvije godine

razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$V(3, 1) = \min\{c_1 + V(3 + 1, 1 + 1), 39 - s_1 + V(3 + 1, 1)\} = \min\{c_1 + V(4, 2), 39 - s_1 + V(4, 1)\} = \min\{14 + 14, 39 - 16 + 5\} = \min\{28, 28\} = 28;$
$V(3, 2) = \min\{c_2 + V(3 + 1, 2 + 1), 39 - s_2 + V(3 + 1, 1)\} = \min\{c_2 + V(4, 3), 39 - s_2 + V(4, 1)\} = \min\{20 + 17, 39 - 9 + 5\} = \min\{37, 35\} = 35;$
$V(3, 3) = \min\{c_3 + V(3 + 1, 3 + 1), 39 - s_3 + V(3 + 1, 1)\} = \min\{c_3 + V(4, 4), 39 - s_3 + V(4, 1)\} = \min\{27 + 20, 39 - 6 + 5\} = \min\{47, 38\} = 38.$

U nastavku računamo vrijednosti funkcije $V(2, i)$ za $i \in [2]$ jer najveća moguća starost vozila na početku 2. godine iznosi $1 + 1 = 2$ godine (dobiva se ukoliko tijekom prve godine razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku te godine). Stoga je:

$V(2, 1) = \min\{c_1 + V(2 + 1, 1 + 1), 39 - s_1 + V(2 + 1, 1)\} = \min\{c_1 + V(3, 2), 39 - s_1 + V(3, 1)\} = \min\{14 + 35, 39 - 16 + 28\} = \min\{49, 51\} = 49;$
$V(2, 2) = \min\{c_2 + V(2 + 1, 2 + 1), 39 - s_2 + V(2 + 1, 1)\} = \min\{c_2 + V(3, 3), 39 - s_2 + V(3, 1)\} = \min\{20 + 38, 39 - 9 + 28\} = \min\{58, 58\} = 58.$

Tako je konačno:

$V(1, 1) = \min\{c_1 + V(1 + 1, 1 + 1), 39 - s_1 + V(1 + 1, 1)\} = \min\{c_1 + V(2, 2), 39 - s_1 + V(2, 1)\} = \min\{14 + 58, 39 - 16 + 49\} = \min\{72, 72\} = 72.$

Iz postupka dobivanja rezultata zaključujemo da postoji više različitih optimalnih politika zamjene vozila, a jedna od njih je:

početak 1., 2. i 4. *godine*: zadržavanje dotad korištenoga vozila;
početak 3. *godine*: zamjena (kupnja novoga) vozila.

Optimalni ukupni neto-troškovi svih optimalnih politika zamjene iznose 72 n.j.

4. Mreža ulica nekoga područja određena je s točno 5 lokacija: *A*, *B*, *C*, *D* i *E*. Ulice *CA*, *CB, DB* i *ED* su jednosmjerne (dozvoljeni smjer je od prve lokacije prema drugoj), a ulice *AD*, *BE* i *CE* dvosmjerne. Najkraće trajanje putovanja između pojedinih lokacija [u minutama] zadano je u sljedećoj tablici.

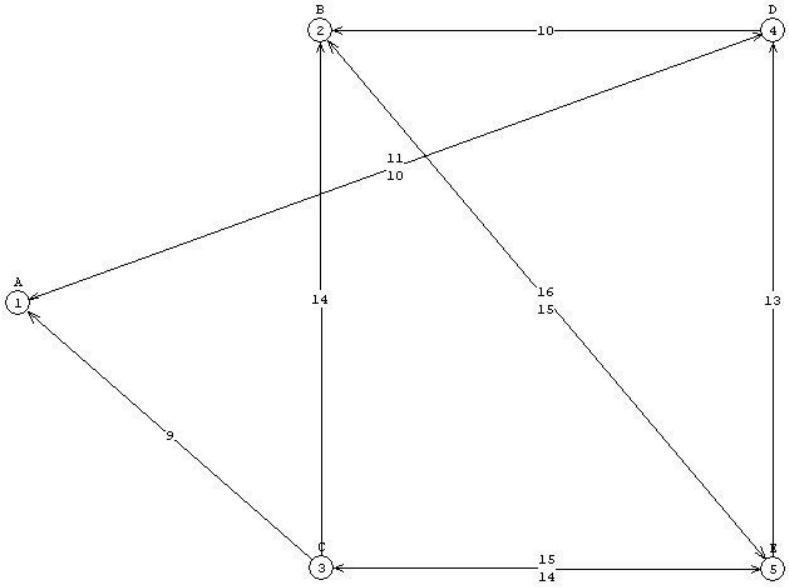
lokacija	*A*	*B*	*C*	*D*	*E*
A	0	0	0	10	0
B	0	0	0	0	15
C	9	14	0	0	14
D	11	10	0	0	0
E	0	16	15	13	0

a) Polazeći s lokacije *A* i vozeći zadanom mrežom ulica, turistički autobus obilazi svaku od preostalih lokacija i vraća se na polazište. Odredite rutu obilaska za koju je potrebno najmanje vremena i pripadno trajanje putovanja.

b) Policijska postaja nalazi se na lokaciji *D*. Može li policijska patrola polazeći s te lokacije i vozeći zadanom mrežom ulica točno jednom obići svaku pojedinu ulicu i vratiti se u postaju? Ako može, navedite pripadnu rutu obilaska. Ako ne može, navedite rutu obilaska takvu da patrola u što kraćem vremenu barem jednom obiđe svaku ulicu i izračunajte pripadno optimalno vrijeme. Sve svoje tvrdnje detaljno obrazložite.

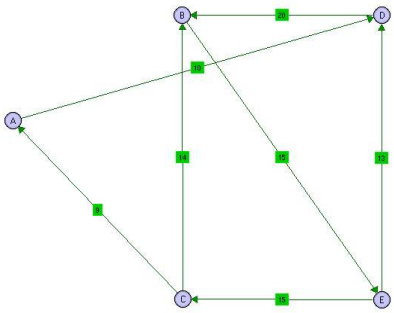
***Rješenje*:** Matematički model je usmjereni težinski graf s ukupno 5 vrhova koji označavaju lokacije, te 7 bridova čije težine su najkraća vremena prolaska ulicom određenom odgovarajućim lokacijama.

a) Jedan od mogućih modela dobiven programom *Grin* je:



Tražimo rješenje problema trgovačkoga putnika. Primjenom procedure *Salesman problem* dobivamo da je jedna moguća tražena ruta 1 – 4 – 2 – 5 – 3 – 1, odnosno *A* – *D* – *B* – *E* – *C* – – *A*. Njezina težina je 59, što znači da je optimalno vrijeme obilaska svih lokacija 59 minuta.

b) Pripadni usmjereni graf moramo modelirati tako da iz svakoga vrha, osim početnoga vrha *D*, imamo barem jedan ulazni i barem jedan izlazni luk. Jedan od mogućih modela je sljedeći graf dobiven programom *Graph Magics* je:



Tražimo postoji li Eulerov ciklus u zadanom grafu. Odmah vidimo da u vrh B ulaze dva luka, a izlazi samo jedan, što znači da će patrola pri svakom od dvaju ulazaka u vrh B morati izaći istim putem (ulicom BE). Dakle, Eulerov ciklus ne postoji, pa rješavamo problem kineskoga poštara za zadani usmjereni graf. Primjenom procedure *Chinese Postman Problem* dobivamo sljedeću rutu obilaska:

$$D - B - E - C - B - E - D - B - E - C - A - D$$

čija je ukupna težina 161. Stoga je optimalno vrijeme potrebno za obilazak svih navedenih ulica 161 minuta.

2.6. **Primjer 6.**

1. Iznos od 40 n.j. treba rasporediti na tri industrijske grane G_1, G_2 i G_3 tako da ukupna ostvarena neto-dobit bude što veća i da iznos uložen u svaku pojedinu granu bude cjelobrojan. U granu G_1 moguće je uložiti najviše 15 n.j., a za x uloženih novčanih jedinica ostvaruje se neto-dobit $0.5 \cdot x^2$ n.j. U granu G_2 moguće je uložiti najviše 20 n.j., a za y uloženih novčanih jedinica ostvaruje se neto-dobit $8 \cdot y$ n.j. U granu G_3 moguće je uložiti najviše 10 n.j., a za z uloženih novčanih jedinica ostvaruje se dobit $9 \cdot z$ n.j. Formirajte odgovarajući matematički model, pa odredite optimalnu razdiobu ukupnoga iznosa i izračunajte pripadnu optimalnu ukupnu ostvarenu neto-dobit.

Rješenje: Za svaki $i \in [3]$ neka je x_i iznos uložen u granu G_i. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 0.5 \cdot x_1^2 + 8 \cdot x_2 + 9 \cdot x_3$$

pod uvjetima

$$x_1 + x_2 + x_3 = 40$$
$$0 \le x_1 \le 15,\ 0 \le x_2 \le 20,\ 0 \le x_3 \le 10,$$
$$x_1, x_2, x_3 \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 15,\ x_2 = 15,\ x_3 = 10,\ f_{max} = 322{,}50.$$

Dakle, optimalan plan ulaganja je: u grane G_1 i G_2 treba uložiti po 15 n.j., a u granu G_3 10 n.j. Pripadna optimalna ukupna neto-dobit iznosi 322,50 n.j.

2. Tvrtka "*Gulikožić* d.o.o." raspolaže s kamionom nosivosti 45 jed. Tim kamionom treba prevesti dvije vrste tereta: A i B, pri čemu se može prevesti najviše 5 jed. tereta B. Masa jednoga komada tereta A iznosi 10 jed., a dobit ostvarena njegovim transportom iznosi 12 n.j. Masa jednoga komada tereta B iznosi 8 jed., a dobit ostvarena njegovim transportom iznosi 10 n.j. Tvrtka želi prevesti što više tereta obiju vrsta tako da ukupna ostvarena dobit bude

maksimalna. Formirajte matematički model promatranoga problema, pa nađite optimalan plan prijevoza i izračunajte pripadnu optimalnu ostvarenu dobit, te postotak iskorištenosti nosivosti kamiona.

Rješenje: Neka je x ukupan broj komada tereta A, a y ukupan broj komada tereta B koji će biti preveženi. Tada je matematički model promatranoga problema:

$$\text{maksimizirati } f(x_1, x_2, x_3) = 12 \cdot x + 10 \cdot y$$

pod uvjetima

$$10 \cdot x + 8 \cdot y \leq 45$$
$$0 \leq y \leq 5,$$
$$x, y \in \mathbf{N}.$$

Koristeći potprogram *DP* programa *Winqsb* dobivamo optimalno rješenje:

$$x_1 = 2, x_2 = 3, f_{max} = 54.$$

Prema tome, optimalan plan prijevoza je: prevesti 2 komada tereta A i 3 komada tereta B. Pripadna optimalna ukupna dobit iznosi 54 n.j., a postotak iskorištenosti nosivosti kamiona

$$p = \frac{10 \cdot 2 + 8 \cdot 3}{45} \cdot 100 = 97{,}78\%.$$

3. Utvrđena poslovna politika neke tvrtke predviđa korištenje određenoga tipa vozila u razdoblju planiranja od $n = 4$ godine i obvezan rashod vozila početkom 5. godine. Na početku razdoblja planiranja raspolaže se s vozilom starim godinu dana. Godišnji neto-trošak održavanja i likvidacijska vrijednost vozila iskazani su tablično kao funkcije starosti vozila:

starost vozila (i) [god.]	0	1	2	3	4	5	6
godišnji neto–trošak (c_i) [n.j.]	9	14	20	25	27	29	30
likvidacijska vrijednost (s_i) [n.j.]	-	16	9	7	4	2	1

Početkom svake pojedine godine razdoblja planiranja (uračunavajući prvu) donosi se odluka o zadržavanju ili zamjeni dotad korištenoga vozila tako da ukupni troškovi u cijelom razdoblju planiranja budu što manji. Pritom se pretpostavlja da je nabavna cijena vozila stalna tijekom cijeloga razdoblja planiranja i jednaka $p = 30$ n.j. Odredite optimalnu politiku zamjene u cjelokupnom razdoblju planiranja i izračunajte pripadne optimalne ukupne neto-troškove.

Rješenje: Definirajmo cjelobrojnu funkciju dvije cjelobrojne varijable $V : \mathbf{N}^2 \to \mathbf{N}$ formulom

$V = V(k, i)$ = optimalni ukupni neto-troškovi od početka godine k do kraja razdoblja planiranja ako početkom godine k raspolažemo s vozilom starim i godina

Budući da na početku prve godine raspolažemo s vozilom starim godinu dana, trebamo izračunati $V(1, 1)$. Odredimo najprije izraz za izračunavanje vrijednosti funkcije $V(k, i)$. Razmatramo dvije alternative:

I. Na početku godine k zadržavamo dotad korišteno vozilo

Tada su ukupni neto–troškovi u godini k jednaki godišnjem neto–trošku održavanja vozila staroga i godina, tj. c_i n.j., pa na početku $(k + 1)$ – ve godine raspolažemo s vozilom starim i + + 1 godina. Optimalni ukupni troškovi od početka godine $k + 1$ do kraja razdoblja ako početkom godine $k + 1$ raspolažemo s vozilom starim $i + 1$ godina iznose $V(k + 1, i + 1)$. Stoga su ukupni optimalni troškovi nastali izborom ove alternative $c_i + V(k + 1, i + 1)$.

II. Na početku godine k kupujemo novo vozilo

Tada u godini k najprije rashodujemo dotad korišteno vozilo staro i godina i ostvarujemo prihod jednak likvidacijskoj vrijednosti vozila staroga i godina, tj. s_i n.j. Potom kupujemo novo vozilo po nabavnoj cijeni p i na njegovo održavanje u godini k utrošimo iznos jednak godišnjem neto–trošku održavanja novoga vozila, tj. iznos jednak godišnjem neto–trošku održavanja vozila staroga 0 godina, tj. c_0. Na početku $(k + 1)$ – ve godine raspolažemo s vozilom starim jednu godinu, pa optimalni ukupni troškovi od početka godine $k + 1$ do kraja razdoblja ako početkom godine $k + 1$ raspolažemo s vozilom starim jednu godinu iznose $V(k + 1, 1)$. Stoga su ukupni optimalni troškovi nastali izborom ove alternative: $p + c_0 - s_i$ + + $V(k + 1, i + 1)$.

Od dvije ponuđene alternative izabrat ćemo onu koja ostvaruje manje ukupne neto-troškove, pa vrijedi jednakost:

$$V(k, i) = \min\{c_i + V(k + 1, i + 1), p + c_0 - s_i + V(k + 1, 1)\},$$

koja za $p = 30$ i $c_0 = 9$ prelazi u

$$V(k, i) = \min\{c_i + V(k + 1, i + 1), 39 - s_i + V(k + 1, 1)\}.$$

Zadajmo sada početne vrijednosti funkcije V. Znamo da početkom 5. godine nastupa obvezan rashod vozila, pa će početne vrijednosti biti vrijednosti $V(5, i)$ za $i \in [5]$ jer najveća moguća starost vozila na početku 5. godine iznosi $1 + 4 = 5$ godina (dobiva se ukoliko tijekom cijeloga razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$$\begin{aligned} V(5, 1) &= -16; \\ V(5, 2) &= -9; \\ V(5, 3) &= -7; \\ V(5, 4) &= -4; \\ V(5, 5) &= -2. \end{aligned}$$

U nastavku računamo vrijednosti funkcije $V(4, i)$ za $i \in [4]$ jer najveća moguća starost vozila na početku 4. godine iznosi $1 + 3 = 4$ godine (dobiva se ukoliko tijekom prve tri godine razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$V(4, 1) = \min\{c_1 + V(4 + 1, 1 + 1), 39 - s_1 + V(4 + 1, 1)\} = \min\{c_1 + V(5, 2), 39 - s_1 +$ $+ V(5, 1)\} = \min\{14 + (-9), 39 - 16 + (-16)\} = \min\{5, 7\} = 5;$

$V(4, 2) = \min\{c_2 + V(4 + 1, 2 + 1), 39 - s_2 + V(4 + 1, 1)\} = \min\{c_2 + V(5, 3), 39 - s_2 + V(5, 1)\} = \min\{20 + (-7), 39 - 9 + (-16)\} = \min\{13, 14\} = 13;$
$V(4, 3) = \min\{c_3 + V(4 + 1, 3 + 1), 39 - s_3 + V(4 + 1, 1)\} = \min\{c_3 + V(5, 4), 39 - s_3 + V(5, 1)\} = \min\{25 + (-4), 39 - 7 + (-16)\} = \min\{21, 16\} = 16;$
$V(4, 4) = \min\{c_4 + V(4 + 1, 4 + 1), 39 - s_4 + V(4 + 1, 1)\} = \min\{c_4 + V(5, 5), 39 - s_4 + V(5, 1)\} = \min\{27 + (-2), 39 - 4 + (-16)\} = \min\{25, 19\} = 19.$

U nastavku računamo vrijednosti funkcije $V(3, i)$ za $i \in [3]$ jer najveća moguća starost vozila na početku 3. godine iznosi 1 + 2 = 3 godine (dobiva se ukoliko tijekom prve dvije godine razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku prve godine). Stoga je:

$V(3, 1) = \min\{c_1 + V(3 + 1, 1 + 1), 39 - s_1 + V(3 + 1, 1)\} = \min\{c_1 + V(4, 2), 39 - s_1 + V(4, 1)\} = \min\{14 + 13, 39 - 16 + 5\} = \min\{27, 28\} = 27;$
$V(3, 2) = \min\{c_2 + V(3 + 1, 2 + 1), 39 - s_2 + V(3 + 1, 1)\} = \min\{c_2 + V(4, 3), 39 - s_2 + V(4, 1)\} = \min\{20 + 16, 39 - 9 + 5\} = \min\{36, 35\} = 35;$
$V(3, 3) = \min\{c_3 + V(3 + 1, 3 + 1), 39 - s_3 + V(3 + 1, 1)\} = \min\{c_3 + V(4, 4), 39 - s_3 + V(4, 1)\} = \min\{25 + 19, 39 - 7 + 5\} = \min\{44, 37\} = 37.$

U nastavku računamo vrijednosti funkcije $V(2, i)$ za $i \in [2]$ jer najveća moguća starost vozila na početku 2. godine iznosi 1 + 1 = 2 godine (dobiva se ukoliko tijekom prve godine razdoblja planiranja zadržimo vozilo staro godinu dana s kojim raspolažemo na početku te godine). Stoga je:

$V(2, 1) = \min\{c_1 + V(2 + 1, 1 + 1), 39 - s_1 + V(2 + 1, 1)\} = \min\{c_1 + V(3, 2), 39 - s_1 + V(3, 1)\} = \min\{14 + 35, 39 - 16 + 27\} = \min\{49, 50\} = 49;$
$V(2, 2) = \min\{c_2 + V(2 + 1, 2 + 1), 39 - s_2 + V(2 + 1, 1)\} = \min\{c_2 + V(3, 3), 39 - s_2 + V(3, 1)\} = \min\{20 + 37, 39 - 9 + 27\} = \min\{57, 57\} = 57.$

Tako se konačno dobiva:

$V(1, 1) = \min\{c_1 + V(1 + 1, 1 + 1), 39 - s_1 + V(1 + 1, 1)\} = \min\{c_1 + V(2, 2), 39 - s_1 + V(2, 1)\} = \min\{14 + 57, 39 - 16 + 49\} = \min\{71, 72\} = 71.$

Prema tome, imamo više različitih optimalnih politika, a jedna od njih je:

početak 1., 2. i 4. *godine*: zadržavanje dotad korištenoga vozila;
početak 3. *godine*: zamjena (kupnja novoga) vozila.
Optimalni ukupni neto-troškovi svih optimalnih politika zamjene iznose 71 n.j.

4. Mreža ulica nekoga područja određena je s točno 5 lokacija: *A*, *B*, *C*, *D* i *E*. Ulice *AD*, *AE*, *BC* i *BE* su jednosmjerne (dozvoljeni smjer je od početne lokacije prema krajnjoj), a ulice *AC*, *BD* i *CE* dvosmjerne. Najkraće trajanje putovanja između pojedinih lokacija [u minutama] zadano je u sljedećoj tablici.

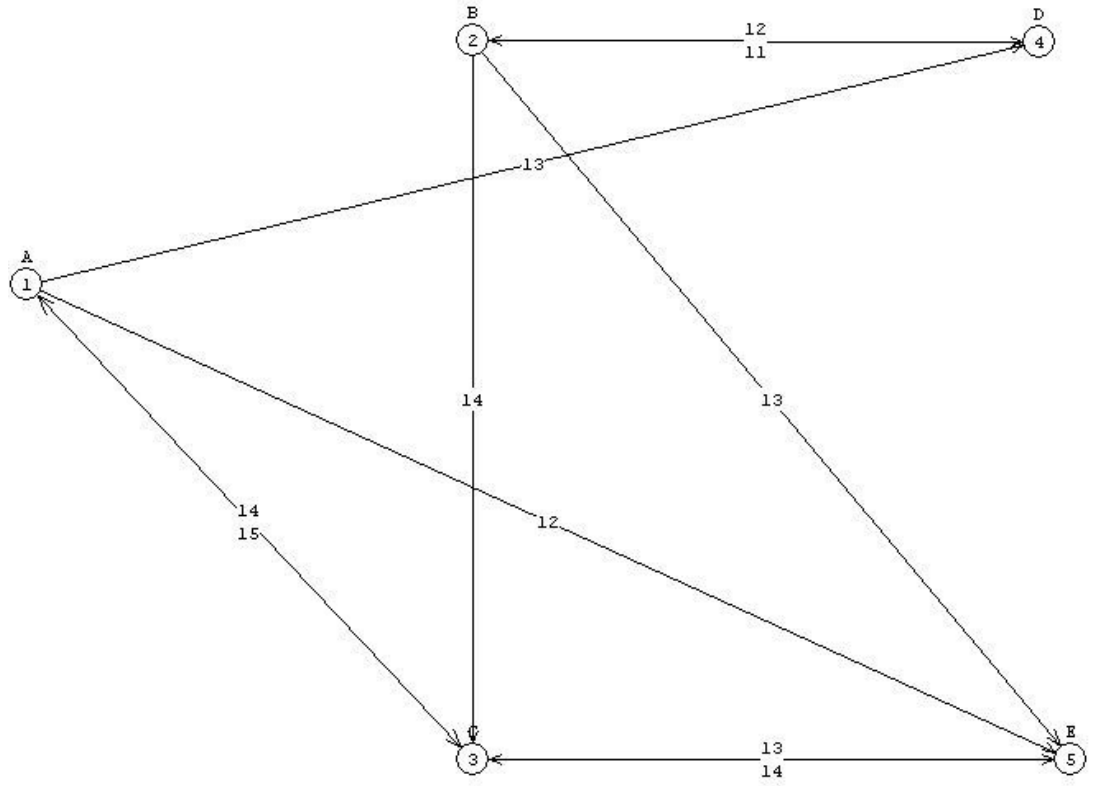
lokacija	*A*	*B*	*C*	*D*	*E*
A	0	0	15	13	12
B	0	0	14	11	13
C	14	0	0	0	14
D	0	12	0	0	11
E	0	0	13	0	0

a) Turistički autobus kreće s lokacije *A* i, vozeći zadanom mrežom ulica, obilazi preostale lokacije i vraća se na lokaciju *A*. Odredite rutu obilaska za koju je potrebno najmanje vremena i pripadno trajanje putovanja.

b) Policijska stanica nalazi se na lokaciji *C*. Može li policijska patrola polazeći s te lokacije i vozeći zadanom mrežom ulica točno jednom obići svaku pojedinu ulicu i vratiti se u postaju? Ako može, navedite pripadnu rutu obilaska. Ako ne može, obrazložite svoj odgovor, pa navedite rutu obilaska takvu da patrola u što kraćem vremenu barem jednom obiđe svaku ulicu i izračunajte pripadno optimalno vrijeme.

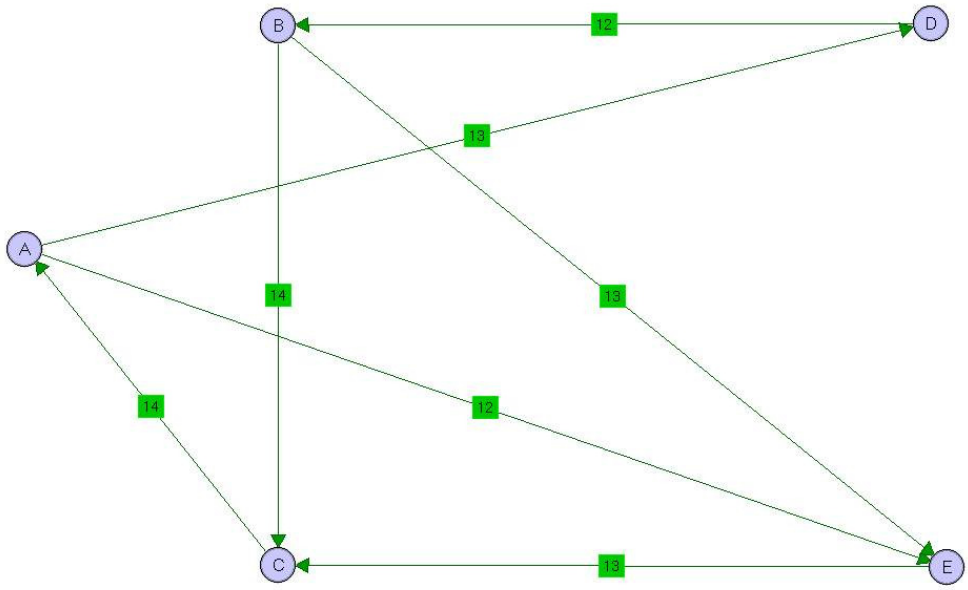
Rješenje: Matematički model je usmjereni težinski graf s ukupno 5 vrhova koji označavaju lokacije, te 7 bridova čije težine su najkraća vremena prolaska ulicom određenom odgovarajućim lokacijama.

a) Jedan od mogućih modela dobiven programom *Grin* je:



Tražimo rješenje problema trgovačkoga putnika. Primjenom procedure *Salesman problem* dobivamo da je jedna moguća tražena ruta 1 – 4 – 2 – 5 – 3 – 1, odnosno $A - D - B - E - C - A$. Njezina težina je 65, što znači da je optimalno vrijeme obilaska svih lokacija 65 minuta.

b) Pripadni usmjereni graf moramo modelirati tako da iz svakoga vrha, osim početnoga vrha C, imamo barem jedan ulazni i barem jedan izlazni luk. Odmah vidimo da, zbog jednosmjerne ulice BC i dvosmjernih ulica AC i CE, broj lukova koji ulaze u vrh C ne može biti jednak broju vrhova koji izlaze iz vrha C, pa će patrola iz taj vrh barem dvaput morati doći ulicom CA. Stoga zadani graf ne sadrži Eulerov ciklus, pa u nastavku rješavamo problem kineskoga poštara za zadani usmjereni graf.



Primjenom procedure *Chinese Postman Problem* dobivamo sljedeću optimalnu rutu obilaska:

$$C - A - D - B - E - C - A - E - C - A - D - B - C$$

čija je ukupna težina 157. Stoga optimalno vrijeme potrebno za barem jedan obilazak svih ulica zadane mreže iznosi 157 minuta.

3. Linkovi za preuzimanje računalnih programa

1. Računalni program *WinQSB* besplatno se može preuzeti sa sljedeće WEB-adrese:

http://www.wiley.com/college/tech/winqsb.htm

Pritom valja napomenuti da, uz polaznu datoteku *winqsb.exe* treba preuzeti i dodatnu datoteku *runtime.exe* i njezin komprimirani sadržaj dekomprimirati u istu mapu u kojoj je dekomprimiran sadržaj datoteke *winqsb.exe*.

2. Studentska verzija računalnoga programa *Grin* (skraćenica riječi *Graph Interface*) besplatno se može preuzeti s osobne WEB-stranice autora programa Vitalyja Pechenkina:

http://www.geocities.com/pechv_ru/

Putem iste stranice može se nabaviti i verzija istoga programa s nešto više dodatnih mogućnosti korisnih za rješavanje problema u teoriji grafova.

3. Besplatna 30-dnevna probna verzija računalnoga programa *Graph Magics* može se preuzeti sa sljedeće WEB-adrese:

http://www.graph-magics.com/download.php

Pritom valja napomenuti da je za ispravan rad programa nužno instalirati program *Microsoft .NET Framework*, što se može učiniti izravno s navedene WEB-stranice. Po isteku 30 dana probnu verziju više nije moguće koristiti (nužna je kupnja programa, odnosno registracija).

4. Literatura (na hrvatskom jeziku)

1. B. Divjak, A. Lovrenčić: *Diskretna matematika s teorijom grafova*, FOI – TIVA, Varaždin, 2005.
2. D. Kalpić, V. Mornar: *Operacijska istraživanja*, DRIP, Zagreb, 1996.
3. Lj. Martić: *Matematičke metode za ekonomske analize* 2, Narodne novine, Zagreb, 1976.
4. J. Petrić: *Operaciona istraživanja*, Nauka, Beograd, 1998.
5. J. Petrić et.al.: *Zbirka zadataka iz operacionih istraživanja*, Nauka, Beograd, 1998.
6. D. Veljan: *Kombinatorna i diskretna matematika*, Algoritam, Zagreb, 2001.

Internetske stranice (sve javno dostupne 19.07.2008.)

1. http://eie507.eie.polyu.edu.hk/ss-submission/B7a/
2. http://www.heinemann.co.uk/shared/Resources/NonSecure/00000000/Chp-03%20044-064.pdf
3. http://www.heinemann.co.uk/shared/Resources/NonSecure/00000000/Chp-02%20023-043.pdf
4. http://csci.biola.edu/csci480spring03/knapsack.pdf
5. http://facultystaff.richmond.edu/~dszajda/classes/cs315/Spring_2008/slides/Chapter5.3.ppt
6. https://www.utdallas.edu/~scniu/OPRE-6201/documents/DP2-Equipment-Replacement.pdf
7. http://www.mpri.lsu.edu/textbook/Chapter7-b.htm#optimale
8. http://en.wikipedia.org/wiki/Route_inspection_problem
9. http://en.wikipedia.org/wiki/Traveling_salesman_problem

5. Dodatak

Na temelju teksta ove skripte objavljeni su sljedeći stručni radovi:

1. B. Kovačić: *Kvantitativne metode odlučivanja – problem optimalne zamjene opreme*, Math.e. Vol. 15, Hrvatsko matematičko društvo, Zagreb, 2009.

2. B. Kovačić, T. Strmečki: *Kvantitativne metode odlučivanja – problem jednostavne razdiobe ulaganja*, Math.e, Vol. 19, Hrvatsko matematičko društvo, Zagreb, 2011.

3. B. Kovačić, I. Božić, T. Strmečki: *Kvantitativne metode odlučivanja – problem složene razdiobe ulaganja*, Math.e., Vol. 22, Hrvatsko matematičko društvo, Zagreb, 2012.